SPARSE DATA-DRIVEN MODELING FRAMEWORK FOR NON-LINEAR FLUID FLOWS

By

CHEN LU

Bachelor of Science in Mechanical Engineering
Oklahoma State University
Stillwater, Oklahoma
2015

Submitted to the Faculty of the
Graduate College of
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December, 2017

SPARSE DATA-DRIVEN MODELING FRAMEWORK FOR NON-LINEAR FLUID FLOWS

Thesis Approved:

Balaji Jayaraman

Thesis Advisor

Brian R. Elbing

Jamey D. Jacob

ACKNOWLEDGMENTS

To my advisor, Dr. Balaji Jayaraman: thank you for your excellent knowledge and unparalleled guidance devoted to this work.

To my committee members, Dr. Brian R. Elbing and Dr. Jamey D. Jacob: thank you for serving as my committee members and providing valuable suggestions to this work.

To Joshua Whitman and Dr. Girish Chowdhary: thank you for the collaboration and contribution for part of this work.

To the members of Computational Flow Data Science Group: thank you for the constant encouragements and useful recommendations.

To my aunt and uncle: thank you for always standing by me and supporting me in any possible way.

To my parents: thank you for your endless love and support throughout my life.

To many individuals who have contributed to this work: thank you!

Especially, I would like to thank Oklahoma State University and Department of Mechanical and Aerospace Engineering for the financial support throughout my graduate program. Thanks also go to the John Brammer Graduate Research Fellowship for the generous support provided during this work.

Acknowledgments reflect the views of the author and are not endorsed by committee members or Oklahoma State University.

Name: CHEN LU

Date of Degree: DECEMBER, 2017

Title of Study: SPARSE DATA-DRIVEN MODELING FRAMEWORK FOR NON-LINEAR FLUID FLOWS

Major Field: MECHANICAL AND AEROSPACE ENGINEERING

Abstract: Although fluid flows have been successfully modeled using the Navier-stokes equations, there exists only a limited class of flows for which exact analytical or numerical solutions are feasible at this time. For many practical flow situations, exact models are hard to find which makes the collection of data from experiments valuable. Consequently, there is a great demand for data-driven approaches that can predict the evolution and shed light on the flow physics when (a) the underlying knowledge of the dynamical system is non-existent and (b) the measurement data is sparse, but streams in over time. Such data-driven models are especially useful for control of dynamically evolving systems and online decision making for achieving improvement in performance and efficiency of many engineering problems such as drag reduction, mixing enhancement for combustion, and unmanned system navigation. The broad scheme of data-driven modeling comprises of three major components: development of reduced-order model from full data, reconstruction from sparse data to full data to analyze fine-scale flow features, and the ability to update the models online when the system is dynamically evolving in time. Although the above components have been studied in detail, integrating them into a practically usable tool has rarely been addressed. Hence, the overall goal of this research project is to develop an integrated dynamic data-driven modeling framework that leverages sparse streaming data with application to non-linear fluid flow systems. In this dissertation, the three major components of the data-driven modeling framework are developed, the various techniques explored and finally specific algorithmic advances for improved performance of prediction and reconstruction are reported as enumerated below. Firstly, a sparse generalized convolution framework is developed which is capable of predicting non-linear fluid flows such as flow past a cylinder and Boussinesq flow accurately and efficiently. Secondly, sparse reconstruction techniques based on L_2 and L_1 minimization, and statistical interpolation methods are investigated to recover incomplete data for flow past a cylinder and turbulent channel flow. The feasibility of incremental singular value decomposition is also examined so that the sparse convolution framework may be updated dynamically to improve non-equilibrium flow predictions.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

1.1 Overview and Motivation

Fluid flow problems are ubiquitous in many engineering applications such as transportation, energy systems, combustion processes, pipe and duct flow in HVAC systems, just to name a few examples. The common goal in such studies involves (a) identifying and understanding the fundamental physics of flow systems and (b) develop predictive models for (a) or (c) to design control strategies for improving efficiency. For instance, the ability to predict and control the wake behind a bluff body is beneficial for drag reduction, mixing and heat transfer enhancement (Cohen et al. 2003).

Many fluid flows are non-linear, multi-scale, and high-dimensional complex dynamical systems which are difficult to model directly. Even with the most accurate numerical methods, solving the governing partial differential equations are often too complex due to the computational effort required and lack of specification of realistic boundary and initial conditions. On the other hand, reduced-order modeling techniques have evolved rapidly in the past few decades with the development of efficient linear algebra libraries and provided reduced computational complexity. For example, Galerkin (Noack et al. 2011, Holmes 2012) projection of the governing equations onto a set of optimal orthogonal basis such as proper orthogonal decomposition(POD) (Lumley 2007) allows us to convert the set of PDEs to a set of ODEs which drastically reduces the computational cost of model prediction. Although it is widely applied (Kunisch & Volkwein 2002, Rapún & Vega 2010, Akhtar et al. 2012, San &

Borggaard 2015), projection-based methods require the knowledge of lower dimensional manifold (POD basis space) in addition to the governing equations, boundary and initial conditions. Note that precise boundary and initial conditions are not always straightforward for many complex flows, e.g.atmospheric flows (Blocken et al. 2007, Wu & Porté-Agel 2011, Dandou et al. 2017). The focus of this dissertation is to build a computational framework for modeling non-linear fluid flows that is (a) data-driven; (b) equation-free; (c) workable with sparse measurements and (d) capable of dynamic modeling as new data is available.

Dynamic mode decomposition(DMD) (Schmid 2010) is a popular choice within the fluid dynamics community as a purely data-driven method. In contrast with the POD-based methods, DMD provides information of the coherent structures as well as their associated temporal information, including the growth/decay rate and frequency. DMD has been extensively applied for extracting physical insights and underlying structures as shown in Schmid et al. (2011), Schmid (2011), Pan et al. (2011), Seena & Sung (2011), Muld et al. (2012), and Grosek & Kutz (2014). The DMD idea is used to build a dynamical model of a system and is based on learning a linear transition operator between snapshots of the dynamical system in time. However, instead of dealing with the higher dimensional state vector, this method essentially takes advantage of a similarity transformation of the transition operator using singular vectors of the data matrix. DMD has connections to linear system identification approaches that are prevalent in the controls community and well-known methods in machine learning such as kernel machines, Gaussian Processes(GP) regression amongst others that combine linear operator learning with kernel regression. A particular class of such methods is the Evolving Gaussian Processes(E-GP) framework that utilizes sparse GP kernels for modeling dynamical systems (Kingravi et al. 2015, 2016) in an evolving environment. The similarity of these methods such as DMD and E-GP is that they intrinsically involve learning a linear model in the trans-

formed space for modeling the dynamical system. For long, many such methods were considered unique approaches. However, recent interest (Mezić 2005, Rowley et al. 2009, Williams et al. 2015) in the pioneering work from Koopman (Koopman 1931) has allowed these methods to be cast in a different light. The work of Koopman essentially states that under an appropriate mapping from the state space to a feature (Hilbert) space, the non-linear dynamics of a system can be transformed into linear at the cost of dealing with an infinite dimensional system. This explains the success of the predominantly linear class of methods to work reasonably well for predicting select nonlinear dynamics. Motivated by this thinking, extensions to DMD have been proposed which are more promising for modeling non-linear systems (Williams et al. 2015, 2014). Known as extended DMD or EDMD (Williams et al. 2015), these methods include a non-linear mapping to a feature space for building the linear model. However, the computational cost of EDMD increases dramatically as the dimension of the system increases. This *curse of dimensionality* of EDMD can be overcome by using the kernel trick (Williams et al. 2014). The above class of pure data-driven methods achieves model reduction by mapping to a feature space that (a) reduces dimensionality and (b) allows the Koopman-based modeling of a non-linear system. This dissertation aims to place all these different established data-driven techqniques into an unified *Sparse Generalized Convolution Framework* where the ultimate goal is to realize a finit-dimensional approximation to the Koopman observable space such that the dynamics can be adequately represented using a linear operator. A major contribution from this thesis work is the exploration of different strategies to build nonlinear convolution maps to the feature space that is optimal for the physics of interest. It is anticipated that this approach might prove to be more robust as against building complex nonlinear regression models such using artificial neural netwroks.

