Utrecht University

Faculty of Science

Department of Mathematics

Numerical comparison of some dimension reduction techniques for time series data from PDEs

Master Thesis

Author:
Rabia Afzal

Supervisor:
Prof. Dr. Ir. J.E. Frank

Second reader:
Dr. P.A. Zegeling

December 6, 2022

Abstract

There has been increasing use of dimensionality reduction (DR) techniques to deal with high dimensional data in order to minimize the number of dimensions, referred to as the number of variables or degrees of freedom. These techniques are applied to the data prior to modeling and give insight into the data, help reduce storage space required, and improve the computational cost of the models, due to reducing the number of variables. In extreme cases, some solutions of the partial differential equation (PDE) may be characterized by a finite number of degrees of freedom (or variables). In this thesis, four DR techniques, namely empirical orthogonal function, diffusion maps, extended dynamic mode decomposition and approximated Lax pairs, will be used for data sets formed from the approximating solution sequence of the two different PDEs. The idea of using the spectral theory for dimensionality reduction will be deployed to form the basis set presented as the set of the coordinates for the DR methods. The main motivation is to analyze and compare the four DR techniques. Consequently, it will be presented which DR method indicates the best approximation of the original data set for both types of data under the mean squared error criterion, in contrast to other methods.

Contents

Chapter 1

Introduction to dimensionality reduction

The number of degrees of freedom or quantities describing each state in a given time series is known as its dimensionality. High dimensionality continues to be a challenge in computational modeling, for instance, large-scale partial differential equation (PDE) models in climate science and engineering. Also, the high dimensionality of the data obviously makes the visualization of objects complex. Some of these variables are partially redundant, adding noise to the data, and are independent of each other. Therefore, these redundant variables can be eliminated from the data. The process used to reduce these variables or columns is called dimensionality reduction (DR). In other words, this is a process to transform data from high-dimensional space (original space) to low-dimensional space while preserving the relationships between the objects in the original space. Dimensionality reduction [24] leads the useful properties for database and machine learning systems such as data compression, better data visualization, and efficient data retrieval. Furthermore, a reduced number of dimensions in the data set provides less computational time and resources. It is often desirable to construct low-dimensional representations to optimize in a low-dimensional space, classify low-dimensional behavior and identify ”reaction coordinates” that provide a simplified observable state of the system.

In order to define the dimensionality reduction, we will consider a high dimensional time-series data set $X = (X_1, X_2, \ldots, X_M) \in \mathbb{R}^{N\times M}$, consisting of M data vectors $X_i \in \mathbb{R}^N$, $i = 1, \ldots, M$. The main objective of dimensionality reduction is to transform the higher dimensional data set X into the lower dimensional representation $\tilde{X} = (\tilde{X}_1, \tilde{X}_2, \ldots, \tilde{X}_M) \in \mathbb{R}^{k\times M}$, consisting M data vectors $\tilde{X}_i \in \mathbb{R}^k$, $i = 1, \ldots, M$, such that $k \ll N$, retaining the meaningful properties of the original data set.

The concept of the geometry of a set $\tilde{X}$ of objects means the set of rules which describe the relationship between the elements of $\tilde{X}$. To describe this, we assume that $\tilde{X}$ is a subset of the set X. The geometry of the set $\tilde{X}$ is referred to as intrinsic geometry [14] if the rules are defined without using a reference to X and possible geometric structures already existing on it. On the other hand, If X has its own geometry and this geometry is induced on $\tilde{X}$, then this geometry is called extrinsic geometry of $\tilde{X}$. Also, we assume that X is with intrinsic dimensionality [25], which means that the points in X lie on or near a manifold with dimensionality k embedded in the N-dimensional space. There is no assumption on the structure or geometry of the manifold and also intrinsic dimensionality k of the data set X is not known. For this reason, dimensionality reduction is an ill-posed problem solved by considering certain properties, such that intrinsic dimensionality, of the data.

Dimensionality reduction techniques can be separated into linear or non-linear techniques. In linear dimensionality reduction methods [6], a linear transformation matrix A is defined such that, for any i, $\tilde{X}_i = A^T X_i$, where X_i is an element of high-dimensional space and $\tilde{X}_i$ is in low-dimensional space. The non-linear DR method is based on the non-linear data model. In the linear dimensionality reduction method, like empirical orthogonal function (EOF), we maximize the variance preservation or minimize the reconstruction error. However, in the nonlinear case [15], distance preservation has become the first standard used to get dimensionality reduction. The main principle behind distance preservation is that we need to describe any space or manifold fully by pairwise distances. If a low dimensional representation can be achieved by reproducing the initial distances and preserving its geometrical structure (information on the manifold), then the dimensionality reduction is successful. This means that if close data points remain close and if far data points remain far, then the original manifold and its low dimensional representation have a similar shape. To deal with non-linear data, the basic idea is to use the kernel function which maps the original high dimensional data into the larger dimensional space. The kernel function separates the different distributions of the data by forming a matrix of pairwise distances. Then the resultant matrix can be processed by using a spectral decomposition as in the diffusion map (DM) process. Non-linear techniques are more powerful and complex than linear ones because the connection between the variables or features might be richer than a simple matrix linear transformation. Furthermore, there are two important categories of dimensionality reduction techniques [24], such as Feature selection techniques and Feature projection methodologies. The feature selection techniques retain the most important dimensions and discard the redundant ones. Whereas the feature projection methodologies project the existing features or dimensions onto the new different dimensions retaining the significant properties of the data set structure and its variance as closely as possible. One of the Feature projection techniques is the Empirical Orthogonal function (EOF) which will be used later.

As is known, some solutions of the partial differential equation (PDE) can be characterized by a finite number of variables. For instance, a motion of a single soliton as the solution of a nonlinear wave equation might be reduced to just a single dynamical variable describing, say, the location of the soliton at time t. In this thesis, the time-series data X is given by a sequence of solution vectors $\{u^n\}_{n=1}^{M}$, with M number of time steps, of the partial differential equation (PDE) such as the 1D advection equation and non-linear Korteweg-de Vries (KdV) equation. We will use four DR techniques to find the new representation $\tilde{X}$ with a reduced number of coordinates compared to the original data X. As is known, the study of eigenvalues provides a way to use the eigenfunctions for dimensionality reduction. The main idea is to form a set of eigenfunctions as a basis in all four DR techniques and the basis set is taken as the set of coordinates. Then our goal is to estimate the smallest order basis as the intrinsic reduced dimensionality for the new representation of the original data. In the EOF, DM, and extended dynamic mode decomposition (EDMD) methods, we consider the entire time series as a data set to create a static orthonormal basis whose dominant modes serve as reduced coordinates to approximate the original data. However, in the case of the ALP algorithm, we will take the initial condition of PDE and also create the basis as the function of time at each time step. Then we will project the initial condition onto a time-dependent basis at each time step and form a data set of new representations of the original data. Ultimately, we will compare and contrast the different kinds of methods in terms of the root-mean-squared approximation error for a given reduced dimensionality as well as temporal error growth.

In the next chapter, we will give an overview of four DR techniques, namely empirical orthogonal function (EOF), diffusion maps (DM), extended dynamic mode decomposition (EDMD), and approximated Lax pairs (ALP). In Chapter 3, we find the optimal approximations for both data sets respectively by implementing these methods in Matlab. Furthermore, we compare the results obtained by the four different algorithms. Finally, Chapter 4 provides the conclusion.

Chapter 2

Dimensionality reduction methods based on data and dynamics

In this chapter, we will review four methods for dimensionality reduction, namely Empirical orthogonal functions (EOFs), Diffusion maps (DM), Extended dynamic mode decomposition (EDMD), and Approximated Lax-pairs (ALP) analysis. The first three of these are data-driven, requiring no explicit knowledge of the underlying process to produce a static basis for reduction. The forth method, ALP, constructs a reduced model in an evolving basis, requiring explicit knowledge of the full model.

2.1 Empirical Orthogonal Function (EOF) analysis

EOF analysis is commonly used for dimensionality reduction, e.g. in atmospheric science. They provide a way to shape the data in space-time relationships. Empirical orthogonal functions (EOFs) are efficient to project the original data on an orthogonal basis obtained by finding the eigenvectors of the covariance matrix of the data set. The corresponding eigenvalues describe a measure of the percentage of the variance contributed by each EOF mode. Thus, the truncated EOF expansion is optimal if in the sense that the maximum variance is captured in reduced dimensional approximations. In fact, EOFs of a space-time process is an orthogonal linear transformation into new lower dimensional space such that the greatest data variance on the first EOF, the second largest variance on the second EOF, and so on are contributed by any projection of the data.

EOFs suffer a geometric drawback [12] owing to the orthogonality of the EOF patterns in space and time and decorrelation of the associated time series components (EOF coefficients, explained in the method). For example, neuroscience data are in rarely orthogonal patterns which cannot be well reflected by EOF.

2.1.1 Formulation of EOFs

The construction of empirical orthogonal functions is largely adapted from [22, 18, 13] for a data set. We consider spatio-temporal processes generated by numerical discretization of PDEs. The numerical time series is given by a data set $X = (X_1, X_2, \ldots, X_M)$, where $X_i \in \mathbb{R}_N$ represents the solution, e.g. on a grid of resolution N at (uniformly spaced) time t_i. The time series generates an $N \times M$ matrix as follows,

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1M} \\ x_{21} & x_{22} & \dots & x_{2M} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N1} & x_{N2} & \dots & x_{NM} \end{pmatrix} \tag{2.1}$$

The mean of the state at time i $X_i = (x_{1i}, x_{2i}, \dots, x_{Ni})^T,\ i = 1, \dots, M$, is given as:

$$\langle X_i \rangle = \frac{1}{N} \sum_{k=1}^{N} x_{ki} \tag{2.2}$$

For convenience, we will also use the notation $\bar{X}_i$ for mean. The data matrix can also be transformed to have a zero mean columns. The nonzero mean is subtracted out from each corresponding column of X and mean centered data is given as:

$$X = X - \mathbf{1}_N \bar{X} \tag{2.3}$$

where $\mathbf{1}_N$ is the column vector of all ones of length N and $\bar{X} = (\bar{X}_1, \bar{X}_2, \dots, \bar{X}_M)$ is the vector of sample means (column-wise means) of length M. The real symmetric covariance matrix [20] of new data matrix X, given in 2.3, is defined by:

$$C(X, X) = XX^T \tag{2.4}$$

where $C(X, X) \in \mathbf{R}^{N \times N}$. We will denote it by C for simplicity. For any vector a and random vector Y, $a^T C(Y, Y) a = C(a^T Y, a^T Y) = \text{Var}(a^T Y) \geq 0$ as covariance is a generalization of variance such that $C(Y, Y) = \text{Var}(Y)$ and $a^T Y$ is a scalar random variable. This implies that the covariance matrix is positive and semi-definite. The eigen structure for covariance matrix C:

$$CE = E\Lambda \tag{2.5}$$

where C is $N \times N$ square matrix, Λ is the $N \times N$ diagonal matrix which have eigen values of C on the diagonal and E is $N \times N$ square matrix with each column of E containing the corresponding eigenvector of C.

For symmetric C, the eigenvectors may be chosen orthonormal. Since positive semi-definite matrix C has all non-negative eigenvalues. We assume the eigenvalues of C are ordered and decreasing such that $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N \geq 0$. Hence the eigenvectors $e_i, i = 1, \dots, N$ of C form an orthonormal basis in $\mathbf{R}^N$, and empirical orthogonal functions (EOFs) can be defined by these orthogonal eigenvectors of C.

Also the eigenvector matrix E, defined in (2.5), gives an orthonormal matrix such that $E^{-1} = E^T$ and $C = E\Lambda E^T$ or $E^T C E = \Lambda$. Now any $z \in \mathbf{R}^N$ can be written in the form of a linear combination of the eigenvectors e_i,

$$z = \sum_{n=1}^{N} c_n e_n = EC_1, \tag{2.6}$$

where C_1 is the matrix of constants c_i. we pre-multiply (2.6) by E^T and obtain

$$E^T z = E^T E C_1 = C_1$$

or it can be written as

$$c_n = z \cdot e_n$$

where expansion coefficients c_n are projections of z on e_n.

Let $Y \in \mathbf{R}^N$, where Y is any column vector of X, with mean 0. We show in what sense EOF projection of X is optimal. In particular, choose an eigenvector (or one pattern) $e_1 \in \mathbf{R}^N$ which satisfies the condition $\|e_1\|_2^2 = \sum_{i=1}^N |e_{1i}|^2 = 1$ such that error $\epsilon_1 = \langle \|Y - (Y.e_1)e_1\|_2^2 \rangle$ is minimized in a mean-square sense. Here, ϵ_1 is the difference between Y and the projection of Y on e_1. To make it minimal, we will first consider:

$$\begin{aligned}\|Y - (Y \cdot e_1)e_1\|_2^2 &= (Y - (Y \cdot e_1)e_1, Y - (Y \cdot e_1)e_1)\\ &= \|Y\|_2^2 - (Y \cdot e_1)^2\end{aligned}$$

This implies

$$\langle \|Y - (Y \cdot e_1)e_1\|_2^2 \rangle = \langle \|Y\|_2^2 \rangle - \langle (Y \cdot e_1)^2 \rangle \tag{2.7}$$

$$\Rightarrow \quad \epsilon_1 = \sum_{i=1}^N \langle |Y_i|^2 \rangle - \langle (Y \cdot e_1)^2 \rangle \tag{2.8}$$

By using the property of the variance and mean such that $\mathrm{Var}(Y) = \langle Y^2 \rangle - (\langle Y \rangle)^2$ and $\mathrm{Var}Y = \sum_{i=1}^N \mathrm{Var}(Y_i)$, equation (2.8) becomes

$$\epsilon_1 = \mathrm{Var}Y - \langle (Y \cdot e_1)^2 \rangle \tag{2.9}$$

Also,

$$(Y \cdot e_1)^2 = \sum_{i=1}^N \sum_{j=1}^N Y_i Y_j e_{1i} e_{1j} = e_1^T Y Y^T e_1 \tag{2.10}$$

Since ϵ_1 is a function of $e_1 = (e_{11}, e_{12}, \ldots, e_{1N})$. We suppose the following function G by using the Lagrange multiplier method,

$$G(e_1) = \epsilon_1 + \lambda(\sum_{i=1}^N e_{1i}^2 - 1)$$

where λ is the Lagrange multiplier. Now equations (2.9) and (2.10) imply that

$$G(e_1) = \mathrm{Var}Y - \sum_{i=1}^N \sum_{j=1}^N Y_i Y_j e_{1i} e_{1j} + \lambda(\sum_{i=1}^N e_{1i}^2 - 1)$$

Also, we assume that partial derivative of G are zero, i.e. $\frac{\partial G}{\partial e_{1i}} = 0$, $i = 1, \ldots, N$, then

$$\frac{\partial G}{\partial e_{1i}} = -2\sum_{j=1}^N Y_i Y_j e_{1j} + 2\lambda e_{1i} = 0, \ i = 1, \ldots, N.$$

In matrix form, above identity implies that

$$C(Y, Y)e_1 = \lambda e_1$$

where $C(Y, Y) = (Y_i Y_j)_{i,j=1,\ldots,N}$ is the covariance matrix of Y, then Since e_1 is an eigenvector of C corresponding to the eigenvalue λ_1 and by the definition of covariance matrix given in (2.4), (2.9) implies:

$$\begin{aligned}\epsilon_1 &= \mathrm{Var}Y - \lambda_1 \sum_{i=1}^N e_{1i}^2\\ &= \mathrm{Var}Y - \lambda_1\end{aligned} \tag{2.11}$$

Thus $\epsilon_1 = \text{Var}Y - \lambda_1$ is minimal as λ_1 is the largest eigenvalue and e_1 is called the first EOF. To continue the procedure, we take second eigenvector $e_2 \in \mathbf{R}^N$ with the conditions $\|e_2\|_2^2 = 1$ and e_2 is orthogonal to e_1 such that the misfit

$$\epsilon_2 = \langle \|Y - (Y \cdot e_1)e_1 - (Y \cdot e_2)e_2\|_2^2 \rangle \tag{2.12}$$

is minimized. By using the (2.7) and (2.11), (2.12) implies,

$$\begin{aligned}\epsilon_2 &= \langle \|Y - (Y \cdot e_1)e_1\|_2^2 \rangle - \langle (Y \cdot e_2)^2 \rangle \\ &= \text{Var}Y - \lambda_1 - \langle (Y \cdot e_2)^2 \rangle\end{aligned}$$

Similar arguments done previously will be used for $\langle (Y \cdot e_2)^2 \rangle$ and since e_2 is the eigenvector corresponding to the second largest eigenvalue λ_2 of C, above equation becomes,

$$\epsilon_2 = \text{Var}Y - \lambda_1 - \lambda_2 \tag{2.13}$$

Thus $\epsilon_2 = \text{Var}Y - \sum_{k=1}^{2} \lambda_k$ is minimal and e_2 is the second EOF. Proceeding this procedure, we conclude that the orthonormal eigenvectors $e_1, e_2, \ldots, e_N$ of covariance matrix C corresponding to the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_N$ such that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N$ form EOFs of X.
More precisely, X_n, for some time level n, can be expressed as a linear combination of the orthonormal eigenvectors $e_k, k = 1, \ldots, N$:

$$X_n = \sum_{i=1}^{N} \alpha_i e_i, \tag{2.14}$$

where the coefficients α_i are chosen in a way such that they are the coefficients of projection of X_n onto the empirical orthonormal basis e_i. Then $\alpha_i = e_i \cdot X_n$. Thus (2.14) for the entire time series X implies that $\alpha = E^T X$ as $E = (e_1, e_2, \ldots, e_N)$ are orthogonal. This means that each solution has the eigenmodes in a particular form of columns of the coefficient matrix α. To summarize the above construction of EOFs, we will describe the following preposition [27] for N-dimensional variable X.