A common issue with data-driven models is that the data is often sourced from sparse measurements. While one can build a dynamical model as above from sparse

data, the resulting predictions will still be in a sparse space and not sufficient for identifying the relevant flow structures of interest. Consequently, sparse reconstruction is a critical component in data-driven approaches for modeling and analyzing non-linear dynamical systems. The earliest known work on recovering the missing data was developed by Yates (1933) using least-squares estimates. In the field of geology, local Kriging has been an effective statistical technique to estimate spatial data using least-squares approach. Venturi & Karniadakis (2004), Gunes et al. (2006), and Gunes & Rist (2008) have adopted local kriging to interpolate fluid flow systems with success. On the other hand, reconstruction in a sparse POD basis space using least-squares method, also known as "gappy POD (GPOD)," has emerged in the image processing community to recover marred faces (Everson & Sirovich 1995). Later, Bui-Thanh et al. (2004) successfully applied GPOD for compressible external aerodynamic flow problems to reconstruct missing data and also for inverse airfoil design. In recent years, compressive sampling as a signal reconstruction techniques has also been explored in the fluid community. Brunton et al. (2013, 2014, 2016), Bai et al. (2017), and Kramer et al. (2017) have integrated compressive sampling strategies (Candès et al. 2006, Tropp & Gilbert 2007, Needell & Tropp 2009) with dynamical modeling techniques such as DMD to build a sparse system identification. In particular, Bright et al. (2013) utilized compressive sampling to characterize the flow around a cylinder with limited pressure measurements and Bai et al. (2014) reconstructed airfoil data using such strategies. Hence, sparse reconstruction is an integral component of data-driven modeling of practical non-linear fluid flows. Exploiting sparse reconstruction techniques can greatly reduce the cost of collecting and analyzing fluid flow data.

For real-time decision making and control, refining the data-driven model online is critical and often desired. This is needed to avoid rebuilding the entire model for systems that have high dimensions and will not be feasible within the time-constraints and on a low-memory onboard computer. Thus, the real-time model generation is

imperative for dynamical systems that are evolving and/or operating in environments that are rapidly changing with new information streaming in. In particular, the online update strategies relevant to this thesis are those that allow to incrementally build reduced-order models by incorporating new information as in Peherstorfer & Willcox (2015) and Peherstorfer & Willcox (2016). These techniques can also incorporate sparse or partial data to evolve the model and potentially extendable to building models from large amounts of data that cannot be handled in bulk. Matsumoto & Indinger (2017) have incorporated incremental singular value decomposition method in DMD to reduce memory consumption while extracting the dominant structures from an unsteady flow. Furthermore, Oxberry et al. (2017) have integrated adaptive method for basis selection which circumvents the offline training of reduced order models. Thus, online updates methods also contribute towards developing effective and efficient dynamical models from large amounts of data.

In summary, data-driven modeling comprises of three main components:

- non-linear reduced-order models from sparse/fine data;
- sparse reconstruction from incomplete data;
- online updates for dynamically evolving systems.

This thesis presents the exploration into various strategies and the generated advancements to accomplish the above three aims. The uniqueness of this work is that the above goals were explored together with the ultimate aim of integrating these different components. Such integrated efforts are rarely pursued due to the associated complexity. Notably, the research group at MIT (Peherstorfer & Willcox 2015, 2016) are the only other research group that has explored along similar ideas. However, their research interests focus predominantly on inverse problem design for structural system while this study investigates highly non-linear fluid flow systems. To this

extent, the ultimate goal of this research is to fill the gaps in knowledge by developing dynamic data-driven models from sparse measurement data with applications to turbulent flow systems where the exact governing equations, boundary and initial conditions may not be available.

1.2 Contribution

In chapter 2, the main contribution is the generalization of the data-driven techniques mentioned previously including DMD (Schmid 2010), E-GP (Kingravi et al. 2015), and EDMD (Williams et al. 2015). The underlying idea of these methods is to identify a mapping to feature space which is strongly related to the Koopman operator theory. This work presents a Koopman-based, *generalized sparse convolution framework* that allows for physics-specific mapping into a feature space using convolution operator. Thus, the emphasis here is that the key to modeling non-linear fluid flow systems from data depends on the choice of convolution and not on nature of the transition operator. Potential choices for convolution operator that are considered in this work include POD modes and GP kernels so that they can be related to DMD and EGP class of methods. Furthermore, the proposed framework is also extended to multi-layer convolution such as layering GP with POD, POD with transfer functions (Rowley & Dawson 2017), and POD with kernel functions (Williams et al. 2014).

In chapter 3, the focus is on the evaluation of the various sparse reconstruction techniques such as L_2 (Everson & Sirovich 1995, Bui-Thanh et al. 2004) and L_1-based minimization (Brunton et al. 2014) for fluid flow systems. The major contribution here is the development of a unified sparse reconstruction framework that combines ideas from compressive sensing, Kriging and Gappy reconstruction procedures that are popular in the signal processing, geosciences, and image processing communities respectively. This framework is developed with potential application to nonlinear fluid flow systems. In particular, turbulent channel flows investigated in this work, while

not the ideal sparse dynamical system, is used a case study for such class of methods so that the precise limitations can be identified and addressed in future. Similar recommendations of applying reconstruction techniques to wall-bounded flows are also suggested by Gunes et al. (2006). Therefore, the performance of these methods are examined, and potential issues for high-dimensional systems are identified. The other major contribution of this work is to extend this unified sparse reconstruction framework for situations when the exact low-dimensional basis space is not known. In this context, this work integrates the ideas of progressive method (Venturi & Karniadakis 2004) and kriging (Gunes et al. 2006) into L_2 and L_1-based iterative methods, and the impact of these improvements is explored.

In chapter 4, online low-rank singular value decomposition updates are investigated, and the major contribution is to generalize the rank-1 updates algorithms (Peherstorfer & Willcox 2015, Oxberry et al. 2017, Matsumoto & Indinger 2017) to variable rank, i.e., rank-K updates. This implementation is validated by comparing the reconstructed error, updated singular vectors and singular value. More importantly, the developed incremental SVD updates can be integrated into the generalized convolution framework presented in chapter 2 for modeling systems that evolves in time. In contrast with rank-1, rank-K updates are more efficient when the model is allowed to update with every K snapshots of new data.

The overall contribution of this dissertation is to advance the development of dynamic data-driven models that can handle sparse data using a three-pronged focus as discussed above. Such capability is potentially useful for many practical flow science applications where only sparse data information is available.

1.3 Scope of Study

The primary scope of this thesis is to explore and advance each of the components mentioned above so that they can be subsequently integrated to build a dynamic

data-driven modeling platform capable of handling realistic non-linear fluid flows. Consequently, the structure of this thesis is aligned as follows.

In chapter 2, a sparse convolution framework is developed as a generalization for modeling the dynamics of non-linear fluid flows. This framework relies on the ideas of the convolution and sparse representation build accurate and efficient dynamical models. Next, sparse reconstruction techniques are investigated to recover full field data from incomplete data with random sensor placement in chapter 3. This capability is useful for physics-based investigation of the predictions from the dynamical models developed in chapter 2 to analyze sparse data. The strategies for online update of the data-driven convolution filter, i.e., online singular value decomposition in a dynamical setting where recently collected data can be leveraged to improve the data-driven model are explored in chapter 4. In each chapter, we will motivate the efforts, present the objectives and delineate the contribution from this dissertation. Also, the underlying methodology and the key results will be discussed in necessary detail. As a wrap, the future research directions to evolve from this work will be presented in the concluding chapter.

1.4 Physics Case Studies

To demonstrate the viability, efficiency, and accuracy of the explored methods in each chapter, standardized test problems that include fluid flows across different flow regimes including bluff body, mixing and wall-bounded turbulent flows are adopted. The selected flow dynamical systems for the analysis exhibit varying levels of complexity, i.e., from periodic (limit-cycle) to transient (non-limit-cycle) system, from low-dimensional to very high-dimensional complex turbulent flow system. The developed data-driven modeling techniques are tested and validated using data from the numerical simulations for flow past a cylinder at different Reynolds numbers, buoyancy-driven mixing flow (Boussinesq), and turbulent channel flow. Our ultimate

goal is to develop data-driven techniques capable of handling three-dimensional turbulent channel flow with high spatiotemporal complexity and large-scale separation. In the following section, the data generation and collection process are detailed.

1.4.1 Flow Past a Cylinder

Flow past a cylinder has been well-studied using laboratory and numerical experiments as a generic substitute for bluff body wake flows(Roshko 1954, Williamson 1989, Noack et al. 2003, Rowley & Dawson 2017). This flow system is rich in its physics content and encompasses many of the non-linear phenomena such as wake instability and mixing that are prevalent in many bluff body wakes. For this study, we intend to explore data-driven models for two different flow regimes: the periodic phase with repeated vortex shedding and the transient phase that includes the onset of vortex shedding. The former exhibits classical limit cycle behavior, whereas the transient phase represents the evolution towards a limit cycle.

To generate two-dimensional cylinder flow data, the spectral method (Cantwell et al. 2015) are utilized to solve incompressible Naiver-Stokes equations, as shown in Equation 1.1, in our simulations.

$$\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = 0, \tag{1.1a}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{1}{\rho}\frac{\partial P}{\partial x} + \nu\nabla^2 u, \tag{1.1b}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{1}{\rho}\frac{\partial P}{\partial y} + \nu\nabla^2 v, \tag{1.1c}$$

where u and v are horizontal and vertical velocity components. P is the pressure field, and ν is the fluid viscosity. The rectangular domain used for this flow problem is $-25D < x < 45D$ and $-20D < y < 20D$, where D is the diameter of the cylinder. For the analysis, a reduced domain, i.e., $-2D < x < 10D$ and $-3D < y < 3D$, is investigated, and the corresponding mesh configuration for $Re = 100$ case is shown in Figure 1.1.

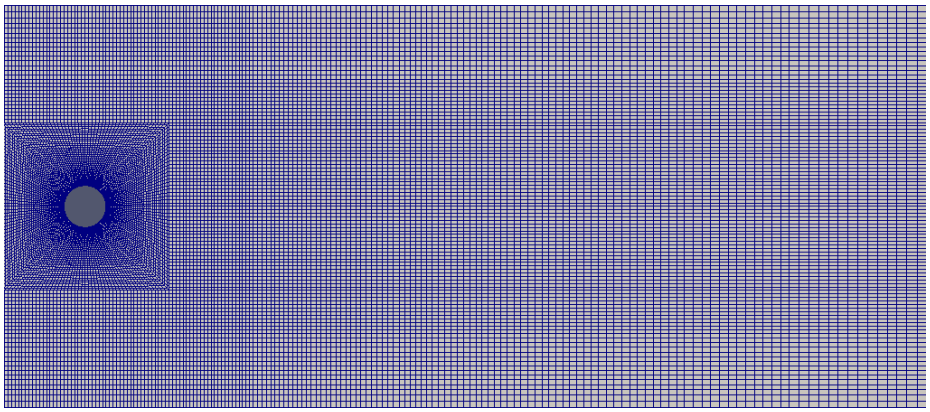


Figure 1.1: The reduced domain of mesh configuration used for flow past a cylinder at $Re = 100$.

For the case of $Re = 100$ the grid includes 24,000 points whereas for $Re = 1000$ the grid is refined to include 95,000 points for the sample flow region. Because $Re = 1000$ has a thinner shear layer compared to $Re = 100$, the near-wall resolution has been increased. The mesh is designed to have resolution concentrated in the region around the cylinder as well as the wake of the cylinder to accurately capture the relevant dynamics. The no-slip boundary condition is applied to the cylinder wall, and a homogeneous Neumann boundary condition is applied at the far-field. At the inlet boundary, the flow velocity vector is specified as a Dirichlet boundary condition. The sampling rate for output of snapshots of data is chosen as $\Delta t = 0.2$ second, and the time unit for each iterative computation is 0.0008 second to satisfy numerical stability criteria.

1.4.2 Unsteady Buoyancy-Driven Mixing (Boussinesq)

The second test flow considered is an unsteady Boussinesq mixing flow (Weinan & Shu 1998, Liu et al. 2003, San & Borggaard 2015), also known as lock-exchange problem (San & Borggaard 2015), which exhibits strong shear and Kelvin-Helmholtz instabilities driving the dynamics. This flow has strong similarities to atmospheric boundary layer flows where shear and buoyancy-driven turbulence interact to deter-

mine the dynamics. Compared to the cylinder flow which has limit-cycle behavior, the Boussinesq flow is highly convection driven and does not display a limit-cycle type dynamics. Such a flow will serve as a useful test for data-driven methods that rely on the limit amount of data in space and time.

To generate the data, the dimensionless form of incompressible Boussinesq equations (San & Borggaard 2015), as shown in Equation 1.2 is solved on a 2-D rectangular domain that is $0 < x < 8$ and $0 < y < 1$.

$$\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = 0, \tag{1.2a}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{\partial P}{\partial x} + \frac{1}{Re}\nabla^2 u, \tag{1.2b}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{\partial P}{\partial y} + \frac{1}{Re}\nabla^2 v + Ri\theta, \tag{1.2c}$$

$$\frac{\partial \theta}{\partial t} + u\frac{\partial \theta}{\partial x} + v\frac{\partial \theta}{\partial y} = \frac{1}{RePr}\nabla^2 \theta, \tag{1.2d}$$

where u, v, and θ are the horizontal, vertical velocity, and temperature components, respectively. The dimensionless parameters Re, Ri, and Pr are the Reynolds number, Richardson number, and Prandtl number, respectively. In this study, $Re = 1000$, $Ri = 4.0$, and $Pr = 1.0$ are used to generate the numerical data. The grid size is chosen to be 1024 by 128. Initially, fluids at two different temperatures are separated by a vertical wall at $x = 4$. The surrounding walls are adiabatic and have no-slip boundary condition. A fourth-order compact finite difference scheme is used to compute the derivatives in Equation 1.2 to obtain an accurate flow representation.