Proposition 2.1.1. *Let eigenvectors $e_1, \ldots, e_N$ be EOFs of X with mean zero, corresponding to the eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$ of covariance matrix C, then the first k EOFs $e_1, \ldots, e_k$, where $k = 1, \ldots, N$, minimize the error $\epsilon_k = \left\langle \|X - \sum_{i=1}^{k}(X \cdot e_i)e_i\|_2^2 \right\rangle$ and the minimum mean squared error $\epsilon_k = \text{Var}X - \sum_{i=1}^{k} \lambda_i$ for the corresponding first k eigenvalues of C.*

Definition 2.1.2. *Suppose that $\{e_1, \ldots, e_N\}$ are EOFs of X. Then $\alpha_i = (X \cdot e_i), i = 1, \ldots, N$ are called EOF coefficients. Sometimes, the EOF coefficients are also called principal components [27].*

Next preposition [27] concludes that the variance contribution of any ith component to the variance is just λ_i in the analysis of empirical orthogonal functions.

Proposition 2.1.3. *If $\alpha_1, \ldots, \alpha_N$, where $\alpha_i, i = 1, \ldots, N$, is one-dimensional variables with mean zero, are EOF coefficients of X. Then $Var\ \alpha_i = \lambda_i, i = 1, \ldots, N$, where λ_i is the ith eigenvalue corresponding to the ith EOF, and covariance of the distinct EOF coefficients is zero.*

Remark 2.1.4. *If EOFs $\{e_1, \ldots, e_N\}$ are orthonormal basis of $\mathbf{R}^N$, then $\|X - \sum_{i=1}^{N}(X \cdot e_i)e_i\|_2^2 = 0$ and preposition (2.1.1) implies that total variance $\text{Var } X = \sum_{i=1}^{N} \text{Var}(X_i) = \sum_{i=1}^{N} \lambda_i$. Thus the following fraction of variance [20] gives the percentage distribution of nth EOF mode:*

$$\frac{\text{Var}(X_n)}{\text{Var}X} = \frac{\lambda_n}{\sum_{j=1}^{N} \lambda_j}. \tag{2.15}$$

Moreover, for $K < N$, consider the truncated EOF expansion $\tilde{X} = \sum_{i=1}^{K} \alpha_i e_i$ which is the optimal $K-$dimensional approximation to $N-$dimensional X if mean squared error between X and $\tilde{X}$

is minimal. In this case, the fraction of variance, given in (2.15)*, which is accounted for by* $K-$*dimensional approximation becomes*

$$\frac{\sum_{k=1}^{K} \lambda_k}{\sum_{j=1}^{N} \lambda_j}.$$

2.2 Diffusion maps

A diffusion map (DM) is a non-linear dimensionality reduction method. It is based on describing a Markov random walk on the graph of the data [5, 25], for the analysis of the geometry of the data. As is known, in the random walk, each jump has a probability (weight) associated with it. Then a connection between two data points is described by the probability of jumping from one point to another point in one step. To perform the random walk on our data set for a number of time steps, we need a measure for the closeness or nearness of the data points. Thus we define diffusion distance, see Section 2.2.3, by using this measure.

In general, diffusion maps framework is used to change the representation of data sets that involve a large number of variables in the original space into a lower dimensional data structure with a small number of parameters based on the underlying geometric structure of the specific data set. The nonlinearity of the map means that there is no explicit relationship between the higher-dimensional space (original space) and the reduced dimensional space (diffusion space). Also, this is a distance-preserving map which means that a diffusion map transforms data from the original space to diffusion space, such that Euclidean distance between data points in diffusion space approximates the diffusion distance between points in the original space. In other words, if the points are close in lower-dimensional space, then they are also close in the original space and vice versa. Now we will study the diffusion map process [5, 7, 17] in the following steps:

2.2.1 Diffusion Kernel function on a data set

We first define a data set $\mathcal{X}$ that contains the data points, i.e. the columns of the matrix $X \in \mathbb{R}^{N\times M}$. In order to determine the connection between the data points of the data set, we define a kernel, i.e. a real valued and non-negative symmetric function. We consider, for instance, the Gaussian kernel map $\mathcal{K} : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$ on the columns of original data set $X \in \mathbb{R}^{N\times M}$, defined by

$$\mathcal{K}(x,y) = e^{-(\frac{||x-y||^2}{\epsilon})}, \tag{2.16}$$

for some parameter ϵ, where $\epsilon > 0$ and $\|.\|$ is a L_2 norm. The choice of ϵ is not trivial and it can be chosen by different ways [2].

From (2.16), we notice that $\mathcal{K}(x,y) \gg 0$ if data points x and y are close from each other (useful information of their distance) and it becomes negligible if points are far apart (irrelevant information of their distance). So, it provides the relationship between pairs of points in the data set X. In fact, a neighborhood is defined at each point of the set X. For instance, a neighborhood of a point x is the set of all points y which are connected to x and x and y are connected if $\mathcal{K}(x,y) > \tau$ for some cutoff parameter $\tau > 0$. Thus the kernel function $\mathcal{K}$ forms a positive semi-definite distance matrix $\mathbf{K} \in \mathbb{R}^{M\times M}$ such that $\mathbf{K}_{ij} = \mathcal{K}(x_i, x_j)$, whose each row of which is obtained by measuring distance of one data point with all other points. Thus, the number of rows of the matrix depends on the number of data points. This implies that there exist real-valued eigenvectors and non-negative eigenvalues for the matrix $\mathbf{K}$.

2.2.2 Diffusion matrix and diffusion process

Next, we can normalize the rows of such a given matrix $\mathbf{K}$ by just dividing the degree of each node. To obtain the degree, we form a vector by taking the sum of each row of the distance matrix $\mathbf{K}$ and form the diagonal matrix $\mathbf{D}$ of this vector. So, we set $\mathbf{P} = \mathbf{D}^{-1}\mathbf{K} \in \mathbb{R}^{M\times M}$, with $\mathbf{P_{ij}} = p(x_i, x_j)$, which is normalized diffusion matrix. Also, the diffusion matrix $\mathbf{P}$ can be expressed as Markov transition matrix [21] such that

$$p(x,y) = \frac{\mathcal{K}(x,y)}{\alpha(x)} \tag{2.17}$$

where α is the normalization constant and given by

$$\alpha(x) = \sum_{y\in\mathcal{X}} \mathcal{K}(x,y)$$

Thus, (2.17) shows that the transition probability $p(x, y)$ between the points x and y is proportional to $\mathcal{K}(x, y)$. If the data points are not sampled uniformaly [23] on the manifold, then normalized kernel $\frac{\mathcal{K}(x,y)}{\alpha(x)\alpha(y)}$ can be used instead of $\mathcal{K}(x, y)$.
Thus, **P** forms a new kernel which preserves the positivity, but it is not symmetric. Lemma 1 [7] depicts that a symmetric matrix $\mathbf{P}'$ can be derived from **P** as

$$\mathbf{P}' = \mathbf{D}^{1/2}\mathbf{P}\mathbf{D}^{-1/2}$$

P and $\mathbf{P}'$ have the same eigenvalues and,

$$\psi_k = \varphi_k \mathbf{D}^{1/2}, \quad \phi_k = \varphi_k \mathbf{D}^{-1/2}$$

where ψ_k and ϕ_k are left and right eigenvectors of **P** respectively and φ_k are the eigenvectors of $\mathbf{P}'$. This matrix **P** gives the probabilities for one time step in a random walk from point i to point j and also contains the information about the geometry of the data set.

2.2.3 Diffusion Distance

The diffusion distance metric is related to the diffusion matrix and is given by:

$$\begin{aligned}\mathbf{D}_t(x, y)^2 &= \sum_{u \in X} |p_t(x, u) - p_t(y, u)|^2 \\ &= \sum_k |\mathbf{P}^t_{ik} - \mathbf{P}^t_{kj}|^2\end{aligned} \tag{2.18}$$

where $p_t(x, .)$ and $p_t(y, .)$ describe the probability vectors defined by the transistion matrix **P** and $\mathbf{P}^t_{ik}$ is the (i, k)-element of $\mathbf{P}^t$, t denotes the (integer) power of the matrix **P**.
The paths along the geometric structure are the main contributors to the diffusion distance. The quantity, $D_t(x, y)$ sums over all possible paths of length t connecting x and y. However, the term $p_t(x, y)$ has large values for paths due to the sums of the probabilities of all possible paths of length t between x and y. To remain the diffusion distance small, the quantity $|p_t(x, u) - p_t(y, u)|$ must approach zero. This implies that probabilities between x and u, and y and u are almost equal if x and y are well connected through u. As linear or non-linear data structures follow some geometric structure and diffusion metrics can approximate distances along this geometric structure. So, in new diffusion space, the diffusion distance in data space becomes Euclidean distance for convenience, and mapping of the data points into a Euclidean space becomes easy. Moreover, from [14], diffusion mapping shows robustness to noise perturbation of the data set and it allows geometric analysis at different time scales due to diffusion distance. Because diffusion distance sums over all paths joining two points and gives a smoothing effect on perturbations if there are small perturbations of the data set.

2.2.4 Diffusion map from the high dimensional space to the lower dimensional space

Lastly, we will talk about the diffusion map which maps coordinates between the original data space and diffusion space. Therefore, we compute the eigenvalues and corresponding eigenvectors of **P** for defining the diffusion map. Since the eigenvalue, $\lambda_0 = 1$ of **P** by the lemma 2 [14] and all eigenvalues are positive and non-increasing by the positivity of the distance matrix **K**. Hence the eigenspectrum of **P** is contained in $[0, 1]$. Now **P** has a discrete sequence of eigenvalues and corresponding eigenfunctions (λ_0, ψ_0), $(\lambda_1, \psi_1)\ldots$ and we sort them in a way such that $1 = \lambda_0 > |\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n| \geq 0$ and $P\psi_i = \lambda_i\psi_i, i = 0, 1, 2, \ldots$. Moreover these eigenvectors form an orthonormal basis in L^2 space. In fact, diffusion distances can be interpreted in terms of dominant eigenvectors and their corresponding eigenvalues of **P** because eigenvectors associated with the largest eigenvalues decay slowly in the sense of diffusion and they represent the dynamic of the data set on a long-time scale. Then diffusion map $\mathbf{\Psi}_t : \mathbb{R}^N \to \mathbb{R}^s$, in t steps, which embeds the data set $\mathcal{X} \subseteq \mathbb{R}^N$ into a Euclidean space $\mathbb{R}$ of s dimension, $s < N$, and is given by scaled eigenvectors such that

$$\mathbf{\Psi}_t(X_i) = \begin{pmatrix} \lambda_1^t \psi_1(X_i) \\ \lambda_2^t \psi_2(X_i) \\ \vdots \\ \lambda_s^t \psi_s(X_i) \end{pmatrix} \tag{2.19}$$

where $X_i \in \mathbb{R}^N,\ i = 1, \ldots, M$ and $\lambda_1, \lambda_2, \ldots, \lambda_s$ are the eigenvalues of matrix $\mathbf{P}$ corresponding to the eigenvectors $\psi_1, \psi_2, \ldots, \psi_s$. Since $\lambda_0 = 1$, so its corresponding trivial eigenvector $\psi_0 = \mathbf{1}$ does not provide information about the data. Therefore, it was not added in the representation given in (2.19). So, the diffusion map is used to find the fewer coordinates to represent data points in the diffusion space. Thus, the Euclidean distance between any two data points, for instance, X_i and X_j, in the new space is defined as

$$\|\mathbf{\Psi}_t(X_i) - \mathbf{\Psi}_t(X_j)\|^2 = \sum_{k=1}^{s} \lambda_s^{2t} (\psi_k(X_i) - \psi_k(X_j))^2 \tag{2.20}$$

By using the definition of diffusion distance in (2.18) and from the proposition 2.2.1, we obtain,

$$\begin{aligned} \|\mathbf{\Psi}_t(X_i) - \mathbf{\Psi}_t(X_j)\|^2 &= \sum_{k=1}^{s} \lambda_s^{2t} (\psi_k(X_i) - \psi_k(X_j))^2 \\ &= \|p_t(X_i, u) - p_t(X_j, u)\|^2 \\ &= \mathbf{D}_t(X_i, X_j)^2 \end{aligned} \tag{2.21}$$

Hence the diffusion map encapsulates the data according to diffusion metric in reduce dimensional space and the Euclidean distance between two mapped points in reduced dimensional space is equal to the diffusion L_2 distance between data points in the original space.

The aim is to find fewer coordinates for the new representation of data points in the diffusion space. Since the eigenvalues, $\lambda_1, \lambda_2, \ldots$ of $\mathbf{P}$ approach zero, and their modulus is strictly less than 1. Therefore, the sum in the definition of diffusion distance defined in (2.21) can be computed for the desired accuracy level $\delta > 0$ with a finite number of terms such that

$$s(\delta, t) = \max\{\ell : |\lambda_\ell|^t > \delta |\lambda_1|^t\}$$

Thus, the diffusion distance in (2.20) becomes

$$\|\mathbf{\Psi}_t(X_i) - \mathbf{\Psi}_t(X_j)\|^2 = \sum_{k=1}^{s(\delta,t)} \lambda_s^{2t} (\psi_k(X_i) - \psi_k(X_j))^2 \tag{2.22}$$

and new $s(\delta, t)$-dimensional diffusion mapping $\mathbf{\Psi}_t : \mathbb{R}^N \to \mathbb{R}^{s(\delta,t)}$ is defined as

$$\mathbf{\Psi}_t(X_i) = \begin{pmatrix} \lambda_1^t \psi_1(X_i) \\ \lambda_2^t \psi_2(X_i) \\ \vdots \\ \lambda_{s(\delta,t)}^t \psi_{s(\delta,t)}(X_i) \end{pmatrix} \tag{2.23}$$

where $X_i \in \mathbb{R}^N,\ i = 1, \ldots, M$, t is the number of steps and $\lambda_1, \lambda_2, \ldots, \lambda_{s(\delta,t)}$ are the eigenvalues of matrix $\mathbf{P}$ corresponding to the eigenvectors $\psi_1, \psi_2, \ldots, \psi_{s(\delta,t)}$.

The two main benefits of diffusion mapping make it convincing over other algorithms. Since it is nonlinear and preserves the local structure [2]. Because the input data is not linear normally and lies on nonlinear manifolds. In most cases, preserving the distance of the close points is more significant than preserving the distances of the points that are far apart.

Proposition 2.2.1. *If the diffusion map $\mathbf{\Psi}_t$ transforms the data into the Euclidean space $\mathbb{R}^s$, then the Euclidean distance in the diffusion space is equal to the diffusion distance between the points in the data set $\mathcal{X}$.*

2.2.5 The Diffusion Mapping Algorithm

1. Consider a data set $\mathcal{X}$ in high dimensional space as the input.
2. Define a kernel $\mathcal{K}(x, y)$ to determine the distance between data points x and y. And we establish a symmetric distance matrix $\mathbf{K}_{ij} = \mathcal{K}(x_i, x_j)$.
3. We create the normalized diffusion matrix $\mathbf{P}$ by the normalization of the rows of the distance matrix.
4. Compute the eigenvectors of the diffusion matrix $\mathbf{P}$.
5. Lastly, map the data from high dimensional data set $\mathcal{X}$ to lower dimensional data set by using the dominant eigenvectors of $\mathbf{P}$ in the definition, given in (2.23), of the diffusion map at time t.

2.3 Extended dynamic mode decomposition (EDMD)

We can associate every dynamical system with a Koopman operator which encodes the significant properties of the system. The Koopman operator describes the evolution of the observables which are the functions of the state of the given system. Also, the Koopman operator framework on the dynamical systems is useful in some ways because the Koopman dynamics are linear and infinite dimensional even when the dynamical system is nonlinear. EDMD algorithm is a popular numerical method that is used to approximate the spectral properties of an infinite dimensional Koopman operator. Here we will focus on the connection between the Koopman operator and EDMD model [16, 26, 4].