1.4.3 Turbulent Channel Flow

For the complex end of the flow system test bed, a turbulent channel flow at a moderate Reynolds number is chosen. Study of near-wall boundary layers provides statistical and structural characteristics (Kim et al. 1987, Moser et al. 1999) that are important for many of the engineering problems, including drag reduction (Du et al.

2002), oil and gas transportation, and heat convection, just to name a few. However, the physics of wall-bounded flows has not been completely understood because they exhibit a broad range of length scales which are associated with one another (Smits & Marusic 2013). In this thesis, such a fully-developed turbulent channel flow data from direct numerical simulations are incorporated into the test bed to assess the effectiveness of the sparse reconstruction models developed in chapter 3.

To generate high fidelity data, the skew symmetric form of the incompressible 3-D Navier-Stokes equations is solved on a rectangular box that is $0 < x < 12.6$, $0 < y < 1$, and $0 < z < 1$. For simplicity, the 2-D version of the equations is shown as follows.

$$\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = 0, \tag{1.3a}$$

$$\frac{\partial u}{\partial t} + \frac{1}{2}\Big[\frac{\partial (u^2)}{\partial x} + \frac{\partial (uv)}{\partial y} + u\frac{\partial (u)}{\partial x} + v\frac{\partial (u)}{\partial y}\Big] = -\frac{1}{\rho}\frac{\partial P}{\partial x} + \nu\Big[\frac{\partial^2 (u)}{\partial x^2} + \frac{\partial^2 (u)}{\partial y^2}\Big] + f_x, \tag{1.3b}$$

$$\frac{\partial v}{\partial t} + \frac{1}{2}\Big[\frac{\partial (uv)}{\partial x} + \frac{\partial (v^2)}{\partial y} + u\frac{\partial (v)}{\partial x} + v\frac{\partial (v)}{\partial y}\Big] = -\frac{1}{\rho}\frac{\partial P}{\partial x} + \nu\Big[\frac{\partial^2 (v)}{\partial x^2} + \frac{\partial^2 (v)}{\partial y^2}\Big] + f_y, \tag{1.3c}$$

where u, v, are the stream-wise and wall-normal velocity components. f_x and f_y are the body forces. $Re = 4200$ and $Re_\tau = 180$. For time integration process, a third-order Adams Bashforth scheme is used. For computing the first and second order derivatives, a sixth-order compact finite difference scheme is employed. Periodic boundary conditions are applied to the inlet and outlet of the channel. Two simulations are performed: a low resolution case with grid size chosen to be 128 by 129 by 84 and a high resolution case with grid size chosen to be 256 by 257 by 168. The grid points are equally spaced in both stream-wise and span-wise direction. However, a non-uniform grid system is used in the wall-normal direction where the grid is stretched from both wall to the middle. For the purpose of analysis, the snapshot range of the data-set has been chosen carefully to ensure that the flows are fully developed.

CHAPTER 2

SPARSE CONVOLUTION MODELS FOR NONLINEAR DYNAMICS

2.1 Motivation and Review

Model reduction techniques have evolved rapidly in the past few decades with the rising need for identification, control, and optimization of complex systems. In particular, low-dimensional models that can accurately and efficiently identify the underlying dynamics for high-dimensional fluid flow systems is often desired. This becomes challenging when building the predictive models using highly sparse measurements without knowledge of the governing equation and/or boundary conditions. Consequently, the desired methods need to be primarily data-driven, handle nonlinear dynamics and offer robust performance when the measurements are sparse. The most widely used class of data-driven models are linear such as DMD (Schmid 2010). Although these methods appear to be unreliable for modeling non-linear systems as pointed out in Taira et al. (2017), they can be linked to the Koopman operator theoretic framework (Koopman 1931), which makes them promising for modeling non-linear systems. For example, Rowley & Dawson (2017) have evolved the original DMD framework based on the Koopman framework to successfully capture the transient growth of the flow instability in the wake of the cylinder. In comparison, DMD as a linear model is insufficient for capturing such transient dynamics. The essential idea of the Koopman theoretic framework is that under an optimal basis transformation to so-called observable function space, a finite-dimensional but non-linear dynamical system evolves as an infinite-dimensional linear system in that Hilbert space (Koopman 1931, Mezić 2005, Rowley et al. 2009). The Koopman theory implies that non-linear systems can

be modeled as a linear one if the appropriate set of observable functions can be identified. However, obtaining the correct observable functions is rather challenging if not impossible since the information about the underlying dynamical system is not always known. In this chapter, a sparse generalized convolution framework is presented as a means to mapping to the Koopman observable space that can be leveraged for data-driven modeling of nonlinear fluid flow systems.

2.2 Objective and Contribution

The objective of this study is to model and predict nonlinear fluid flow dynamical systems accurately and efficiently from data. The contribution from this dissertation is a generalized convolution framework that allows one to potentially map from the state space to the Koopman observable space. The underlying principle of this convolution framework is that complex mappings can be represented, at least in theory, by layer multiple convolution operators on top of each other. In this study, the focus is primarily on convolution operators based on POD bases and Gaussian kernels. The Gaussian Processes (GP) convolution is inspired by the E-GP (Kingravi et al. 2015, 2016) framework in the machine learning community, and the matlab toolbox for learning E-GP algorithm can be found in *Evolving Gaussian Processes and Kernel Observers* (n.d.). Such multi-layer convolution operators can potentially allow for the prediction of the non-linear dynamics in complex flow systems. In conjunction with the convolutions, the strategies for optimal sparse representation based on the user-defined criteria are developed. The POD basis functions are derived from the constraint that they capture most of the energy content in the data. Other studies (Chen et al. 2012, Wynn et al. 2013, Jovanović et al. 2014) have focused on applying different user-defined constraints to compute the optimal basis which is not the focus of this research effort. For Gaussian Processes convolution, the Gaussian kernels are constructed at centers that are learned from different methods: k-means

clustering (Bishop 2007) with divided domains, k-means clustering combined with local variability, and Sparse Online Gaussian Processes (Csató & Opper 2002). These methods will be explained in detail in the subsequent sections.

The content of this chapter can be structured as follows. In the following section, the Koopman operator theory and its connections to the sparse convolution framework will be discussed. In section 2.4, a generalized sparse convolution framework is presented. In section 2.5, the strategies for modeling the dynamical system is explored using different sparse representation. In section 2.6, the effectiveness of the proposed framework for modeling the cylinder and Boussinesq flows is examined.

2.3 The Koopman Operator Theory

In this section, the Koopman operator theory is briefly introduced and explained for modeling non-linear system. Consider a discrete-time dynamical fluid flow system:

$$y = \mathcal{F}(x), \tag{2.1}$$

where x and y are N-dimensional state vectors, e.g., velocity components, and separated by an appropriate unit of time Δt. The operator $\mathcal{F}$ is non-linear and finite-dimensional which evolves state x to state y in time. Note that the continuous-time formulation can be equivalently represented. From the Koopman theory (Rowley et al. 2009), an infinite-dimensional but linear operator $\mathcal{K}$ is acting on the observable function g as:

$$\mathcal{K}g(x) = g(y), \tag{2.2a}$$

$$\mathcal{K}g(x) = g(\mathcal{F}(x)). \tag{2.2b}$$

$\mathcal{K}$ is called the Koopman operator that governs the linear evolution of the observable function g, which is a scalar valued function of x or y. As pointed out by Williams et al. (2014) and many others (Williams et al. 2015, Rowley & Dawson 2017), the Koopman operator maps the observable function $g(x)$ to $g(y)$, not the state vector

x to y. Function $g(x)$ can be represented as a linear combination of the Koopman eigenfunctions (ϕ_j) and eigenmodes (v_j) as shown in Equation 2.3. Furthermore, combined with the Koopman eigenvalues (μ_j) which are often referred to the Koopman tuples, they enable to predict the observable function in the future time as shown in Equation 2.4. The eigenmodes represent the spatial structures of the dynamical system, whereas the eigenfunctions provide the weighting coefficients that are required to reconstruct the observable function $g(x)$. The eigenvalues contain the temporal information that allows evolving $g(x)$ in time.