2.3.1 The Koopman operator

We first consider discrete-time nonlinear dynamical system $(\mathcal{M}, n, \mathbf{F})$, where the state space $\mathcal{M} \subset \mathbb{R}^N$, discrete-time $n \in \mathbb{Z}$ and $\mathbf{F} : \mathcal{M} \to \mathcal{M}$ is the evolution operator.
Since the Koopman operator [16] acts on the evolution of observables represented by functions of state space $\mathcal{M}$ in the function space $\mathcal{F}$. Here we consider $\mathcal{F} = L^2(\mathcal{M}, \rho)$ the Hilbert space, such that

$$L^2(\mathcal{M}, \rho) = \{g : \mathcal{M} \to \mathbb{C} : \|g\|_{\mathbb{L}^2(\mathcal{M},\rho)} < \infty\}$$

where

$$\|g\|_{\mathbb{L}^2(\mathcal{M},\rho)} = \int_{\mathcal{M}} |g(x)|^2 \, \rho(dx).$$

where ρ is a positive, single-valued analytic function defined on $\mathcal{M}$. Here we assume that g is full state observable such that $g(x) = x$.
The linear Koopman operator [19] $\mathcal{K} : \mathcal{F} \to \mathcal{F}$ associated with the map $\mathbf{F} : \mathcal{M} \to \mathcal{M}$ is defined as:

$$\mathcal{K}g = g \circ \mathbf{F} \tag{2.24}$$

Thus, the Koopman operator defines a new linear dynamical system $(\mathcal{F}, n, \mathcal{K})$ which governs the evolution of each observable value. In other words, the new linear dynamical system is defined by the evolution of observables.
Notably, a nonlinear dynamical system in finite dimensional space is representable by an equivalent linear dynamical system in infinite dimensional space. We obtain a linear approximation of a nonlinear system and also trade linearity for the dimensionality as the Koopman operator $\mathcal{K}$ is infinite-dimensional. It is also noticeable in (2.24), for a given state x, $(\mathcal{K}g)(x)$ is computed by applying $\mathcal{K}$ to the observable g and then evaluate $\mathcal{K}g$ at x or we can apply F to x and then evaluate g at this updated position. In fact, $\mathbf{F}$ and $\mathcal{K}$ act on different spaces, but they show the same fundamental dynamics.
Moreover, ρ and $\mathcal{M}$ are chosen in a way such that $\mathcal{K}$ is bounded operator. Now we can study the spectral analysis of a bounded linear operator $\mathcal{K}$. Additionally, the Koopman operator has only a discrete spectrum [3] and its eigenfunctions $\{\Phi\}_{i=1}^{\infty}$ span the space of observables. Although each i-th component g_i of the vector-valued observable g is a scalar valued observable. And we assume that this scalar valued observable g_i is in the span of the set of K eigenfunctions which do not form a basis for $\mathcal{F}$. So, we will describe the Koopman mode decomposition for a given vector-valued observable g,

$$x = g(x) = \sum_{k=1}^{K} \nu_k \phi_k(x) \tag{2.25}$$

where K is infinite, Φ_k is the k-th Koopman eigenfunction corresponding to the eigenvalue μ_k and the vectors ν_k are the Koopman modes associated with the observable g and the k-th eigenfunction.
By applying $\mathcal{K}$ on both sides of (2.25) and the linearity of $\mathcal{K}$ and (2.24) give the following identity,

$$\mathbf{F}(x) = \mathcal{K}g(x) = \sum_{k=1}^{K} \mu_k \nu_k \phi_k(x) \tag{2.26}$$

The above expression indicates that the dynamics associated with each of the eigenfunctions are described by its corresponding eigenvalue.

Theoretically, the linear Koopman dynamics is easy to do analysis, but it might be challenging to compute its spectral properties owing to its infinite dimensionality. Then we will briefly describe the EDMD method to compute the Koopman mode decomposition.

2.3.2 The EDMD algorithm

In this case, the EDMD algorithm [16] is used to approximate the finite-dimensional representation of the Koopman operator $\mathcal{K}$ in the form of a linear finite-dimensional map $\mathbf{K}$. Thus, the spectral properties of the Koopman operator $\mathcal{K}$ will be approximated by the spectral properties of $\mathbf{K}$. The EDMD algorithm [26] requires, firstly, a pair of data sets at different time steps

$$\mathbf{X} = [x_1, x_2, \dots, x_M]$$

and

$$\mathbf{Y} = [y_1, y_2, \dots, y_M]$$

where $x_i, y_i \in \mathcal{M}$ are the snapshots of the system state with $y_i = \mathbf{F}(x_i)$. Also, $y_i = \mathbf{F}(x_i)$ is the snapshot at the next time step for a given snapshot x_i, and secondly, we pick a dictionary of observables

$$\Psi = \{\Psi_1, \Psi_2, \dots, \Psi_K\}$$

where each $\Psi_i \in \mathcal{F}$, $i = 1, \dots, K$. Now, we assume the span $\mathbf{U}$ of Ψ such that $\mathbf{U} = \{a^T \Psi : a \in \mathbb{R}^K\}$ which is a linear subspace of $\mathcal{F}$. The optimal choice of the set of dictionary remains an open question. In this case, we assume that the dictionary of observables is rich enough to approximate a few of the leading Koopman eigenfunctions accurately.

The main idea is to form a finite-dimensional approximation $\mathbf{K} \in \mathbb{C}^{K \times K}$ of the Koopman operator $\mathcal{K}$. For any $\Phi \in \mathbf{U}$ such that $\Phi = \sum_{k=1}^{K} a_k \Psi_k = a^T \Psi$, we can have this,

$$\mathcal{K}\Phi = a^T \mathcal{K}\Psi = a^T \Psi \circ \mathbf{F}$$

If $\mathbf{U}$ is invariant subspace under the action of the Koopman operator, then $\mathcal{K}\Phi = b^T \Psi$, for some $b \in \mathbb{R}^K$. Then, a finite-dimensional representation of Koopman operator $\mathcal{K}$ is defined as the matrix $\mathbf{K} \in \mathbb{C}^{K \times K}$ with $b^T = a^T \mathbf{K}$. Thus, for all a, we obtain $\mathbf{K}\Psi = \Psi \circ \mathbf{F}$. In order to find the matrix $\mathbf{K}$ [16], we will use the snapshots of the two data sets at different time steps defined previously with the condition $y_i = F(x_i)$ and solve the minimization problem,

$$\mathbf{K} = \overset{\text{argmin}}{\tilde{K}} \in \mathbb{C}^{k \times K} \; J(\tilde{K}) = \sum_{m=1}^{M} \|\Psi(y_m) - \tilde{K}\Psi(x_m)\|^2$$

If $\mathbf{U}$ is invariant under $\mathcal{K}$, then residual $J(K)$ is zero. Otherwise, for $J(K) > 0$, the previous procedure seeks to find the K which minimizes the residual. The unique solution to the above identity is,

$$\mathbf{K} = G^+ A \tag{2.27}$$

with

$$G = \frac{1}{M} \sum_{m=1}^{M} \Psi(x_m)^* \Psi(x_m)$$

$$A = \frac{1}{M} \sum_{m=1}^{M} \Psi(x_m)^* \Psi(y_m)$$

where $+$ denotes the pseudoinverse and $\mathbf{K}, \mathbf{G}, \mathbf{A} \in \mathbb{C}^{K \times K}$. Consequently, the eigenvalues of $\mathbf{K}$ are the EDMD approximations of the eigenvalues of the Koopman operator. If ξ_j is the j-th right eigenvector of $\mathbf{K}$ corresponding to the eigenvalue μ_j, then the EDMD approximation of the

Koopman eigenfunction is $\Phi_j = \xi_j^T \Psi$ with the same eigenvalue μ_j. And for any observable vector $g = B\Psi$, the j-th Koopman mode associated with observable is defined as $\nu_j = B\alpha_j$, where α_j is the left eigenvector of $\mathbf{K}$.

A drawback of the EDMD algorithm is making a priori choice of the dictionary which becomes a challenge for the general applicability of the EDMD algorithm. The selection of a dictionary impacts the approximation of the spectral properties of the dynamical system. To overcome this issue, this paper [16] provides the development of an iterative approximation algorithm that couples the EDMD with a trainable dictionary that is defined by an artificial neural network.

2.4 Reduced-Order Modeling analysis based on Approximated Lax pairs (ALP)

In order to study the ALP method [9, 10], we first take a general form of an evolution PDE in a bounded domain Ω of $\mathbb{R}^d,\ d \geq 1$, given as

$$\partial_t u = F(u), \tag{2.28}$$

with an initial condition $u(x,0) = u_0(x),\ x \in \Omega$. Here $F(u)$ is a function of u and its derivatives with respect to $x_1, x_2, \dots, x_d$. We look for the solution

$$u(x,t) = \sum_{j=1}^{N} \beta_j(t)\phi_j(x,t)$$

where $\phi_j(x,t)$ is the time-dependent basis. The next task is to compute this reduced order time-dependent basis. For this, we will introduce approximated Lax pairs (ALP). The term Lax pairs refer to a pair of two linear operators $\mathcal{L}(u)$ and $\mathcal{M}(u)$ depending the solution $u(x,t)$, where $x \in \Omega$ is the spatial variable of PDE at time t. The main idea for this method is to use the eigenfunctions $\Phi_1, \Phi_2, \dots$ of the operator $\mathcal{L}(u)$. These eigenfunctions form a time-dependent basis $\Phi_m(x,t)$, where $m > 0$, which is used to approximate the solutions of PDE. In fact, they form a complete Hilbert basis of Hilbert space $\mathrm{L}^2(\Omega)$ with the inner product $\langle \Phi, \Psi \rangle$. Moreover, we assume that the operator $\mathcal{L}$ is self-adjoint such that $\langle \mathcal{L}\Phi, \Psi \rangle = \langle \Phi, \mathcal{L}\Psi \rangle$ for all $\Phi,\ \Psi \in L^2(\Omega)$.
Next, we consider spectral equation for $\mathcal{L}$,

$$\mathcal{L}\Phi_m = \lambda_m \Phi_m \tag{2.29}$$

where $\lambda_m = \lambda_m(t),\ m > 0$, are the eigenvalues, such that $\lambda_m(t) \to +\infty$ as $m \to \infty$, corresponding to the eigenfunctions $\Phi_m(x,t)$. To propagate the eigenmodes in time, we assume an orthogonal evolution operator $Q(t)$, such that $Q^T Q = QQ^T = I$, which satisfies the following condition,

$$\Phi(t) = Q(t)\Phi(0),\ \forall \Phi \tag{2.30}$$

By differentiating the above equation with respect to t and use $\Phi(0) = Q^T(t)\phi(t)$, we have

$$\begin{aligned} \Phi_t &= Q_t \Phi(0) \\ &= Q_t Q^T \Phi \end{aligned} \tag{2.31}$$

We define the other linear operator $\mathcal{M}(t) = Q_t(t)Q^T(t)$, then (2.31) becomes,

$$\Phi_t(t) = \mathcal{M}(t)\phi(t) \tag{2.32}$$

$\mathcal{M}(t)$ is skew-symmetric as $\mathcal{M}^T = QQ_t^T = -Q_t Q^T = -\mathcal{M}$. We differentiate (2.29) with respect to t, use the equation (2.32) and obtain,

$$\begin{aligned} \mathcal{L}_t\Phi_m + \mathcal{L}(\Phi_m)_t &= (\lambda_m)_t\Phi_m + \lambda_m(\Phi_m)_t & (2.33)\\ \Rightarrow \mathcal{L}_t\Phi_m + \mathcal{L}\mathcal{M}\Phi_m &= (\lambda_m)_t\Phi_m + \lambda_m\mathcal{M}\Phi_m \\ &= (\lambda_m)_t\Phi_m + \mathcal{M}(\lambda_m\Phi_m) \\ &= (\lambda_m)_t\Phi_m + \mathcal{M}\mathcal{L}\Phi_m \\ \Rightarrow (\mathcal{L}_t + \mathcal{L}\mathcal{M} - \mathcal{M}\mathcal{L})\Phi_m &= (\lambda_m)_t\Phi_m & (2.34) \end{aligned}$$

To solve the equation (2.34) for non-trivial eigenfunctions $\Phi_m(x,t)$, we have $\mathcal{L}_t + \mathcal{L}\mathcal{M} - \mathcal{M}\mathcal{L} = 0$. We define the commutator $[\mathcal{L}, \mathcal{M}] = \mathcal{L}\mathcal{M} - \mathcal{M}\mathcal{L}$ of two operators $\mathcal{L}$ and $\mathcal{M}$, then we obtain the following Lax equation:

$$\mathcal{L}_t + [\mathcal{L}, \mathcal{M}] = 0 \tag{2.35}$$

Now we take the equation (2.33) and use the Lax equation for $\mathcal{L}_t$ to obtain

$$\begin{aligned}(\lambda_m)_t\Phi_m &= (\mathcal{L}-\lambda_m)(\Phi_m)_t + (\mathcal{M}\mathcal{L}-\mathcal{L}\mathcal{M})\Phi_m \\ &= (\mathcal{L}-\lambda_m)((\Phi_m)_t - \mathcal{M}\Phi_m) \qquad (2.36)\end{aligned}$$

We take the inner product of Φ_m on both sides of the above equation and use the fact that $\mathcal{L}-\lambda_m$ is self-adjoint,

$$\langle \Phi_m, (\lambda_m)_t\Phi_m\rangle = \langle (\mathcal{L}-\lambda_m)\Phi_m, (\Phi_m)_t - M\Phi_m\rangle.$$

Since λ_m is an eigenvalue corresponding to the eigenfunction Φ_m of $\mathcal{L}$, then above equation implies

$$\langle \Phi_m, \Phi_m\rangle(\lambda_m)_t = \langle 0, (\Phi_m)_t - M\Phi_m\rangle$$

$$\Rightarrow \langle \Phi_m, \Phi_m\rangle(\lambda_m)_t = 0$$

For non-trivial eigenfunctions, the above identity implies that we have isospectral evolution for any m,

$$(\lambda_m)_t = 0 \qquad (2.37)$$

This means that eigenvalues λ_m, $m > 0$, are independent of time and they are constant. It is concluded that if an integrable evolution PDE can be expressed as the Lax equation with (2.32) and (2.37), then eigenfunctions of operator $\mathcal{L}$ can be used to reconstruct the solution u of PDE at every time step. Next, we consider the linear Schrödinger operator $\mathcal{L}_\chi$ associated with a real number $\chi > 0$,

$$\mathcal{L}_\chi(u)\Phi = -\Delta\Phi - \chi u\Phi \qquad (2.38)$$

where $u(x)$ is the solution of PDE and is called a real potential function and the Laplacian Δ in d dimensions.

First, we will study the ALP algorithm from [10]. The approximation of the solution u of PDE in the reduced basis,

$$u \approx \tilde{u} = \sum_{m=1}^{N_M}\beta_m\Phi_m, \quad F(u) \approx \tilde{F}(u) = \sum_{m=1}^{N_M}\gamma_m\Phi_m$$

Then the given PDE $\partial_t u = F(u)$,

$$\Rightarrow \sum_{m=1}^{N_M}\dot{\beta}_m\Phi_m + \beta_m\partial_t\Phi_m = \sum_{m=1}^{N_M}\gamma_m\Phi_m$$

.

$$\Rightarrow \dot{\beta} + M\beta = \gamma$$

The proposition 2.4.1 explains that approximation M of $\mathcal{M}(u)$ can be computed in the space defined by the eigenfunctions of $\mathcal{L}_\chi(u)$ and the evolution equation of the eigenvalues of $\mathcal{L}_\chi(u)$ can be driven. We define $\Theta_{ij} = \langle \tilde{F}(u)\Phi_j, \Phi_i\rangle$. Then the reduced order approximation of the Lax equation, (2.41) and (2.43) is given by

$$\frac{d\Lambda}{dt} + \chi\Theta = \Lambda M - M\Lambda,$$

$$\dot{\lambda}_m = -\chi\Theta_{mm},$$

and

$$M_{mp}(u) = \frac{\chi}{\lambda_p - \lambda_m}\Theta_{mp}, \text{ for } \lambda_m \neq \lambda_p$$

respectively. Now representation of the multiplication by $\tilde{F}(u)$ is defined as

$$\Theta_{ij} = \langle \tilde{F}(u)\Phi_j, \Phi_i\rangle = \sum_{k=1}^{N_M}\gamma_k T_{ijk}$$

where $T_{ijk} = \langle \Phi_k \Phi_j, \Phi_i \rangle$ is the third order tensor. We compute the time derivative of third order tensor T_{ijk} and given as

$$\dot{T}_{ijk} = \langle \partial_t \phi_k \phi_j, \phi_i \rangle + \langle \phi_k \partial_t \phi_j, \phi_i \rangle + \langle \phi_k \phi_j, \partial_t \phi_i \rangle$$

Then

$$\dot{T}_{ijk} = \{M, T\}^{(3)}_{ijk}$$

where $\{M, T\}^{(3)}_{ijk} = \sum_{l=1}^{N_M} (M_{li} T_{ljk} + M_{lj} T_{ilk} + M_{lk} T_{ijl})$. In order to summarize the above procedure, we have a system of equations which demonstrates the dynamics in the reduced order space:

- Reduced order equation $\dot{\beta} + M\beta = \gamma$
- Representation of $F(u)$, $\Theta_{ij} = \sum_{k=1}^{N_M} \gamma_k T_{ijk}$
- Evolution of the eigenvalues, $\dot{\lambda}_i = -\chi \Theta_{ii}$
- Approximation of Lax operator, $\mathcal{M}$, $M_{ij}(u) = \frac{\chi}{\lambda_j - \lambda_i} \Theta_{ij}$, for $\lambda_i \neq \lambda_j$
- Evolution of the tensor, $\dot{T}_{ijk} = \{M, T\}^{(3)}_{ijk}$
- Relation between β and γ, $\gamma_i = \gamma_i(\beta)$

where $i, j, k = 1, \ldots, N_M$. One of the limitations of the above ALP algorithm is disappointing speed-up. Because we need to propagate the third order tensor which is quite expensive.