$$g(x) = \sum_{j=1}^{\infty} \phi_j v_j \tag{2.3}$$

$$g(y) = \mathcal{K}g(x) = \sum_{j=1}^{\infty} \phi_j v_j \mu_j \tag{2.4}$$

As mentioned before, current methods for approximating the Koopman tuples are limited and based on DMD and its derivatives EDMD, and KEDMD. For the detailed discussion on the Koopman theory, tuples, and approximation techniques, the readers are encouraged to explore a vast amount of articles (Rowley et al. 2009, Schmid 2010, Mezić 2013, Williams et al. 2014, 2015).

The key to accurately capturing the system dynamics using a linear model relies heavily on knowing the observable function g. Ideally, one can choose an infinite-dimensional set of observations of the state vectors and use them to obtain a perfect model. However, the computation will be intractable. Then the question is how to identify an appropriate finite set of observable functions that can capture the relevant dynamics. One approach is to build g using a rich choice of basis functions of the state vector x, e.g., polynomials such as x, x^2, or more complex functions such as $sin(x)$, e^x, etc. The hope is that such functions may be sufficient to capture the dominant underlying dynamics. However, for systems that are highly non-linear, building appropriate g while not impossible is extremely challenging and problematic because of the need to identify 'magic' functions. Not any observable functions would

suffice for learning the Koopman operator $\mathcal{K}$ that is linear and yet can describe the non-linear dynamics. With this brief introduction to the Koopman theory, the central focus of the research presented in this chapter is to identify a framework for building observable functions in the form of a convolution operator, which is a mapping to the feature (weights) space. To this end, a generalized sparse convolution framework is proposed for modeling non-linear fluid flow systems in the following section.

2.4 A Generalized Sparse Convolution Framework

2.4.1 Basic Formulations

To describe this framework, the notations from section 2.3 are adopted as much as possible. Given pairs of snapshot data: $X = [x_1, x_2, ..., x_M]$ and $Y = [y_1, y_2, ..., y_M]$, where $X, Y \in \mathbb{R}^{N \times M}$. N is the total number of dimensions, and M is the total number of snapshots for the data. To approximate the true non-linear operator $\mathcal{F}$, which governs the time evolution from x_i to y_i $(i = 1, 2, 3, ..., M)$, a operator A is mapping the snapshot data X to Y in time. The approximated model can be represented as below:

$$AX = Y. \tag{2.5}$$

Similar to the model in Equation 2.5, the Koopman theory states that for an appropriate observable functions g that acts on X and Y:

$$\mathcal{K}g(X) = g(Y), \tag{2.6}$$

and a linear and infinite-dimensional operator $\mathcal{K}$ is evolving $g(X)$ to $g(Y)$. By comparing Equation 2.5 and 2.6, the Koopman representation is mapping X and Y to an observable function space. Equation 2.6 is the ideal dynamical model that one can obtain. However, the observable function space is infinite dimensional and therefore computationally intractable. To overcome this, an alternative approach is to come up with an appropriate finite-dimensional representation in the form of a convolution

operator. The underlying principle being that the observable function space g can be approximated by a finite-dimensional convolution operator. The accuracy and generality of this approximation needs to be evaluated for the nonlinear physics of interest. To accomplish this, a convolution operator $C \in \mathbb{R}^{N \times K}$ is defined as follows.

$$X = C\bar{X}, \tag{2.7}$$

$$Y = C\bar{Y}, \tag{2.8}$$

where $\bar{X} \in \mathbb{R}^{K \times M}$ and $\bar{Y} \in \mathbb{R}^{K \times M}$ are the associated convolved weights for X and Y, and K is the number of weights. Equation 2.5 can be written as:

$$AC\bar{X} = C\bar{Y}. \tag{2.9}$$

Pre-multiplying with the pseudo-inverse of C, Equation 2.9 can be rearranged as:

$$C^{+}AC\bar{X} = \bar{Y}. \tag{2.10}$$

Next, $\bar{A} = C^{+}AC$ is defined as the convoluted system operator such that:

$$\bar{A}\bar{X} = \bar{Y} \tag{2.11}$$

where $\bar{A} \in \mathbb{R}^{K \times K}$, $\bar{X}$ and $\bar{Y}$ are the finite approximation forms of the Koopman operator, the observable function $g(X)$, and $g(Y)$, respectively. Equation 2.11 is the foundation of this convolution framework. In principle, if these approximations of the observable functions $g(X)$ and $g(Y)$ are appropriate for the dynamics of interest, the convolved linear system operator $\bar{A}$ will be an accurate approximation of the Koopman operator $\mathcal{K}$. Moreover, the choice of the convolution operator allows for dimensionality reduction (when the state vector $X \in \mathbb{R}^{N}$ is high-dimensional) which makes the resulting system operator $\bar{A} \in \mathbb{R}^{K \times K}$. In the convolved space, the computational cost to learn $\bar{A}$ is greatly reduced if K is considerably smaller than M and N. The key aspect of this approach is to choose convolution operator C such that the

convolved system operator $\bar{A}$ captures most if not all of the relevant system dynamics. Particularly, for complex fluid flows, choosing the correct form of this operator is essential for capturing the various non-linear dynamics such as bifurcation, instability modes, shock formations, and turbulence. Hence, the optimal choice of convolution operator could be problem dependent.

2.4.2 The POD Convolution as DMD

Dynamic mode decomposition is a popular method for learning the Koopman operator that employs a linear convolution operator as first developed by Schmid (2010). In the DMD algorithm, Schmid efficiently employs the SVD as a 'POD convolution'. It is important to note that computing the SVD does not require subtracting the mean component out of the system as is done in the traditional POD method. The DMD technique can be introduced by first computing the SVD of X as:

$$X = U\Sigma W^T \tag{2.12}$$

and similarity transformation of A to map from physical space into low dimensional space:

$$U^T AU = U^T YW\Sigma^{-1}, \tag{2.13}$$

Define a reduced operator $\tilde{A} = U^T AU$, Equation 2.13 becomes:

$$\tilde{A} = U^T YW\Sigma^{-1}. \tag{2.14}$$

Rearrange Equation 2.13:

$$\tilde{A}\Sigma W^T = U^T Y \tag{2.15}$$

where ΣW^T and $U^T Y$ are equivalent to $\bar{X}$ and $\bar{Y}$ in Equation 2.11. The connection between DMD and the generalized convolution framework presented here is that the reduced operator $\tilde{A}$ above is equivalent to the convolved system operator $\bar{A}$. The convolution operator C used in this example is the left singular vector U, which represent

the POD modes if the mean of the data is taken out during the SVD computation. Because of the connection of these singular vectors to the more common POD modes in the fluids community, this approach is referred to as the POD-convolution. The benefits of using SVD or POD modes are two-fold. The first is it generates low-dimensional basis that captures optimal energy content of the given data-set as long as the system is sparse in this basis. For many common fluid flows, e.g., flow past a cylinder, the POD basis is typically sparse. Furthermore, one can employ truncation of the modal basis to reduce the dimensionality. With POD-convolution, the dimension K is the number of the POD modes retained. Secondly, since the modes are orthogonal one can build exact deconvolution operator C^+ through a simple transpose, i.e., $C^+ = C^T$ or $U^+ = U^T$. A downside to this method is that these POD modes (singular vectors) are data dependent and normally includes only a subset of the feature space that is represented in the training data. However, the convolution framework is not limited to using the singular vectors or POD modes. Other operators can be built using kernels such as Fourier functions, wavelets and Gaussian Processes (GP) which is introduced in the following section.

2.4.3 Gaussian Processes Convolution

In the machine learning community, kernel-based regression, such as those using Gaussian Processes (GP) (Bishop 2007, Kingravi et al. 2015, 2016), Random Kitchen Sinks(RKS) (Rahimi & Recht 2009), and Fastfood (Le et al. n.d.), are commonly used as convolution operators to build predictive models for dynamical systems. More importantly, these kernel functions can be generalized within this convolution framework. However, their suitability for capturing fluid flow dynamics is not established. In this study, the primary focus is on Gaussian functions (kernels) as employed within the convolution framework. The GP-convolution is presented in Equation 2.16.

$$C_{ij} = C(z_i, \bar{z}_j) = \exp \frac{-||z_i - \bar{z}_j||^2}{2\sigma^2}, \tag{2.16}$$

where $C \in \mathbb{R}^{N \times K}$, i.e., $i = 1, 2, ..., N$ and $j = 1, 2, ..., K$. In the above formulation, z is a vector of spatial locations where all the state information of the flow system is available, i.e., $x = x(z, t)$, and $\bar{z}$ is a vector of predetermined spatial centers. σ is the hyper-parameter that determines the width of the GP kernels. To summarize, GP-convolution is a kernel regression of the state vector representing spatiotemporal data at predetermined locations in space, called centers, with pre-specified hyperparameter σ.

The centers $\bar{z}_j$ are identified by analyzing the available snapshot data and can be viewed similarly to identifying locations for measurement probes in lab experiments. There are many well-known techniques to identify the location of the centers including those that use POD-modes (Cohen et al. 2003), k-means clustering (Bishop 2007), Sparse Online Gaussian Processes (Csató & Opper 2002), amongst others. If the system is well known, one can choose the centers where the interesting dynamics occur, e.g., maxima or minima of the POD modes (Cohen et al. 2003). However, that requires computing the gradient of the POD modes, which can be non-trivial at times if the structure of the data is not known in advance. On the other hand, k-means clustering is a statistical technique that identifies the centers by minimizing the distance between all the physical points and the chosen centers (Bishop 2007) and is iterative. It is well known that the centers computed with this method tend to be biased towards the regions with a higher density of the data. Typically, the number of centers(weights) K is chosen to be much smaller than the dimension N which allows for a low-dimensional and sparse representation of the state vector. It is beneficial if one deals with a high-dimensional system. Compare to POD-convolution, GP-convolution does not require computing an eigen-decomposition, which is time efficient if the number of snapshots M is larger than N. The disadvantage of using GP-convolution is that the choice of the centers and the hyperparameters may cause errors during deconvolution, called the aliasing effect. Specifically, the convolution

operator C could be singular and may not have an exact inverse. Hence, applying regularization techniques to solve C^+ tends to alter the convolved weights and distort the flow field. Illustrative examples will be provided to demonstrate the aliasing effect in section 2.6.

2.4.4 Multi-layer Convolution

The convolution proposed in this dissertation is not restricted to POD modes or GP functions. In fact, the novelty of this work lies in the realization that convolution operators can be built in a variety ways as follows. An easy approach is to replace z, the spatial locations in Equation 2.16 with x, the state information which could be velocity or pressure or temperature components. This implies that an non-linear convolution operator C is built, where $C_{ij} = C(x_i, x_j)$. However, building a convolution operator correctly is a challenging task since the representative mapping, which is effective for capturing the dynamics of interest, is not always known in advance. A more generalized way for building convolution operator is to recursively layer multiple convolution operators such as:

$$X = C_L ... C_3 C_2 C_1 \bar{X}, \tag{2.17}$$

$$Y = C_L ... C_3 C_2 C_1 \bar{Y}. \tag{2.18}$$

Similarly, substituting Equation 2.17 and 2.18 into Equation 2.11:

$$AC_L ... C_3 C_2 C_1 \bar{X} = C_L ... C_3 C_2 C_1 \bar{Y}. \tag{2.19}$$

Pre-multiplying the pseudoinverse of each convolution operator: C_1, C_2, C_3, ..., C_L, Equation 2.19 can be rearranged as:

$$C_1^+ C_2^+ C_3^+ ... C_L^+ AC_L ... C_3 C_2 C_1 \bar{X} = \bar{Y} \tag{2.20}$$

with a convoluted system operator represented as:

$$\bar{A} = C_1^+ C_2^+ C_3^+ ... C_L^+ AC_L ... C_3 C_2 C_1. \tag{2.21}$$

Equation 2.21 can be further simplified with $C_l = C_1^+ C_2^+ C_3^+ ... C_L^+$ and $C_r = C_L ... C_3 C_2 C_1$, where C_l and C_r are the "global" convolution operator. The above formulation represents the idea behind the generalized convolution framework where layers of convolution operators are expected to replace the 'magic functions' needed for effective modeling. As examples, one can layer both POD-convolution and GP-convolution to form a two-layer convolution. A GP-POD-convolution can be built such that C_1 is GP-convolution operator, and C_2 is POD-convolution operator. Similarly, a POD-GP-convolution is possible by reversing the order of the operators. In this thesis, the focus is on GP-POD-convolution instead.

The multi-layer convolution can also include nonlinear transfer functions (TF), especially if prior knowledge of the underlying physics is available. For example, Rowley & Dawson (2017) propose an innovative idea to generate appropriate observable functions by first performing POD-convolution on the flow states to extract POD weights, i.e., $\bar{X} = [\bar{x}_1 \ \bar{x}_2 \ , ..., \ \bar{x}_R]^T$ and $\bar{Y} = [\bar{y}_1 \ \bar{y}_2 \ , ..., \ \bar{y}_R]^T$. In order to model the non-linearity existing in the system, they build a sufficiently rich vector of basis functions by combining the linear and quadratic-non-linear terms of POD weights into $\bar{X}$ and $\bar{Y}$ such that: $\bar{X} = [\bar{x}_1 \ \bar{x}_2 \ , ..., \ \bar{x}_R \ \bar{x}_1\bar{x}_1 \ \bar{x}_1\bar{x}_2 \ , ..., \ \bar{x}_R\bar{x}_R]^T$ and $\bar{Y} = [\bar{y}_1 \ \bar{y}_2 \ , ..., \ \bar{y}_R \ \bar{y}_1\bar{y}_1 \ \bar{y}_1\bar{y}_2 \ , ..., \ \bar{y}_R\bar{y}_R]^T$. This approach can be related to the framework above as an example of a non-linear convolution operator without explicitly specifying C_2 which can be viewed as POD-TF-convolution (POD with transfer functions). This approach performs well if the dynamical system requires the inclusion of quadratic nonlinearities. However, if the dynamics are unknown, a generalized approach such as the one proposed in this section may be more reasonable. It is worth noting that this generalized approach has similarities to artificial neural networks in deep learning where multiple hiddlen layers are used to improve predictions of complex nonlinear data. Furthermore, the rich set of basis functions required to sufficiently approximate the dynamics of interest can become too large such that the compu-

tation becomes intractable. Possible solution is to employ the kernel trick (Bishop 2007) which is essentially equivalent to adding an extra convolution layer implicitly. This approach has been tried by Williams et al. (2014), and who demonstrate success by using polynomial kernel function as shown in Equation 2.22. Alternatively, radial basis function kernels shown in Equation 2.23 can also be employed to build the mapping. However, this requires specification of a hyper-parameter β which acts as a regularizer. In section 2.6.2, the performance of a polynomial kernel function with POD-convolution, termed as POD-PolyK-convolution, is demonstrated.