Next, We will study the Lax method by using the inverse scattering method [9]. Figure 2.1 describes the scattering transform, which is used to solve the nonlinear PDE, for instance, KdV equation.

$$\begin{tikzcd}
u(x,0) \arrow[r, "\text{Scattering}"] & \tilde{u}(0) \arrow[d, "\text{Time evolution}"] \\
u(x,t) \arrow[u, dashed, "\text{PDE}"] & \tilde{u}(t) \arrow[l, "\text{Inverse Scattering}"']
\end{tikzcd}$$

Figure 2.1: It describes the inverse scattering transform for the PDE. Source: [8]

From (2.38), the bounded solution of the equation $\Delta\Phi + \chi u \Phi = 0$ implies that if $u(x) > 0,\ x \in \mathbb{R}$, then there is a finite number of negative (discrete) eigenvalues. Now, we find the eigenfunctions of the operator $\mathcal{L}_\chi(u_0)$, where u_0 is the initial condition of the given PDE. In this case, there are two types of eigenfunctions corresponding to the discrete spectrum of negative eigenvalues and the continuous spectrum of positive eigenvalues. We first define $\kappa_m = \sqrt{-\lambda_m},\ m = 1, 2, \ldots$ for the negative eigenvalues λ_m and order them in this way $\kappa_1 > \kappa_2 > \ldots$.

To characterize the behaviour of the bounded solution for $x \to +\infty$, eigenfunctions $\Phi_m, m = 1, 2, \ldots$ can be written as,

$$\Phi_m = c_m e^{-\kappa_m x} \tag{2.39}$$

where fixed real constant c_m is obtained by normalizing the mth eigenfunction in $L^2(\Omega)$ such that $\int_\Omega \Phi_m^2 dx = 1$. Thus (2.39) means that eigenfunctions decay exponentially rapidly for negative eigenvalues as $x \to +\infty$. This implies that there exists a finite number of negative eigenvalues.

Thus, the finite-dimensional subspace V_m of $L^2(\Omega)$ spanned by eigenfunctions $\Phi^2_{m,h}, m = 1, \ldots, N_-$ corresponding to the negative eigenvalues of $\mathcal{L}(u)$. Here N_- denotes the number of negative eigenvalues. Now approximation $\tilde{u}_0 \in V_m$ of the solution u can be projected on the eigenmodes $\Phi^2_{m,h}$ corresponding to the negative eigenvalues $\lambda_{m,h},\ m = 1, 2, \ldots, N_-$ by using Deift-Trubowitz formula from [9]

$$\tilde{u}_0(x) = \frac{1}{\chi} \sum_{m=1}^{N_-} \kappa_{m,h} \Phi^2_{m,h} \tag{2.40}$$

where $\kappa_{m,h} = \sqrt{-\lambda_{m,h}},\ m = 1, 2, \ldots, N_-$. For large values of $\chi > 0$, the more negative eigenvalues exist, the approximation is more accurate. Thus χ is chosen in such a way that the error between the exact solution u and approximating solution $\tilde{u}$ is minimal in L^2-norm.
The next proposition from [9] gives an approximation of $\mathcal{M}(u)$ in the space $L^2(\Omega)$ spanned by the eigenfunctions of $\mathcal{L}(u)$ and describes how to derive an evolution equation satisfied by the eigenvalues for any arbitrary PDE for which the construction of both operators $\mathcal{L}$ and $\mathcal{M}$ satisfies (2.29) and (2.32) even if $\mathcal{M}$ is not defined explicitly. Also, the eigenfunctions, used to approximate the solution $u(t)$, of $\mathcal{L}(u(t))$ are denoted by $(\Phi_m(t))_{m=1 \ldots N_-}$ and the eigenfunctions, which are used to approximate the operator $\mathcal{M}(u(t))$ or the evolution equation of the eigenvalues, of $\mathcal{L}(u(t))$ are denoted by $(\Psi_m(t))_{m=1 \ldots N_M}$.

Proposition 2.4.1. *Let u be a solution of PDE given in (2.28) and let $\mathcal{L}_\chi(u)$ defined in (2.38) for a real number $\chi > 0$. Also suppose $N_M \in \mathbb{N}$ and for $m \in \{1, \ldots, N_M\}$, $\lambda_m(t)$ is an eigenvalue of $\mathcal{L}_\chi(u(x,t))$ corresponding to the eigenfunction $\Psi_m(x,t)$ normalized in $L^2(\Omega)$. Suppose that the operator $\mathcal{M}(u)$ defined by $\partial_t \Psi_m = \mathcal{M}(u) \Psi_m$.*
Then the evolution of λ_m is given by

$$\partial_t \lambda_m = -\chi \langle F(u) \Psi_m, \Psi_m \rangle, \tag{2.41}$$

and the evolution of eigenfunction Ψ_m satisfies for $p \in \{1, \ldots, N_M\}$, where N_M denotes the number of modes associated with negative eigenvalues and some positive eigenvalues if it is required,

$$\langle \partial_t \Psi_m, \Psi_p \rangle = \mathbf{M}_{mp}(u) \tag{2.42}$$

where

$$\mathbf{M}(u) = \begin{cases} \mathbf{M}_{mp}(u) = \frac{\chi}{\lambda_p - \lambda_m} \langle F(u) \Psi_m, \Psi_p \rangle, & if\ p \neq m\ and\ \lambda_p \neq \lambda_m, \\ \mathbf{M}_{mp}(u) = 0, & if\ p = m\ or\ \lambda_p = \lambda_m, \end{cases} \tag{2.43}$$

and $\mathbf{M}(u) \in \mathbb{R}^{N_M \times N_M}$ is the skew-symmetric matrix.

Next, we find approximating solution $\tilde{u}_h^{n+1} \in V_h$ for different time steps, $n = 0, 1, \ldots,$ of u. Also the previous proposition provides an approximate way to propagate the eigenfunctions and eigenvalues at time t which are linked to Lax pairs. We will write $\tilde{u}^{n+1}$ by using the form illustrated in (2.40) for known eigenvalues λ_m and eigenmodes Φ_m of $\mathcal{L}_\chi(u)$ for $m = 1, 2, \ldots, N_-$,

$$\tilde{u}_h^{n+1} = \sum_{m=1}^{N_-} \alpha_m^{n+1} (\Phi_{m,h}^{n+1})^2 \tag{2.44}$$

By inserting the above expression in $\langle \mathcal{L}_\chi(\tilde{u}_h^{n+1}) \Phi_m,\ \Phi_p \rangle = \lambda_m \langle \Phi_m, \Phi_p \rangle$ and using the orthonormality of the eigenfunctions, we get the following system of linear equations for α_m^{n+1} at each time step,

$$\sum_{m=1}^{N_-} \alpha_m^{n+1} \langle (\Phi_{m,h}^{n+1})^2, (\Phi_{p,h}^{n+1})^2 \rangle = -\frac{1}{\chi} \left(\lambda_m^{n+1} + \langle \Delta \Phi_{p,h}^{n+1}, \Phi_{p,h}^{n+1} \rangle \right) \tag{2.45}$$

Now we need to get the approximated propagation of eigenvalues $\lambda_{m,h}^{n+1}$ and eigenmodes $\Phi_{m,h}^{n+1}$ at each time step as they are time-dependent and evolve as t changes. As the evolution equation of the eigenvalues is already defined in (2.41) in proposition 2.4.1. So, we apply numerical explicit Euler scheme for evolution equation (2.41), then we obtain,

$$\lambda_{m,h}^{n+1} = \lambda_{m,h}^{n} - \delta t(\chi\langle F(u_h^n)\Psi_{m,h}^n, \Psi_{m,h}^n\rangle) \tag{2.46}$$

where λ^n and $\Psi_{m,h}^n,\ m = 1, \dots, N_M$ are known at time step n. The expression for eigenfunctions, taken from [9], is used to compute the eigenfunctions for the next time steps,

$$\Psi_{m,h}^{n+1} = \sum_{p=1}^{N_M} \left(I + \delta t \mathbf{M}(u_h^n) + \frac{\delta t^2}{2} \mathbf{M}(u_h^n)^2 \right)_{mp} \Psi_{p,h}^n \tag{2.47}$$

where $m, p = 1, \dots, N_M$ and the matrix $\mathbf{M}(u_h^n)$ approximates the operator $\mathcal{M}(u(t^n))$ with the definition in (2.43).

Chapter 3

Numerical comparison of dimensionality reduction methods

So far, we have demonstrated EOF, DM, EDMD, and ALP in the previous chapter. Now, we will study the numerical results obtained by using these methods for two data sets, formed from the approximating solution sequence of the advection equation and the KdV equation. In general, we will use the notation X for the original data set. We will visualize how the methods behave in order to reduce the dimensionality.

3.1 Numerical setup and data

3.1.1 Problem formulation and numerical discretization of the advection equation

Here, we first consider the 1D advection equation (or wave equation) with periodic boundary conditions:

$$u_t + cu_x = 0 \tag{3.1}$$

where $u = u(x,t),\ x \in [0,L],\ t \in [0,T]$, and $c \neq 0$ a constant velocity. The analytical solution $u = u_0(x - ct)$ is determined by the initial condition $u_0(x) = (\frac{1+\cos x}{2})^{100}$ and represents a wave that moves to the positive x-direction if velocity $c > 0$. As we know that PDE can be discretized in terms of ordinary differential equation (ODE) with infinite number of variables. The solution to the PDE is used as the source of data for dimension reduction methods and obtained by using the finite-difference methods to discretize the PDE with respect to $N = 100$ grid points along the spatial variable x at $M = 2N$ time steps. Here we discretize the advection equation by applying the fourier transform for the Matlab code. The numerical method generates the sequence $(u^1, u^2, \ldots, u^M)$ of approximating solutions at M different time steps. The wave solution is just a vector of length N at each time step and the solution is translated at each time step and returns to its initial state due to periodic boundary conditions. Its snapshot is illustrated in Figure 3.1. Each solution approximation is a vector of length N. So, we have M number of data points with N-dimensional coordinates. Hence the original data set $X \in \mathbb{R}^{N \times M}$. Thus X has M vectors $\{X_1, X_2, \ldots, X_M\}$, as the numerical solutions of the advection equation at M steps, of length N.

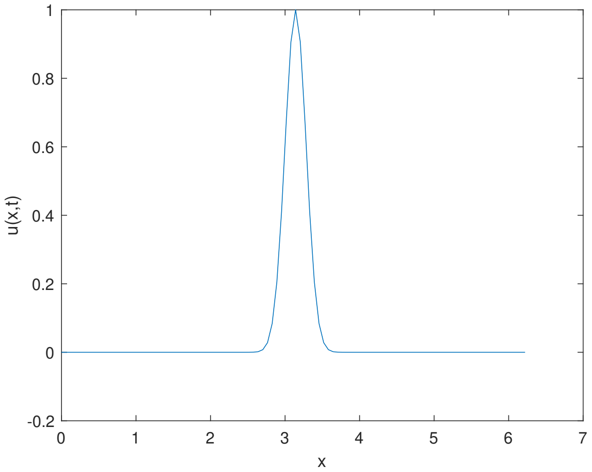


Figure 3.1

3.1.2 Problem formulation and numerical discretization of KdV equation

In this section, we will first mention the well-known nonlinear KdV equation, with periodic boundary conditions:

$$u_t + uu_x - u_{xxx} = 0 \tag{3.2}$$

with $u = u(x,t),\ x \in [0,L],\ t \in [0,T]$ and the given initial condition $u(x,0) = 6\ \mathrm{sech}^2(x - \frac{L}{2})$. Its exact solution $u(x,t) = \frac{c}{2}\mathrm{sech}^2(\frac{\sqrt{c}}{2}(x - ct))$ is known as a soliton which moves at the constant speed c. We discretize KdV-equation by using the scheme based on a classical semi-discretization in space, taken from section 3 of [1], for $N = 200$ interval points with respect to spatial variable x and for $M = 6000$ time steps. We get a sequence $(u^1, u^2, \ldots, u^M)$ of approximating solutions at M different time steps. The snapshot of the numerical solution is displayed in Figure 3.2. Thus, the sequence of approximating solutions of KdV-equation at M time steps form the data $X \in \mathbb{R}^{N \times M}$. Finally, $X = \{X_1, X_2, \ldots, X_M\}$ contains M vectors of length N. These vectors are the approximating solutions of KdV equation at M steps.

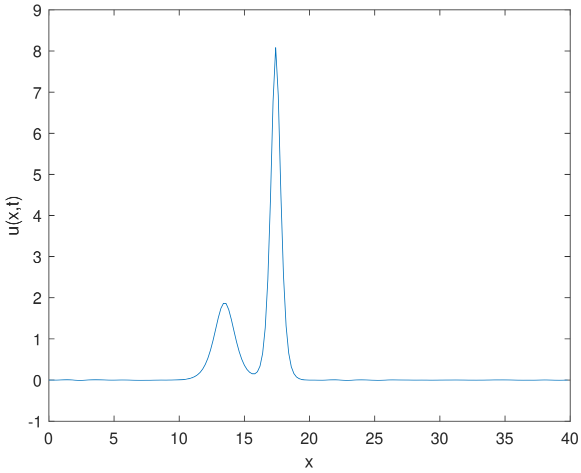


Figure 3.2

3.2 Numerical results with EOFs

First, we will implement EOF and thus we find the covariance matrix C of both data sets, the advection equation data and KdV equation data, respectively and find eigenvectors $e_i, i = 1, \dots, N$, of $C \in \mathbb{R}^{N\times N}$ corresponding to the eigenvalues which are sorted in descending order. Also, in the matrix form, X can be expressed as a linear combination of the EOFs,

$$X = E\alpha \tag{3.3}$$

where basis $E = (e_1, \dots, e_N)$, each $e_i,\ i = 1, \dots, N$ is of length N and the coefficients $\alpha = (\alpha_1, \dots, \alpha_M)$, each $\alpha_j,\ j = 1, \dots, M$ has length N, are chosen in a way such that they are the coefficients of approximation $\tilde{X}$ of X. Thus, for entire time series X, (3.4) implies that $\alpha = E^T X$ as $E = (e_1, \dots, e_N)$ are orthogonal. For some $k \ll N$, the approximation of X is

$$\tilde{X} = \tilde{E}\tilde{\alpha} \tag{3.4}$$

where $\tilde{E} = (e_1, \dots, e_k)$, each $e_i,\ i = 1, \dots, k$ has length N and $\tilde{\alpha} = (\alpha_1, \dots, \alpha_M)$, each $\alpha_j,\ j = 1, \dots, M$ has length k.

Hence the basis set E works as the coordinates. The truncated EOF expansion becomes better if maximum variance is obtained in approximation. As is already mentioned in a proposition 2.1.3, the variance contribution can be obtained by each eigenvalue. If we include more terms in the linear combination, then it results in a closer approximation to the original data set. Also, the eigenvalues are sorted from largest to smallest. The largest eigenvalue means that the associated eigenvector contributes more heavily to the reconstruction of the original data. Therefore, each subsequent term used in the linear combination contributes less and less to the reconstruction of the original data. We have the freedom to choose the lower number of basis. Moreover, there is no clear cutoff point in Figure 3.3 to make a specific selection for the number of basis. Here, we take, for instance $k = 15$, to get the approximation $\tilde{X}$ of X. Figure 3.3 shows that 15 modes give us more or less 10 percent projection error. First 15 EOF modes are implemented for advection equation data in Figure 3.4. To check the quality of the solution, the mean squared error between the original data and approximate data is illustrated in Figure 3.5.

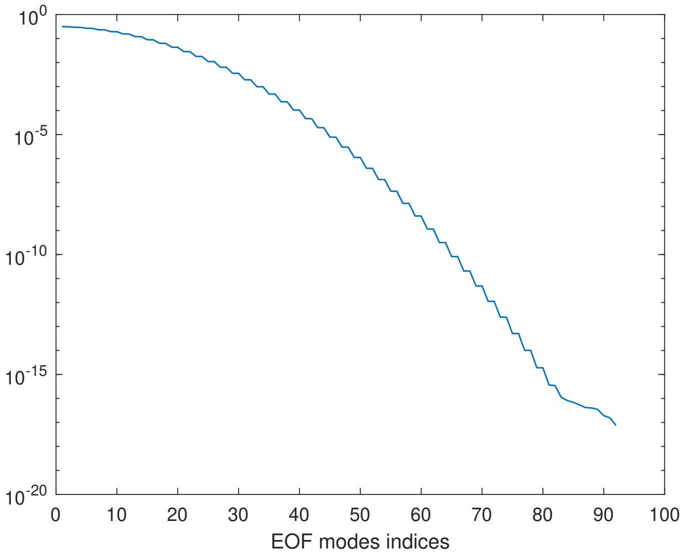


Figure 3.3

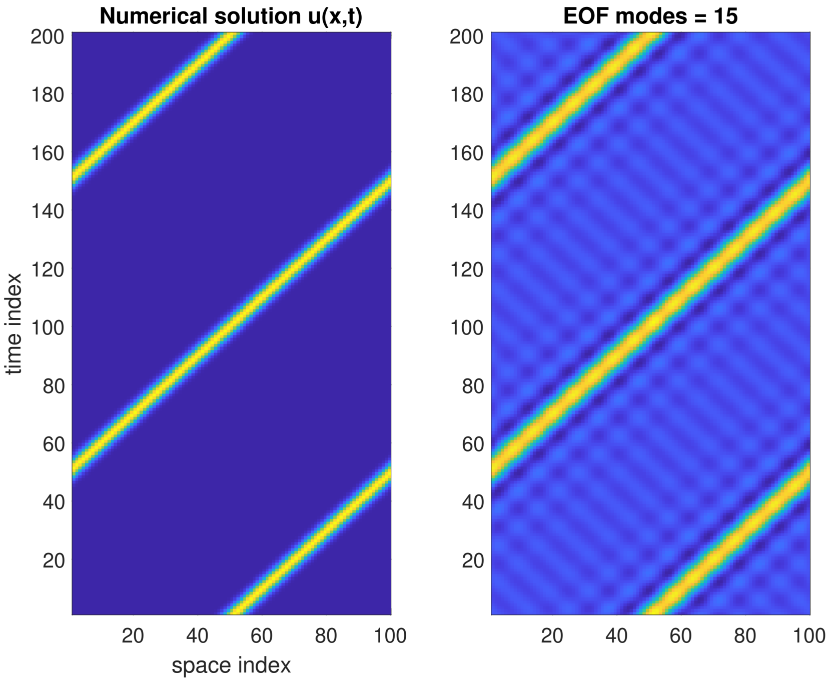


Figure 3.4: Left: the numerical solution $u(x,t)$ for the advection equation data X with $N = 100$. Right: the new representation of X by using the first 15 EOFs.

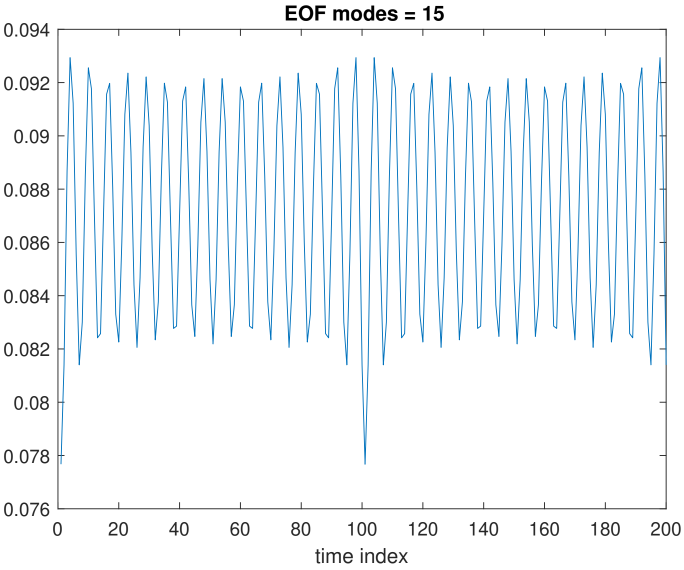


Figure 3.5: Temporal error growth for the approximation obtained from EOFs for the advection equation data X.

Our aim is to compare the dimensionality reduction methods. For comparison, we will take 15 modes for all methods. we also choose 15 EOF modes for the KdV equation data and they are implemented in Figure 3.6 with the minimized mean squared error illustrated in 3.7.

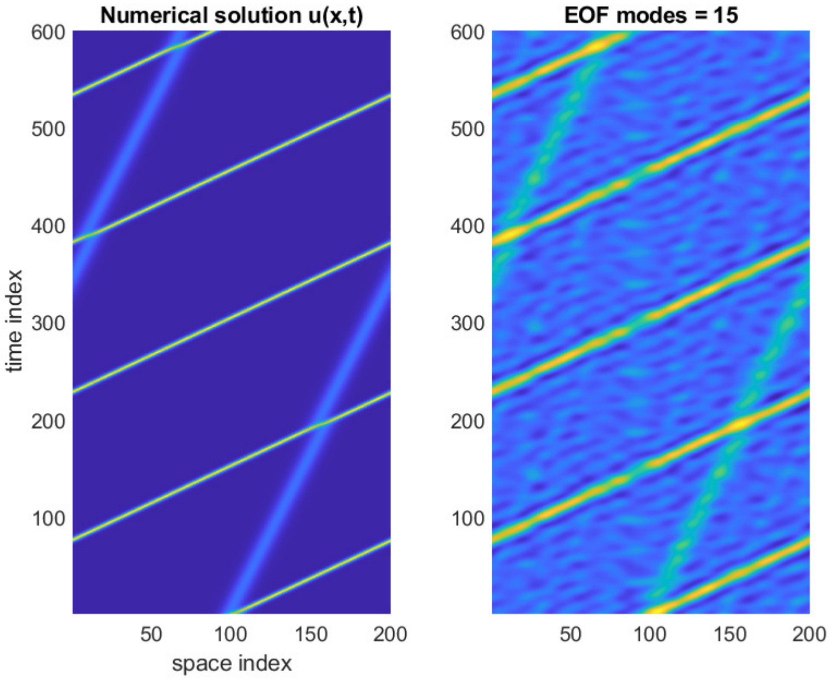


Figure 3.6: Left: the original data set X, taken from the KdV equation data, is shown with 200 eigenvectors. Right: the approximation of X by using the first 15 EOF modes.

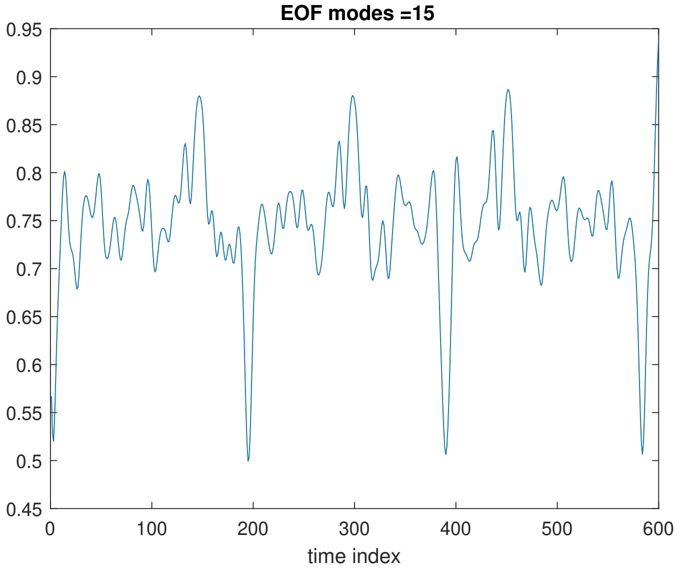


Figure 3.7: This plot shows the temporal error growth achieved by using 15 EOF modes for the approximation of X from the KdV equation data.

3.3 Numerical results with Diffusion maps (DM)

The construction of the diffusion map (DM) process is now implemented for the advection equation data and KdV equation data respectively. Thus, the original set $X \in \mathbb{R}^{N\times M}$ of data points is shown in Figure 3.8 on the left. In this plot, three lines show that the wave travels the first half, then covers the full path, and again travels half respectively. Firstly, we compute kernel matrix K for $\epsilon = 0.7$ for the advection data. We know that kernel measures the distance of each vector $X_i,\ i = 1,\dots,M$ from all M vectors. Each row of the kernel matrix contains the distance of one vector from all other vectors. Thus $K \in \mathbb{R}^{M\times M}$. Next, we find Markov transition matrix $P = \text{diag}(d)^{-1}K$, where d is the column vector containing the sum of each row of kernel matrix K. Then we find eigenvalues and eigenvectors of P and sort the eigenvalues in descending order. Lastly, we compute two components of the diffusion map for $t = 1$. Diffusion map embeds data points from $\mathbb{R}^N$ into 2-dimensional reduced space by taking two eigenvectors ψ_1 and ψ_2 of P. Here $\Psi_1 = \lambda_1\psi_1$ and $\Psi_2 = \lambda_2\psi_2$ by using the definition of diffusion map in (2.23). As is already known that the wave solution is just a vector of length N at each time step and the solution is translated at each time step and returns to its initial state due to periodic boundary conditions. Therefore, wave solution at different time steps moves around a closed curve, then the data points are mapped into the circle in the right plot of Figure 3.8 and diffusion map preserves the underlying topology, for instance, periodicity, of the data points.

Next, we compute kernel matrix K for $\epsilon = 2$ and $N+1$ eigenvalues of P. Then we find their corresponding eigenvectors used in (2.23) to compute the diffusion map (DM) modes $(\Psi_1, \Psi_2, \dots, \Psi_N)$ with each Ψ_i of length N. For DM modes, we also use the similar matrix form representation, presented in (3.3) and (3.4), for X and $\tilde{X}$ respectively. In this case, Figure 3.9 depicts around 10 percent projection error for almost 60 modes. And Figure 3.10 provides the comparison between the data X and its projection onto 60 DM modes.

For comparison between the DR methods, we choose the same $k = 15$ to get the new representation of X in the case of a diffusion map. In Figure 3.11, the original data set $X \in \mathbb{R}^{N\times M}$ is with eigenvectors $(\Psi_1,\dots,\Psi_N)$ and its reconstruction $\tilde{X} \in \mathbb{R}^{N\times M}$ is projected onto scaled eigenvectors $(\Psi_1,\dots,\Psi_k)$ of diffusion map. Also, the quality of the reconstructed solution is measured by computing and plotting the mean squared error, which is the function of time, in Figure 3.12.

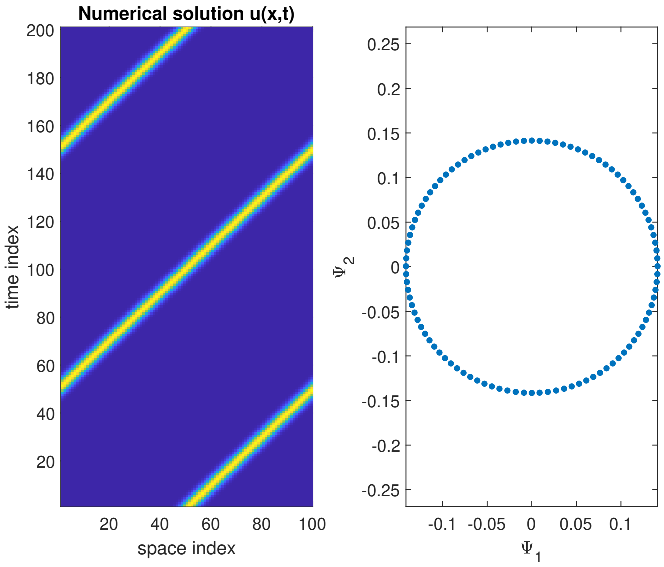


Figure 3.8: Space-time representation of data points taken from the advection equation data are shown in the left plot, and the right plot shows that data points are mapped into a perfect circle in diffusion coordinates.

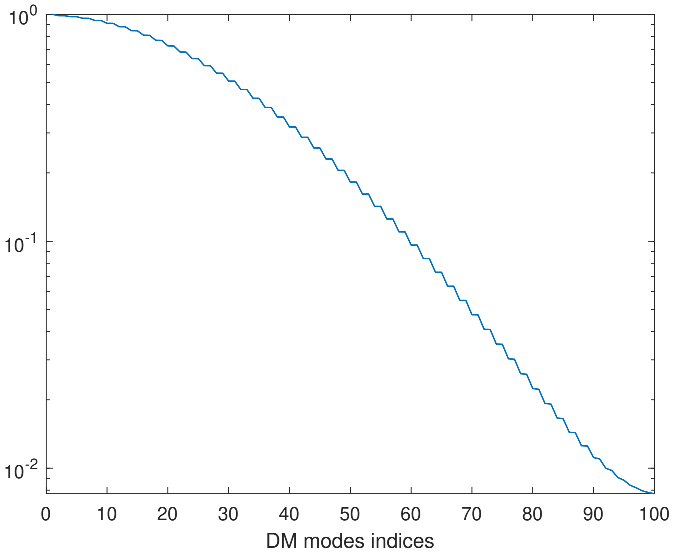


Figure 3.9

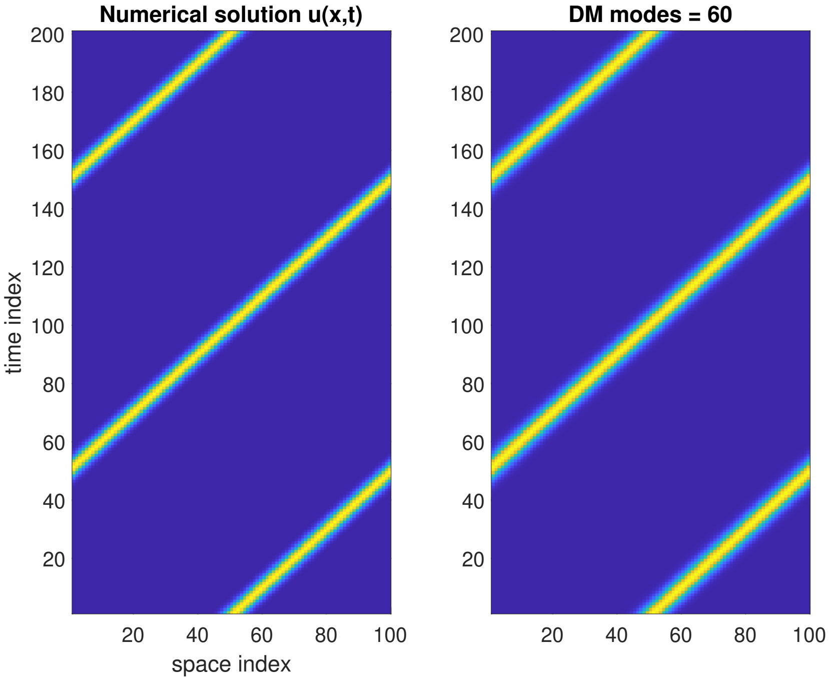


Figure 3.10: Comparison of the advection equation data X with 100 eigenvectors (left) with the approximation $\tilde{X}$ using first 60 scaled DM eigenvectors (right) in the extended space.

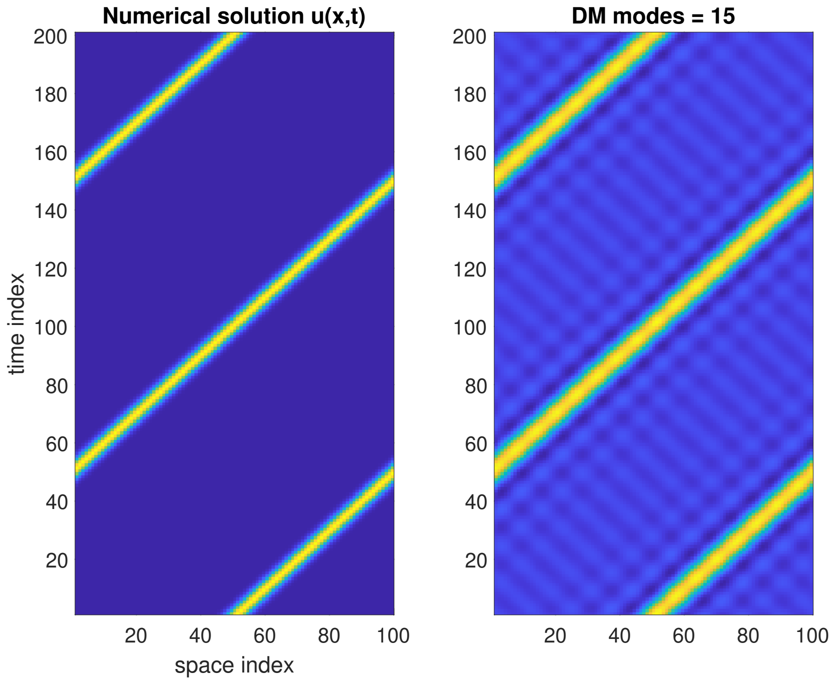


Figure 3.11: Comparison of the advection equation data X with 100 eigenvectors (left) with the low dimensional approximation $\tilde{X}$ with first 15 scaled DM eigenvectors (right) in the extended space.

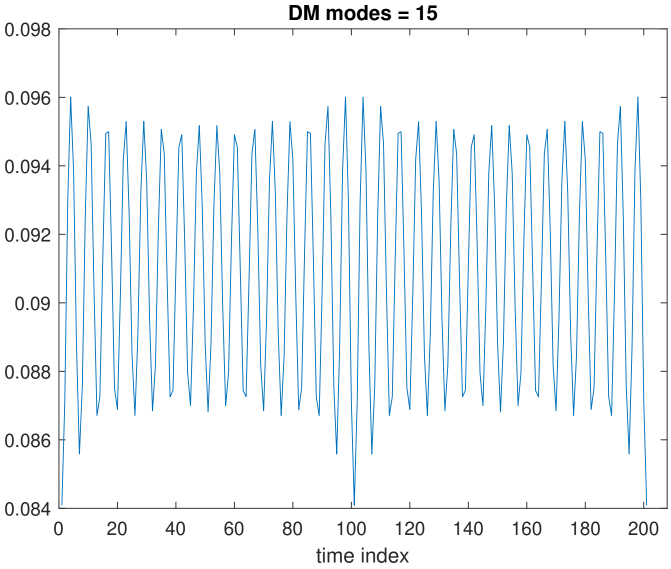


Figure 3.12: Temporal error between the original data set and the reconstructed solution for the advection equation data.

Next, the diffusion map process will be used for KdV equation data $X \in \mathbb{R}^{N \times M}$. In a similar way, the distance matrix $\mathbf{K}$ is computed by choosing $\epsilon = 165$. Figure (3.13) provides the around 50 percent projection error for almost 60 DM modes and Figure (3.14) provides the comparison between X and its approximation using 60 DM modes. The comparison of original data X and its reconstruction $\tilde{X}$ with $k = 15$ is illustrated in Figure 3.16 with the minimum error shown in 3.17.

Notably, Figure 3.15 indicates that the numerical rank of $\mathbf{P}^t$ decays by increasing the values of t. As explained in [5], the rows of the Markov transition matrix $\mathbf{P}^t$ provides the probability of transition in t steps which are based on the connectivities on the data set X. This means that eigenvectors associated with the largest eigenvalues are smooth and have the slowest decay and they provide the long-term behavior of the solution.