$$f(x, y) = (1 + y^T x)^\alpha \tag{2.22}$$

$$f(x, y) = \exp \frac{-||x - y||^2}{2\beta^2} \tag{2.23}$$

2.4.5 Computing the Approximated Koopman Operator

Having developed the above framework, the Koopman operator $\bar{A}$ in Equation 2.11 can be approximated as follows. Recall that $\bar{A}\bar{X} = \bar{Y}$ where $\bar{A} \in \mathbb{R}^{K \times K}$ and X,$Y \in \mathbb{R}^{K \times M}$. The key to compute $\bar{A}$ depends on the dimensions M and K. Taking the transpose of Equation 2.21:

$$\bar{X}^T \bar{A}^T = \bar{Y}^T \tag{2.24}$$

If $K < M$, an overdetermined system for which Equation 2.24 becomes a least-squares problem is obtained. To solve such a problem, pre-multiplying by $\bar{X}$ as shown below:

$$\bar{X}\bar{X}^T \bar{A}^T = \bar{X}\bar{Y}^T. \tag{2.25}$$

In Equation 2.25, $\bar{X}\bar{X}^T$ becomes a square matrix of size $K \times K$ that is invertible if it is full rank. Then pre-multiply Equation 2.25 with the inverse of $\bar{X}\bar{X}^T$ to get:

$$\bar{A}^T = (\bar{X}\bar{X}^T)^{-1}\bar{X}\bar{Y}^T, \tag{2.26}$$

where $\bar{A}$ can be obtained from taking the transpose of Equation 2.26. When $K > M$, an underdetermined system that has infinite number of solutions is obtained. In this

case, a $M \times M$ matrix $\bar{X}^T\bar{X}$ is generated which can be inverted if it is full rank. The corresponding approximated Koopman operator can be solved found Equation 2.27.

$$\bar{A}^T = \bar{X}(\bar{X}^T\bar{X})^{-1}\bar{Y}^T \tag{2.27}$$

Note that the approximations for $\bar{A}$ in Equations 2.26 and 2.27 may require some form of regularization for matrix inversion. In this study, a Tikhonov regularization method with parameter γ is exploited as shown in Equations 2.28 and 2.29.

$$\bar{A}^T = (\bar{X}\bar{X}^T + \gamma I)^{-1}\bar{X}\bar{Y}^T, \tag{2.28}$$

$$\bar{A}^T = \bar{X}(\bar{X}^T\bar{X} + \gamma I)^{-1}\bar{Y}^T \tag{2.29}$$

2.5 Sparse Representation

In this section, the strategies of sparse representation to reduce the computational complexity of modeling dynamical systems are presented. For POD-convolution, sparsification occurs in the basis space by optimizing the energy capture for the training data. A straightforward approach is to retain the POD modes with higher energy content and truncate the ones with very little energy. However, energy-based truncation can be unreliable when the low energy modes are dynamically important. However, if energy capture is not the primary goal, other techniques for optimal mode selection in the basis space can be employed. For instance, Chen et al. (2012) propose an optimized basis selection framework that relates the number of modes to minimize error in each data snapshot. This method is advantageous if one intends to retain fewer mode but has no clear knowledge on how to truncate the basis in a more complex system. Similar to Chen et al. (2012), Wynn et al. (2013) have also attempted to minimize the error in each data snapshot by constructing optimal singular vectors and transformed low dimensional system matrix using matrix manifold theory (Absil et al. 2009, Edelman et al. 1998, Goulart et al. 2012). As a result, they have achieved

a better approximation of both synthetic and experimental data as compared to DMD by extracting more accurate eigenvalues and modes. Moreover, this method is particularly useful for the system that contains high-frequency contributions, e.g., turbulent flows. Jovanović et al. (2014) optimize the amplitude of the DMD modes by introducing a user-specified regularization parameter that accommodates mode truncation and approximation accuracy using convex optimization techniques (Boyd & Vandenberghe 2004, Boyd et al. 2011). The optimization method that Jovanović et al. (2014) used has proved to be effective in identifying the dominant structures for various fluid flow systems with limited sparsity. Although these sparse basis selection techniques have demonstrated their success and usefulness in many of the fluid flow applications, the focus on the energy optimization is to employ truncation method for POD-convolution in the following discussion.

In contrast to the basis optimization in POD-convolution, Gaussian Process regression provides a framework for sparsification in the physical space. This idea originates from the need to deal with sparse measurement data or generate quick on-demand models from very few sensors for fast online decision making. In GP-convolution, the optimal sparse representation is achieved by identifying optimal sensor locations in space where data is made available with a sufficient number of sensors. The resulting model built from GP-convolution can hopefully provide an accurate and efficient representation of the overall field in a least-squares sense. The performance of this approach depends heavily on the choice of centers, i.e., both location and quantity of the sensors, and the user-specified hyper-parameter. The research performed for this dissertation adopts three different algorithms for learning centers: k-means clustering with divided domains, k-means clustering with local variability, and Sparse Online Gaussian Processes (Csató & Opper 2002). The terminology chosen is as follows: GP-convolution with centers learned from k-means clustering with divided domains as GP-k-convolution, centers learned from k-means clustering with

local variability as GP-kv-convolution, and centers learned from Sparse Online Gaussian Processes algorithm as GP-sv-convolution.

For the k-means clustering with divided domains, the idea is to apply the k-means algorithm on the divided domains. The k-means algorithm can be found in Bishop (2007), and Algorithm 1 summarizes the detailed procedure. K-means algorithm is basically to locate k number of points that have the minimum spatial distance to the associated grid points. To divide the domain, more centers are placed in the region where the important dynamics are observed. For flow past a cylinder, the wake of the cylinder is the region that exhibits the most interesting dynamics - vortex shredding and thus garners the most centers as shown in Figure 2.1. Specifically, the domain is divided into three regions: the wake of the cylinder(250 centers), around the cylinder(30 centers), and top and bottom side(20 centers).

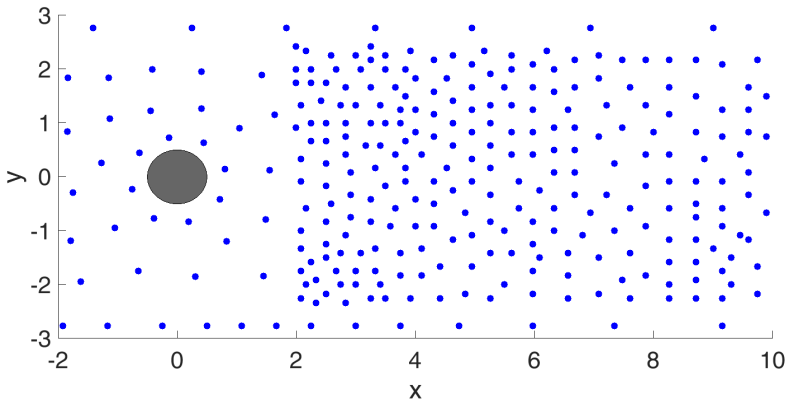


Figure 2.1: Centers placement from the k-means clustering algorithm with divided domains for cylinder flow at $Re = 100$. 300 centers is chosen in this case.