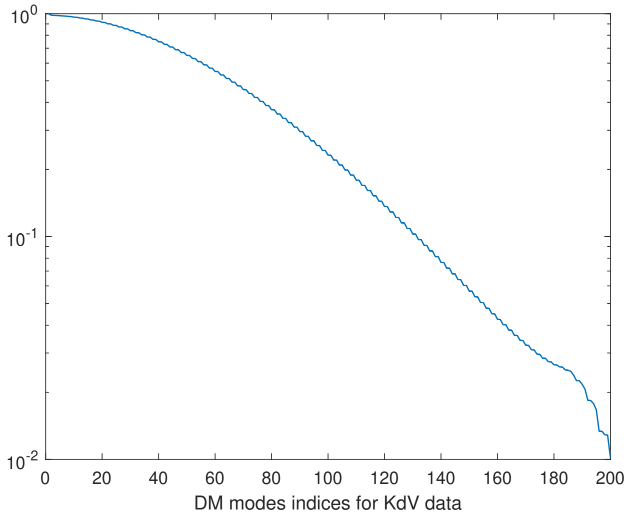


Figure 3.13

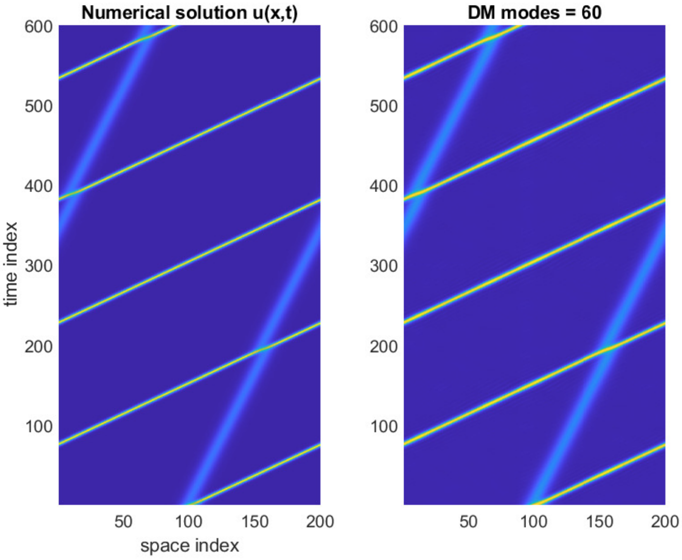


Figure 3.14: Comparison of KdV data X and its projection using 60 DM modes.

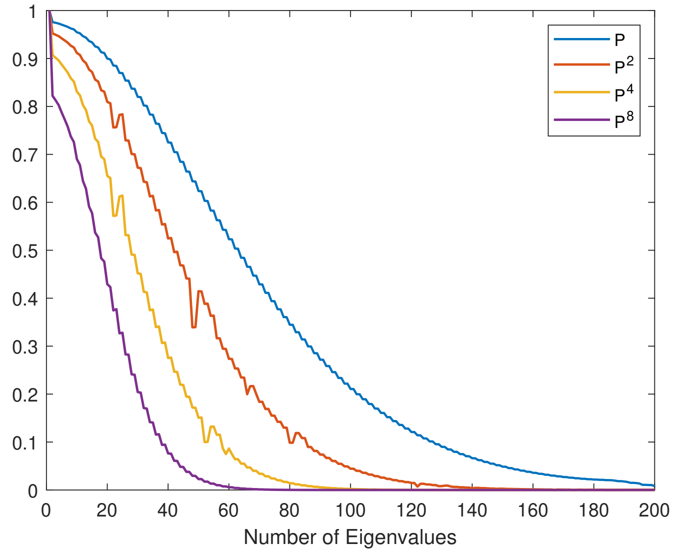


Figure 3.15: The decay of the eigenvalues for higher powers of transition matrix **P**.

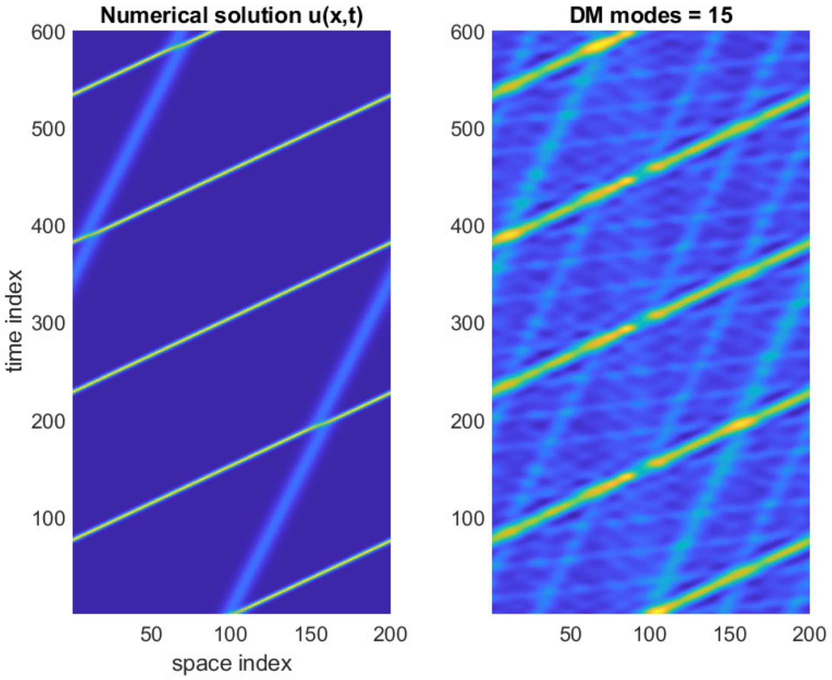


Figure 3.16: The KdV advection data (original data set) X is plotted by using 200 eigenvectors (left). While its approximation $\tilde{X}$ is projected onto the first 15 scaled diffusion map (DM) eigenvectors (right).

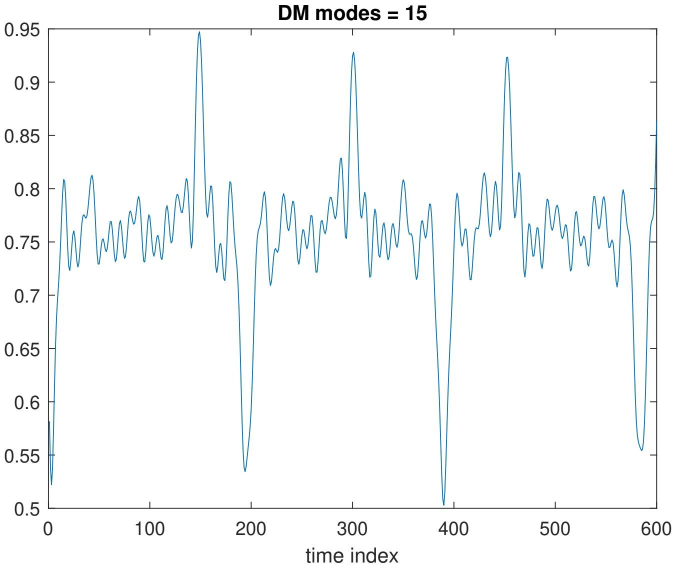


Figure 3.17: Above plot indicates the bounded temporal error between X and $\tilde{X}$.

3.4 Numerical results with Koopman basis

First, we will use the EDMD algorithm for the advection equation data and then for the KdV equation data. We know that the Extended Dynamic Mode Decomposition (EDMD) algorithm approximates Koopman eigenvalues and Koopman eigenfunctions. For EDMD, we form a pair of data sets from the original data set $X \in \mathbb{R}^{N\times M}$, such that $X = [X_1, X_2, \ldots, X_M]$ and $Y = [Y_1, Y_2, \ldots, Y_{M-1}]$, with $Y_i = X_{i+1}$. The matrix $\mathbf{K}$ is the finite-dimensional approximation of infinite-dimensional Koopman operator $\mathcal{K}$ and given as:

$$\mathbf{K} = G^{+}A,$$

where + denotes the pseudoinverse,

$$G = \frac{1}{M}\sum_{i=1}^{M} X_i X_i^T,$$

and

$$A = \frac{1}{M}\sum_{i=1}^{M-1} X_i Y_i^T$$

with $K, G, A \in \mathbb{C}^{N\times N}$. Here, we use the right eigenvectors of the matrix $\mathbf{K}$. The results are illustrated in Figures 3.18, 3.20, and 3.21 for the new representation $\tilde{X}$. For the Koopman operator, the eigenvalues are in ascending order. This case is quite interesting as compared to EOF and DM because Figure 3.19 illustrates that projection error does not change much after more or less 30 Koopman eigenfunctions.

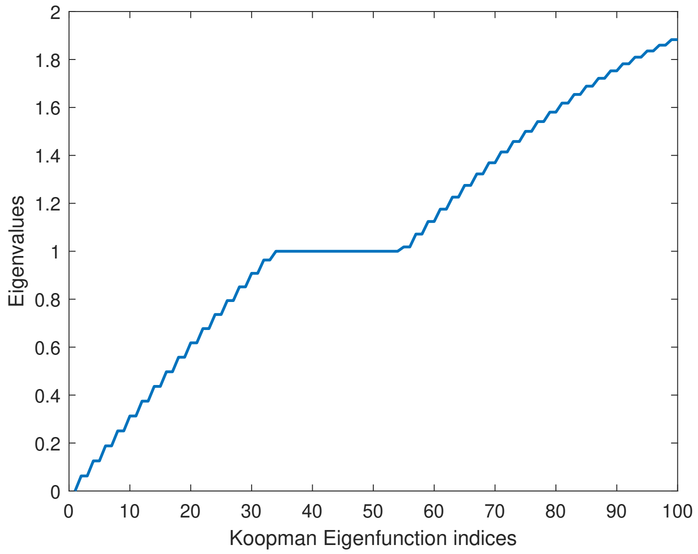


Figure 3.18

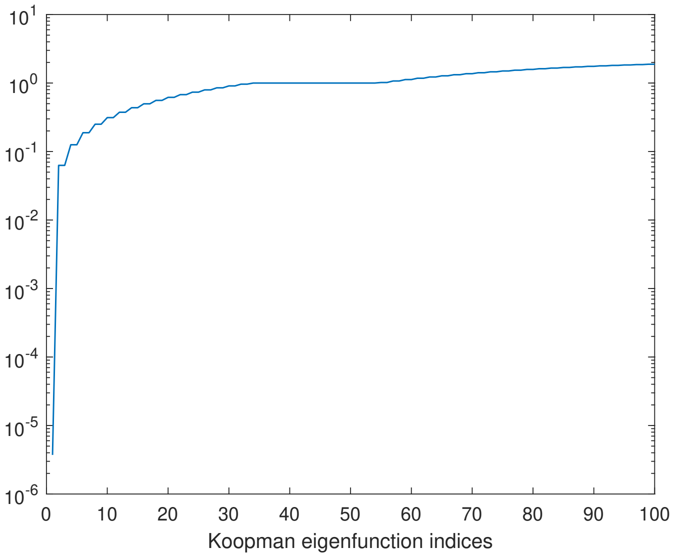


Figure 3.19: In contrast to EOF and DM methods, the projection error remains almost stable and does not increase after around 30 Koopman eigenfunctions for advection data.

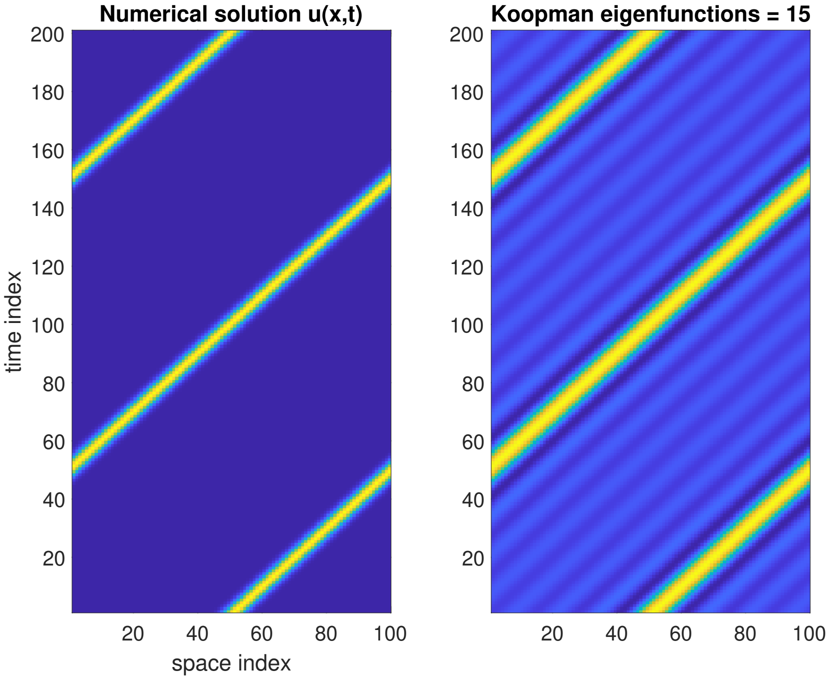


Figure 3.20: Left: the advection equation data X for $N = 100$. Right: the approximation of X by using the first 15 Koopman eigenfunctions.

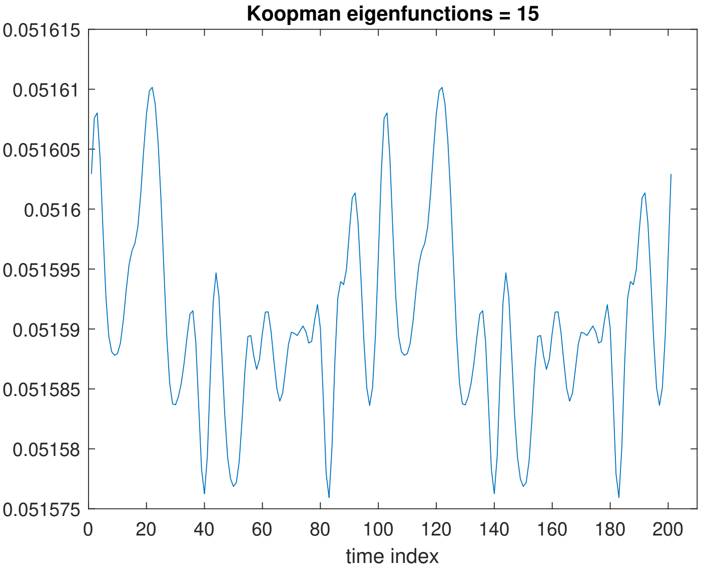


Figure 3.21: Error for the approximation obtained from Koopman basis for the advection equation data.

Now we will use the EDMD algorithm for the KdV equation data in a similar way as we mentioned for the advection data previously. However, in this case, we use left eigenvectors, known as Koopman modes, of the approximating matrix $\mathbf{K}$ because they give much better results than the right eigenvectors of **K**. We choose $k = 15$ for the new representation to comapare the results. Also, the results are illustrated in Figures 3.22, 3.23, and 3.24.

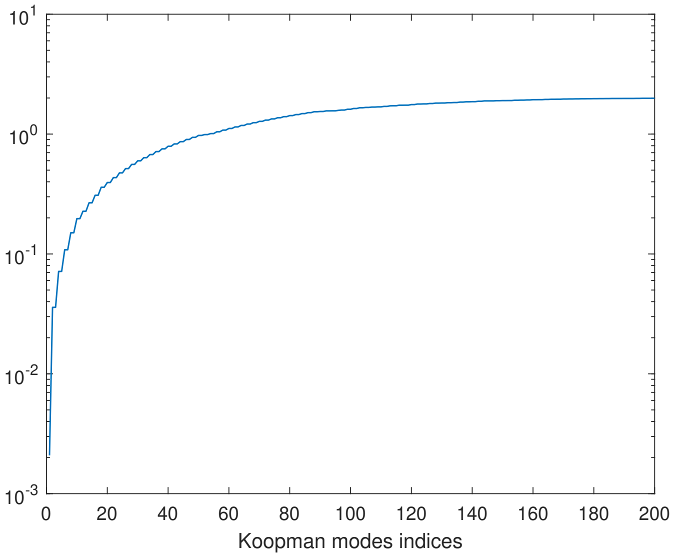


Figure 3.22: The projection error becomes almost constant after around 70 Koopman modes for KdV data, compared to EOD and DM modes

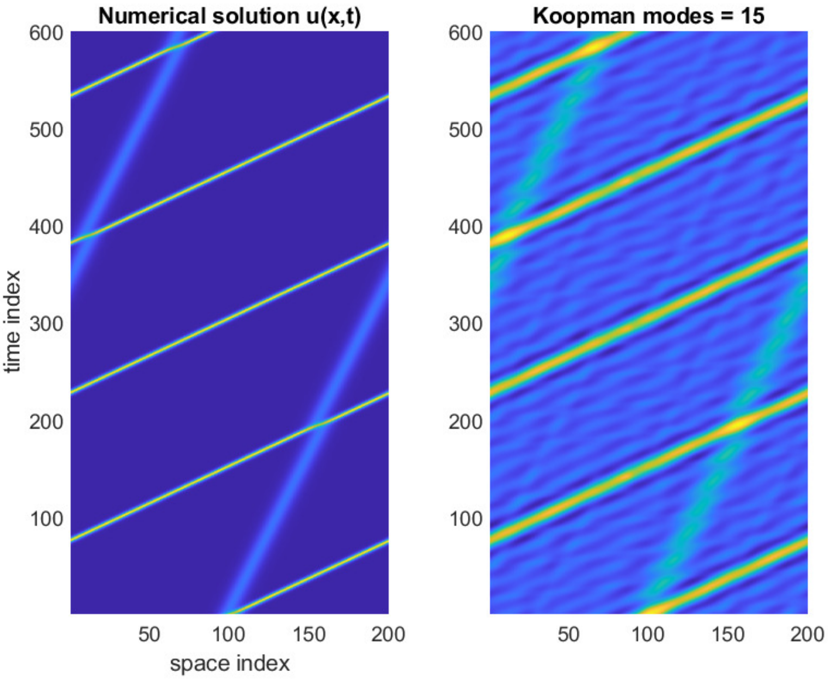


Figure 3.23: Left: the KdV equation data X for $N = 200$. Right: the new representation of X by using the first 15 Koopman modes.

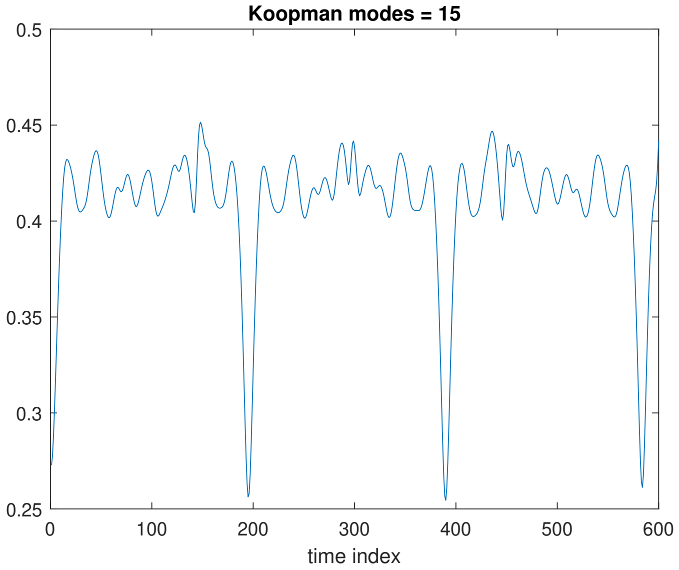


Figure 3.24: Time error for the approximation obtained from Koopman modes for the KdV data.

3.5 Numerical results with ALP

Now we will apply ALP algorithm [10] on the advection equation, described in (3.1). It is an integrable equation because the advection equation is exactly solvable. Also an exact Lax pair is defined by the Schrödinger operator $\mathcal{L}_\chi$, $\chi > 0$, given in (2.38) and the operator $\mathcal{M} = -cu_x$ defined in (2.32). Thus from [11], $\mathcal{L}_\chi$ and $\mathcal{M}$ satisfy the Lax equation (2.35) and advection equation is the compatibility condition for $\mathcal{L}_\chi$ and $\mathcal{M}$. Here we choose $\chi = 150$, N=100, Number of time steps (M) = 200 and $T_{max} = 1$, and we have $\Delta t = \frac{T_{max}}{M}$.

We take initial condition $u_0(x) = e^{-250(x-0.25)^2}$, which is a periodic function on the domain $[0, 1]$, as the data and find the eigenfunctions of Schrödinger operator $\mathcal{L}_\chi(u_0)\Phi = -\Delta\Phi - \chi u_0\Phi$. These eigenfunctions create the initial condition for the time-dependent basis called ALP modes. We project u on the orthonormal eigenfunction ϕ_i of Schrödinger operator $\mathcal{L}$ as an expansion of the form $u \approx \sum_{i=1}^{N} \beta_i(t)\phi_i(x, t)$. Then we get:

$$\sum_{j=1}^{N} \dot{\beta}_j\phi_j + \beta_j\partial_t\phi_j + c\sum_{j=1}^{N} \beta_j\partial_x\phi_j = 0$$

$\Rightarrow$

$$\dot{\beta}_i + \sum_{j=1}^{N} M_{ij}\beta_j + c\sum_{j=1}^{N} D_{ij}\beta_j = 0,$$

where $M_{ij} = \langle\partial_t\phi_j, \phi_i\rangle$ and $D_{ij} = \langle\partial_x\phi_j, \phi_i\rangle$. Thus, the ALP reduced-order method gives the following systems of ODEs:

$$\begin{aligned}
&\dot{\beta}_i + \sum_{l=1}^{N} M_{il}\beta_l - \gamma_i = 0\\
&\dot{\lambda}_i + \chi\sum_{l=1}^{N} T_{iil}\gamma_l = 0\\
&\dot{T}_{ijk} = \sum_{l=1}^{N}(M_{li}T_{ljk} + MljT_{ilk} + M_{lk}T_{ijl})\\
&\dot{D} = \sum_{l=1}^{N}(-D_{il}M_{il} + M_{il}D_{il})
\end{aligned} \tag{3.5}$$

where,

$$\gamma_i = -c\sum_{l=1}^{N} D_{il}\beta_l \quad \text{and} \quad M_{ij} = \frac{\chi}{\lambda_j - \lambda_i}\sum_{l=1}^{N} T_{ijl}\gamma_l$$

for $i, j, k = 1, \ldots, N$. For advection velocity $c = 0.5$, the Matlab code is implemented for the above ALP algorithm. In Figure 3.25, we compare exact solution of the advection equation and its projection $\tilde{u}$ achieved by projecting the solution u onto 15 ALP modes at M time steps.

Next, we form the advection data $X \in \mathbb{R}^{N\times M}$ by computing the exact solution $u(x, t)$ with respect to $N = 100$ intervals along the spatial variable x at $M = 200$ time steps. In order to get a new representation $\tilde{X}$ of X, we use the projection $\tilde{u} = \sum_{i=1}^{N} \beta_i(t)\phi_i(x, t)$, where $\phi_i(x, t)$ is changing in time, of u onto the 15 ALP modes at M time steps. In Figure, 3.26, X and $\tilde{X}$ are compared and it is shown that wave travels with given 200 time steps. Their mean squared error is shown in Figure 3.27.

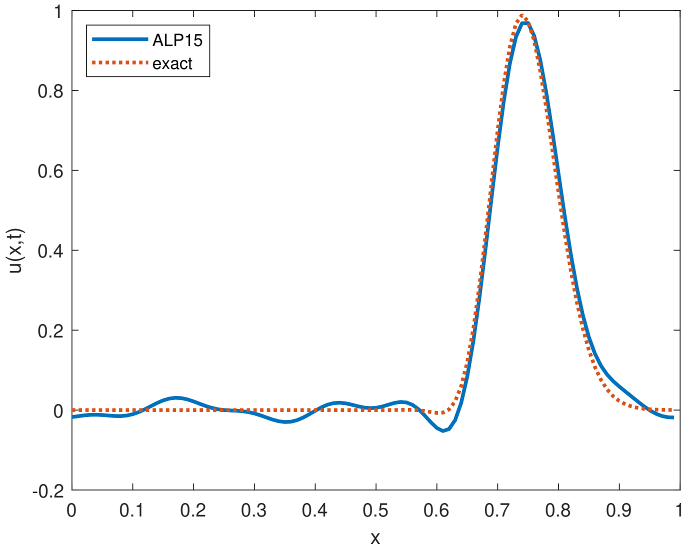


Figure 3.25: Above figure shows the exact solution $u(x, t)$ and its approximation with 15 ALP modes.

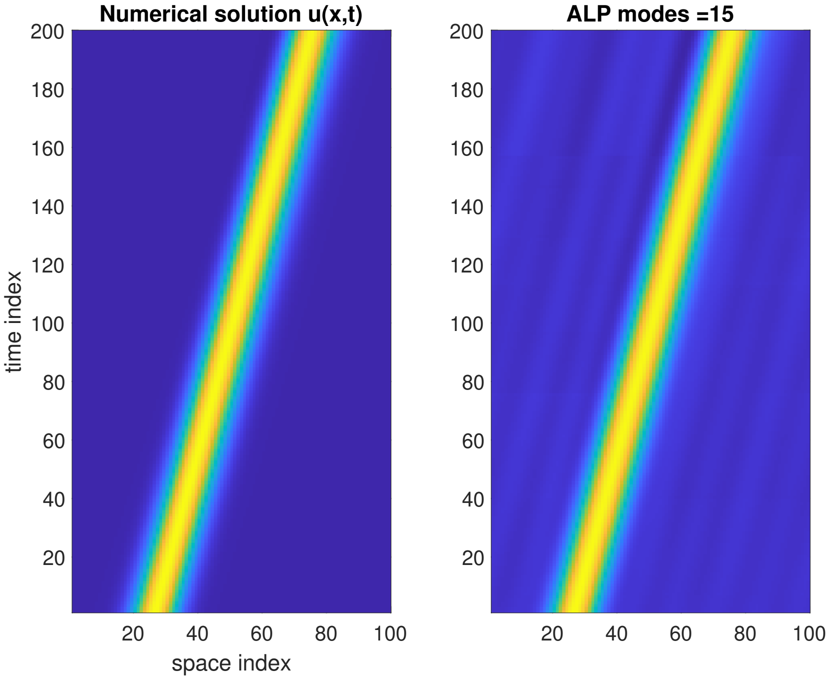


Figure 3.26: Comparison of the advection data X and its new representation $\tilde{X}$ are presented.

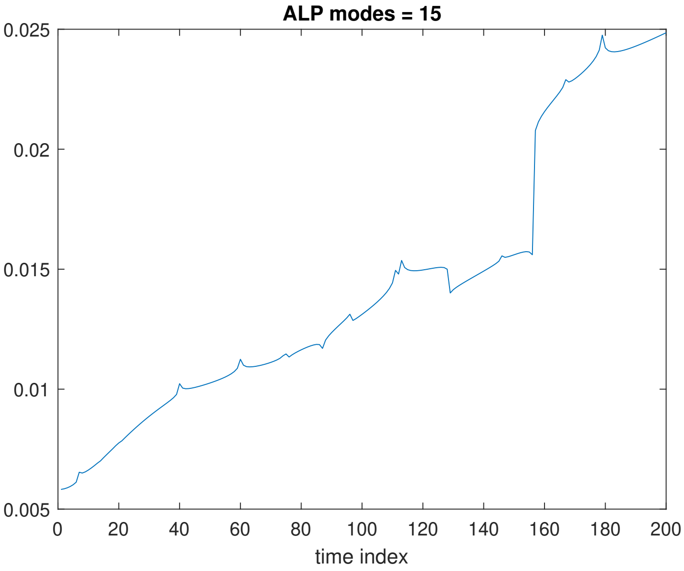


Figure 3.27: Time increasing error for the approximation obtained from ALP time-dependent basis for the advection equation data.

We will next apply ALP algorithm [9] on KdV equation data. KdV is also an integrable system. As is known that an exact Lax pair is defined by the Schrödinger operator $\mathcal{L}_\chi,\ \chi > 0$, given in (2.38) and the operator $\mathcal{M} = -6u\partial_x u - \partial_x^3 u$ defined in (2.32). Thus $\mathcal{L}_\chi$ and $\mathcal{M}$ satisfy the Lax equation (2.35), and KdV equation is compatibility condition [11] for $\mathcal{L}_\chi$ and $\mathcal{M}$. In this case, we choose $\chi = 1$, N=200, number of time steps (M) = 2000 and $T_{max} = 5$.
To form the data, we take initial condition $u_0(x) = 2\text{sech}^2(x - 15)$ on the domain $[0, 40]$ and find the eigenfunctions of Schrödinger operator $\mathcal{L}_\chi(u_0)\Phi = -\Delta\Phi - \chi u_0\Phi$. These eigenfunctions form the initial condition for the time-dependent basis. We compute the exact solution $u(x,t) = 2\text{sech}^2(x-4t)$ of the KdV equation and its projection $\tilde{u}$ achieved by projecting the solution u onto 15 ALP time-dependent modes at M time steps.
Next, we form original data set $X \in \mathbb{R}^{N\times M}$ by computing the exact solution $u(x,t)$ with respect to $N = 200$ intervals along the spatial variable x at $M = 600$ time steps. In order to get a new representation $\tilde{X}$ of X, we project the initial condition onto the 15 ALP dynamic modes at M time steps. In Figure 3.28, X and $\tilde{X}$ are compared and it is shown that wave travels with a given 600-time steps. Their mean squared error is shown in Figure 3.29. Figure 3.30 presents the better projection of X onto the 20 ALP modes and their lower error is shown in Figure 3.31.

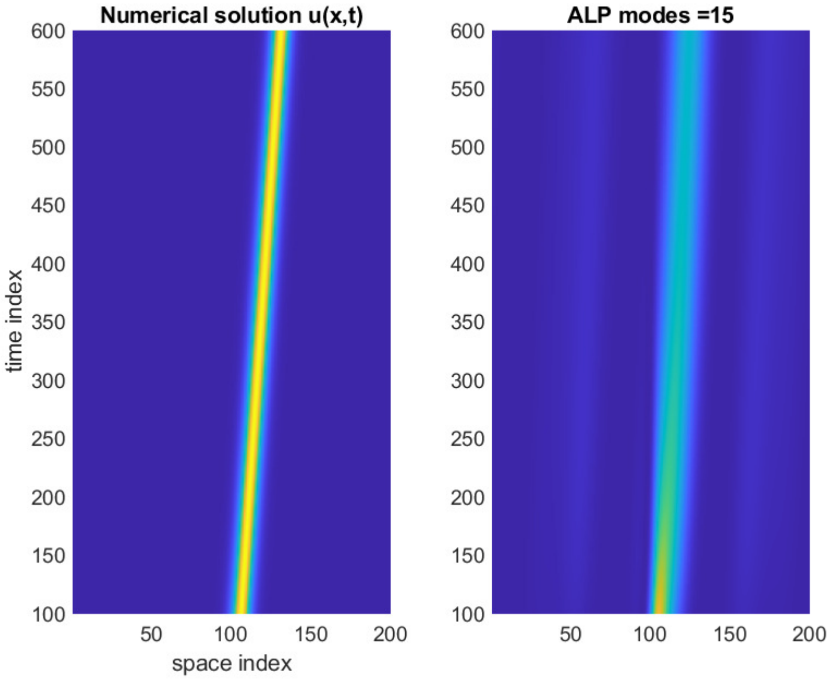


Figure 3.28: Comparison of KdV data X and its new representation $\tilde{X}$ are presented.

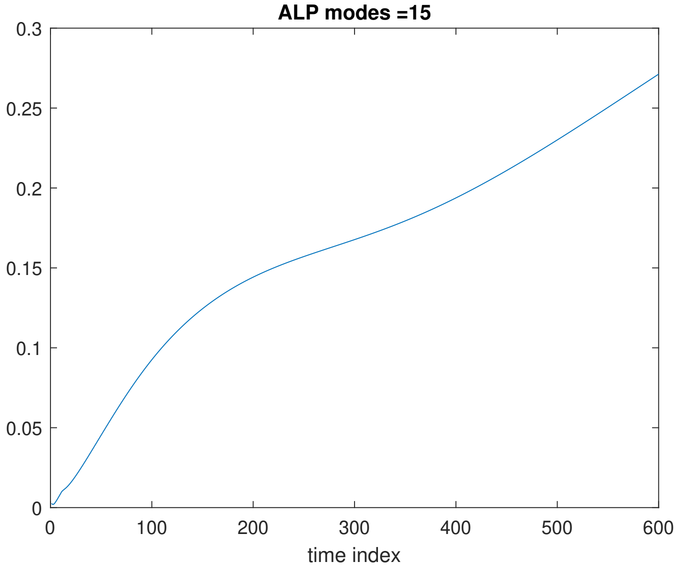


Figure 3.29: Time increasing smooth error for the approximation obtained from ALP time-dependent basis for the KdV equation data.

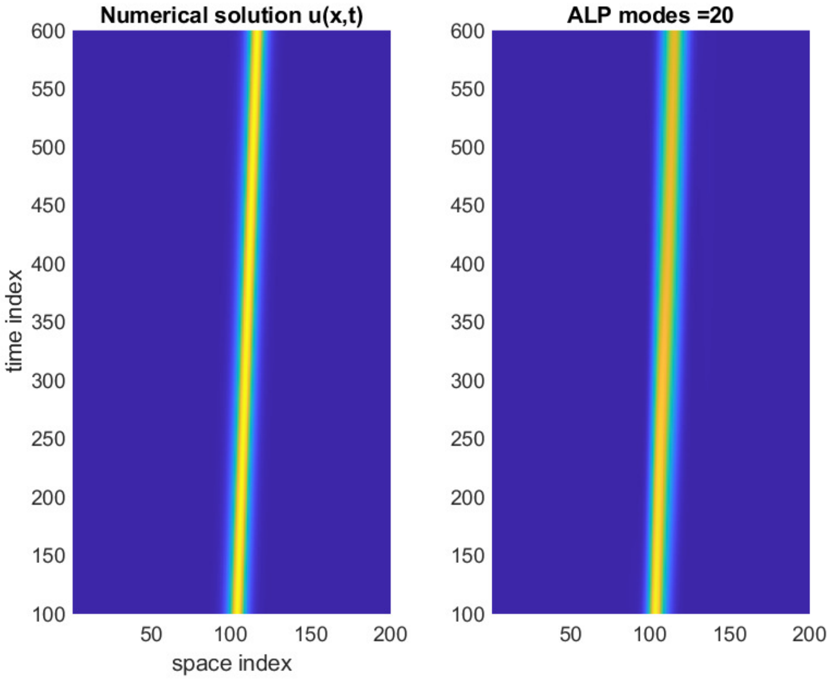


Figure 3.30: Comparison of KdV data X and its approximation $\tilde{X}$ are presented with 20 ALP modes.

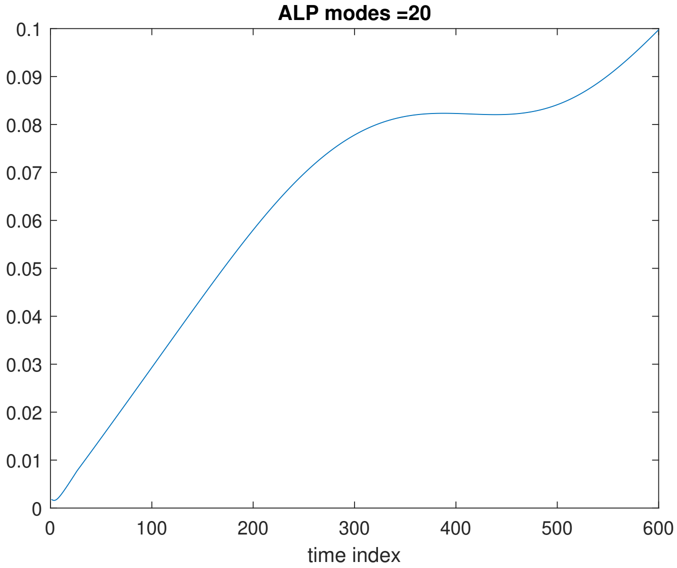
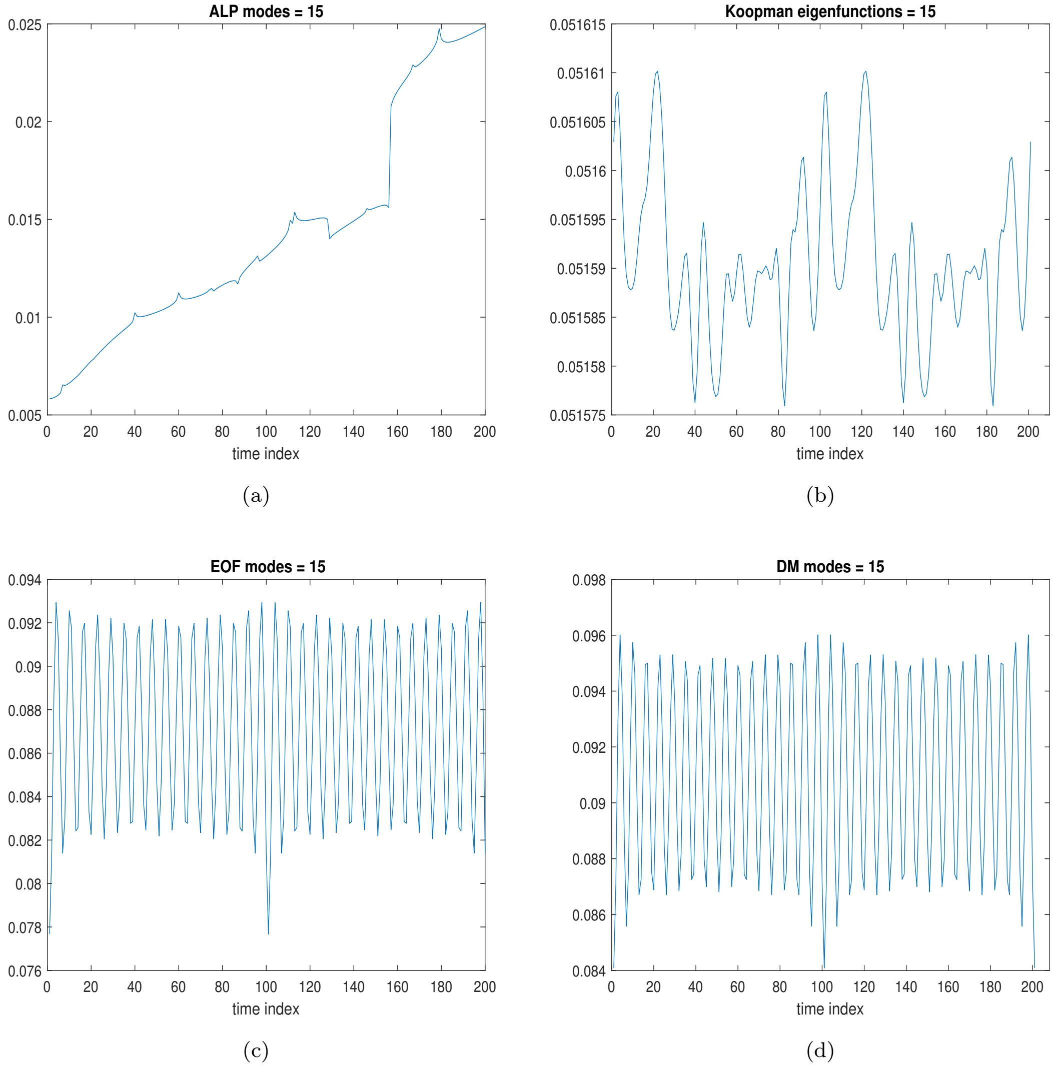


Figure 3.31: Time increasing error for the approximation obtained from ALP time-dependent basis for the KdV equation data.

3.6 Analysis for the numerical results

For advection equation data, in Figure 3.32, EDMD provides lower error than EOF and DM algorithms with fewer oscillations. Error due to ALP is the lowest, but this is not smoother and increases in time due to the time-dependent basis.

In Figure 3.33, in the case of KdV data, the error by the EOF method is more uniform and lower than the error due to DM modes. But the smoothness of error by DM modes might depend on the chosen value of $\epsilon = 165$. Also, the choice of ϵ is not trivial and is data-independent. This means that the smoothness of the error can be controlled by selecting different values of ϵ. EDMD error is surely better and more uniform than EOF. The ALP algorithm provides the least error, but ALP temporal error increases because of the dynamic nature of the approximation. Overall, the ALP algorithm works the best in both examples.



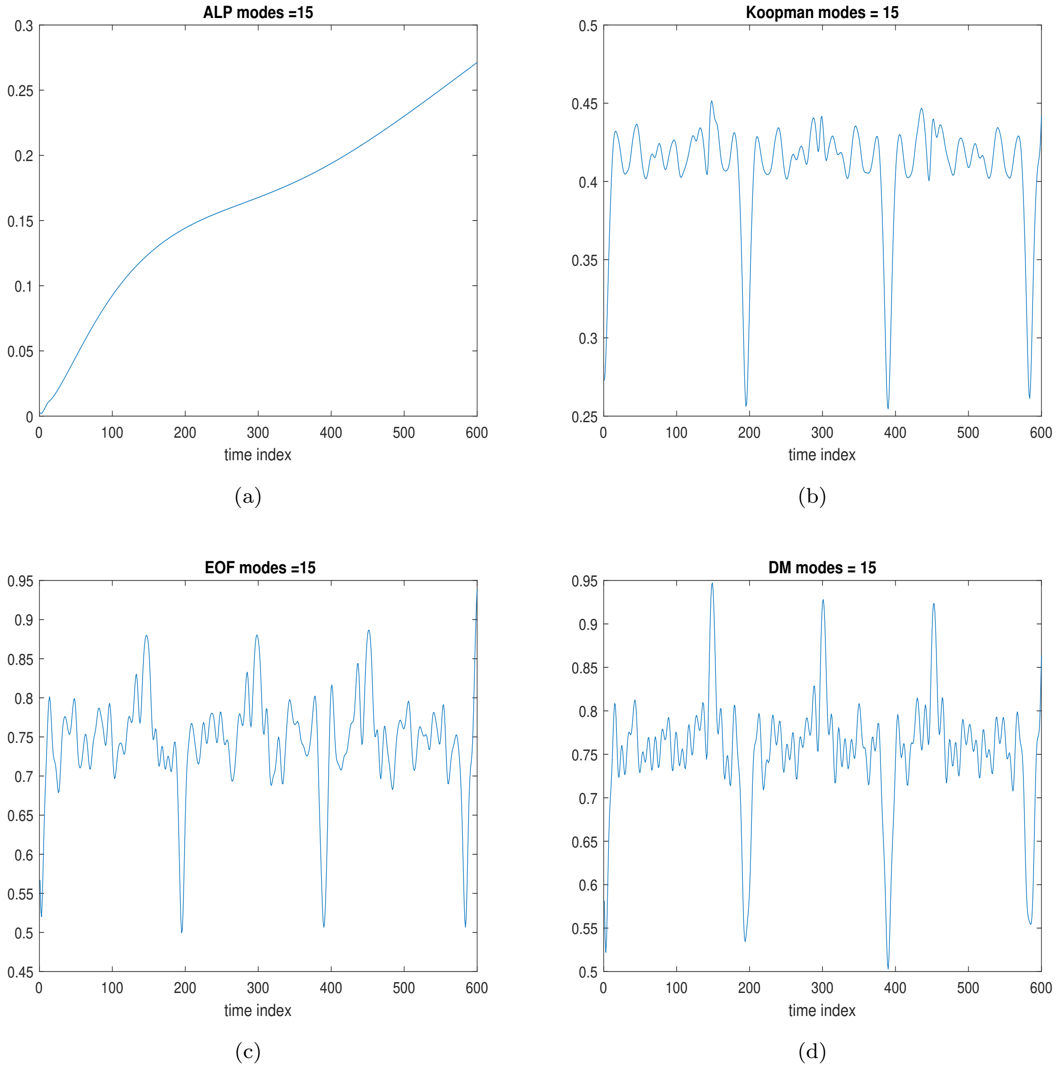
(a)

(b)

(c)

(d)

Figure 3.32: Comparison of errors achieved by the ALP, EDMD (or Koopman eigenfunctions), EOF, and DM modes for the advection equation data respectively. ALP shows the least error, but the error increases in time due to the dynamic nature of the approximation.



(a)

(b)

(c)

(d)

Figure 3.33: Comparison of errors achieved by the ALP, EDMD (or Koopman modes), EOF, and DM modes respectively by using the data from the KdV solution sequence.

Chapter 4

Conclusion

In order to summarize the key points, the main motivation of the thesis is to analyze and compare four dimension reduction (DR) methods, EOF, EDMD, DM, and ALP. For this, we have constructed data sets, taken from the sequence of the approximating solution set of two PDEs, namely the Advection equation and KdV equation, and used these two data sets for DR methods. Our results from DR methods indicate the importance of dimensionality reduction which plays a vital role in the analysis and understanding of the numerical data. In fact, dimensionality reduction helps to extract some useless information from data. As is already seen in the results, the error is not much affected by including more dimensions after the intrinsic dimensionality. Thus, we have the freedom to select a lower number of the basis for making the data size small and enhance the readability of data. Especially, there is no more change in the projection error by increasing the dimensions in the case of Koopman eigenfunctions. The projection error becomes almost constant after a particular number of Koopman modes.

It is noticeable in the results; four DR methods were applied for the advection equation data and the KdV equation data respectively. It is presented that the basis (modes) from EOF, EDMD, and DM are independent of time, whereas the basis set from the ALP algorithm evolves in time. And the basis from ALP changes as a function of time at each time step. Consequently, the ALP algorithm provides a better basis with more work, in contrast to other methods, but we cannot store ALP basis efficiently as it changes as a function of time. In the case of the advection data, the results indicate that the ALP algorithm works the best and gives the least mean squared error in contrast to EOF, EDMD, and DM methods. Although, the EOF shows better results than DM for both examples. For both data sets, the Koopman eigenfunctions and modes provide lower errors than EOF and DM errors. While an error due to DM algorithm depends on the positive constant. Concerning freedom in the choice of the parameter that does not depend on the data in the DM algorithm, DM can be more convincing than the EOF technique. Owing to the dynamic nature of the approximation, the ALP error increases over time. Overall the ALP algorithm gives the least error in both examples.

There has been a growing use of DR techniques to handle the high dimensionality of the data. As can be seen, several companies and organizations deal with high dimensional data sets which are deployed to solve complex and real-world problems, for instance, in analyzing MIR brain imaging scans and minimizing the cancer patient waiting time, etc. To remove poor-quality data or misleading columns or variables from the high dimensional data, DR techniques are very useful and beneficial. No doubt, DR techniques help improve the accuracy of the model of data sets due to less redundant data, speed up the model training time due to fewer dimensions and make the model simpler for data professionals and researchers. Additionally, Artificial intelligence (AI) developers work with massive data. To produce an AI model, it is required to take the three stages, such as understanding the complex data, cleaning the data of unnecessary information, and using the model. Thus, the DR techniques are very significant and used at the stage of cleaning in the process.

4.1 Acknowledgements

I would like to thank my supervisor Jason Frank for providing me with an interesting topic for my thesis work and for supporting me throughout. His knowledge and explanations also made me more motivated and enthusiastic about the topic. I also appreciate his kindness and patience during the discussions. Furthermore, I would like to thank Paul Zegeling for being the second reader of my thesis.

Finally, I must express my profound gratitude and love to both of my children for their patience and cooperation throughout my study years.

Bibliography

[1] Uri M Ascher and Robert I McLachlan. On symplectic and multisymplectic schemes for the kdv equation. *Journal of Scientific Computing*, 25(1):83–104, 2005. 21

[2] Bubacarr Bah. Diffusion maps: analysis and applications. 2008. 8, 10

[3] Nibodh Boddupalli. Extending dynamic mode decomposition to data from multiple outputs. *arXiv preprint arXiv:2108.01490*, 2021. 12

[4] Nibodh Boddupalli. An introduction to extended dynamic mode decomposition: Estimation of the koopman operator and outputs. *arXiv e-prints*, pages arXiv–2108, 2021. 12

[5] Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006. 8, 28

[6] John P Cunningham and Zoubin Ghahramani. Linear dimensionality reduction: Survey, insights, and generalizations. *The Journal of Machine Learning Research*, 16(1):2859–2900, 2015. 2

[7] J De la Porte, BM Herbst, W Hereman, and SJ Van Der Walt. An introduction to diffusion maps. In *Proceedings of the 19th symposium of the pattern recognition association of South Africa (PRASA 2008), Cape Town, South Africa*, pages 15–25, 2008. 8, 9

[8] Philip G Drazin and Robin Stanley Johnson. *Solitons: an introduction*, volume 2. Cambridge university press, 1989. 17

[9] Jean-Frédéric Gerbeau and Damiano Lombardi. Reduced-order modeling based on approximated lax pairs. *arXiv preprint arXiv:1211.4153*, 2012. 15, 17, 18, 19, 39

[10] Jean-Frédéric Gerbeau and Damiano Lombardi. Approximated lax pairs for the reduced order integration of nonlinear evolution equations. *Journal of Computational Physics*, 265:246–269, 2014. 15, 16, 36

[11] Graham W Griffiths. Lax pairs. 2012. 36, 39

[12] Abdel Hannachi. Regularised empirical orthogonal functions. *Tellus A: Dynamic Meteorology and Oceanography*, 68(1):31723, 2016. 3

[13] Abdel Hannachi, Ian T Jolliffe, and David B Stephenson. Empirical orthogonal functions and related techniques in atmospheric science: A review. *International Journal of Climatology: A Journal of the Royal Meteorological Society*, 27(9):1119–1152, 2007. 3

[14] Stéphane S Lafon. *Diffusion maps and geometric harmonics*. Yale University, 2004. 1, 9

[15] John A Lee and Michel Verleysen. *Nonlinear dimensionality reduction*, volume 1. Springer, 2007. 2

[16] Qianxiao Li, Felix Dietrich, Erik M Bollt, and Ioannis G Kevrekidis. Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the

koopman operator. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(10):103111, 2017. 12, 13, 14

[17] Ofir Lindenbaum, Arie Yeredor, Moshe Salhov, and Amir Averbuch. Multi-view diffusion maps. *Information Fusion*, 55:127–149, 2020. 8

[18] Douglas G Martinson. *Quantitative methods of data analysis for the physical sciences and engineering.* Cambridge University Press, 2018. 3

[19] Alexandre Mauroy, Yoshihiko Susuki, and Igor Mezić. Introduction to the koopman operator in dynamical systems and control theory. In *The Koopman Operator in Systems and Control*, pages 3–33. Springer, 2020. 12

[20] Adam H Monahan, John C Fyfe, Maarten HP Ambaum, David B Stephenson, and Gerald R North. Empirical orthogonal functions: The medium is the message. *Journal of Climate*, 22(24):6501–6514, 2009. 4, 6

[21] Boaz Nadler, Stéphane Lafon, Ronald R Coifman, and Ioannis G Kevrekidis. Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. *Applied and Computational Harmonic Analysis*, 21(1):113–127, 2006. 8

[22] Antonio Navarra and Valeria Simoncini. *A guide to empirical orthogonal functions for climate data analysis.* Springer Science & Business Media, 2010. 3

[23] Moshe Salhov, Amit Bermanis, Guy Wolf, and Amir Averbuch. Approximately-isometric diffusion maps. *Applied and Computational Harmonic Analysis*, 38(3):399–419, 2015. 9

[24] Claude Sammut and Geoffrey I Webb. *Encyclopedia of machine learning.* Springer Science & Business Media, 2011. 1, 2

[25] Laurens Van Der Maaten, Eric Postma, Jaap Van den Herik, et al. Dimensionality reduction: a comparative. *J Mach Learn Res*, 10(66-71):13, 2009. 1, 8

[26] Matthew O Williams, Ioannis G Kevrekidis, and Clarence W Rowley. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015. 12, 13

[27] Zhihua Zhang and John C Moore. *Mathematical and physical fundamentals of climate change.* Elsevier, 2014. 6