However, the user-defined choice may not alway produce the most accurate representation. Therefore, in an alternate approach, k-means is combined with local temporal variability of the velocity field to identify the optimal centers locations where the physics varies most over time. Similar to the previous approach, the k-means algorithm is used to determine centers for the regions around the cylinder, top

and bottom sections with 30 and 20 locations, respectively. Concurrently, local variability with velocity components calculated in Equation 2.30 is used to find another 250 centers.

$$\sigma_b = \frac{1}{M}\frac{1}{N_b}\sum_{j=1}^{M}\sum_{i=1}^{N_b}(u_{i,j} - \bar{u_i})^2 \qquad b = 1, 2, ..., B \tag{2.30}$$

where B is the number of bins specified. In this example, the k-means clustering algorithm is used to compute 400 bins, i.e., $B = 400$. M is the number of snapshots, and N_b is the number of grid points in the b^{th} bin. $\bar{u}$ is the ensemble average. The objective is to select the first 250 largest σ from these 400 bins for the centers. This results in 250 reasonably concentrated centers in the region of the cylinder wake as shown in Figure 2.2. The detailed algorithmic procedure for local variability is shown in Algorithm 2. The remaining 50 centers are computed from the k-means algorithm as mentioned before.

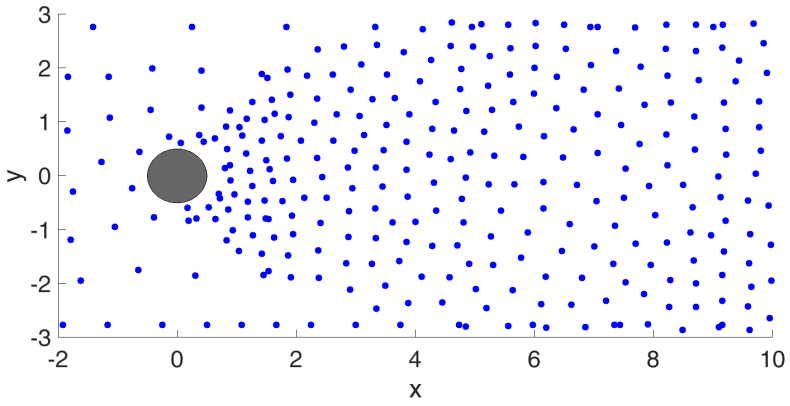


Figure 2.2: Centers placement learned from k-means clustering with local variability algorithms for cylinder flow at $Re = 100$. 300 centers is chosen in this case.

Sparse Online Gaussian Processes is an improved algorithm that identifies the centers in a real-time setting using a field of choice (in this case velocity components) such that the linear dependence of the data from one snapshot to another is minimized. In this study, Sparse Online Gaussian Processes algorithm from the Matlab toolbox (*Evolving Gaussian Processes and Kernel Observers* n.d.) is utilized

for locating the centers, which are shown in Figure 2.3.

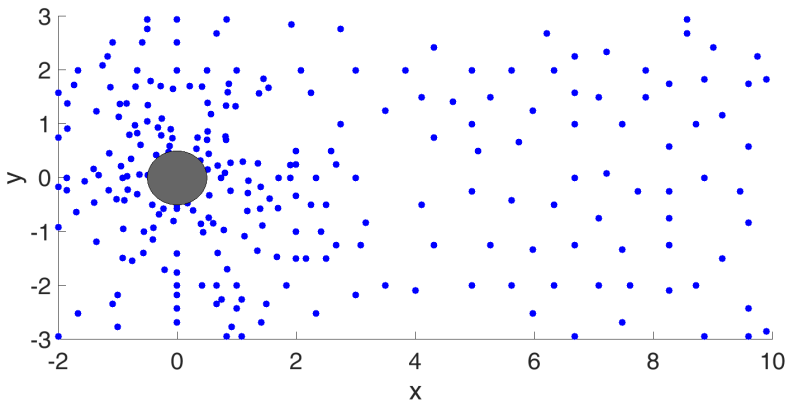


Figure 2.3: Centers placement learned from Sparse Online Gaussian Processes algorithm for cylinder flow at $Re = 100$. 300 centers is chosen in this case.

In subsection 2.6.2, the effectiveness of the center placements from these learning algorithms as well as a different number of centers(300 and 600) for modeling flow past a cylinder will be explored. All of the above cases are illustrated with 300 centers for demonstration purposes.

Algorithm 1: K-mean clustering algorithm to learn centers

input : Spatial points $[X, Y] \in \mathbb{R}^{N\times 2}$, $Tolerance$

output: Center points $[\overline{X}, \overline{Y}] \in \mathbb{R}^{K\times 2}$

N is number of spatial points; K is number of center points

Randomly choose K centers, $\tilde{X}$ and $\tilde{Y}$ from X and Y

while $maxdis > Tolerance$ **do**

  for $i \leftarrow 1$ **to** N **do**

    for $j \leftarrow 1$ **to** K **do**

      $D_{ij} = \sqrt{(Xi - \tilde{X}_j)^2 + (Yi - \tilde{Y}_j)^2}$

    end

  end

  $Npts = zeros(K, 1)$

  for $i \leftarrow 1$ **to** N **do**

    $Index(i) = j$ .st. $D_{ij} = min(D_{i:})$; $\quad Npts(j) = Npts(j) + 1$

  end

  $\overline{X} = \vec{0}$; $\overline{Y} = \vec{0}$

  for $i \leftarrow 1$ **to** N **do**

    $\overline{X_{Index(i)}} = \overline{X_{Index(i)}} + X_i$; $\quad \overline{Y_{Index(i)}} = \overline{Y_{Index(i)}} + Y_i$

  end

  for $j \leftarrow 1$ **to** K **do**

    $\overline{X_j} = \overline{X_j}/Npts(j)$; $\quad \overline{Y_j} = \overline{Y_j}/Npts(j)$

  end

  for $j \leftarrow 1$ **to** K **do**

    $distance_j = \sqrt{(\overline{X_j} - \tilde{X}_j)^2 + (\overline{Y_j} - \tilde{Y}_j)^2}$

  end

  $maxdis = \mathbf{max}(distance)$

end

Algorithm 2: Local variability algorithm to learn centers

input : Spatial points $[X, Y] \in \mathbb{R}^{N\times 2}$, Velocity field $U \in \mathbb{R}^{N\times M}$

output: Center points $[\overline{X}, \overline{Y}] \in \mathbb{R}^{K\times 2}$

N is number of spatial points; K is number of centers; M is number of snapshots; B is number of bins; N_b is the number of spatial points in b^{th} bins.

Use k-means algorithm to determine B number of bins

Find the ensemble average $\bar{u}$

Find the variance for each bins:

for $b \leftarrow 1$ **to** B **do**

    $\sigma_b = 0$

    for $j \leftarrow 1$ **to** M **do**

        for $i \leftarrow 1$ **to** N_b **do**

            $\sigma_b = \sigma_b + (u_{i,j} - \bar{u}_i)^2$

        end

    end

    $\sigma_b = \frac{1}{M}\frac{1}{N_b}\sigma_b$

end

Sort σ and pick the K largest values and their corresponding spatial points in X and Y as centers $[\overline{X}, \overline{Y}]$.

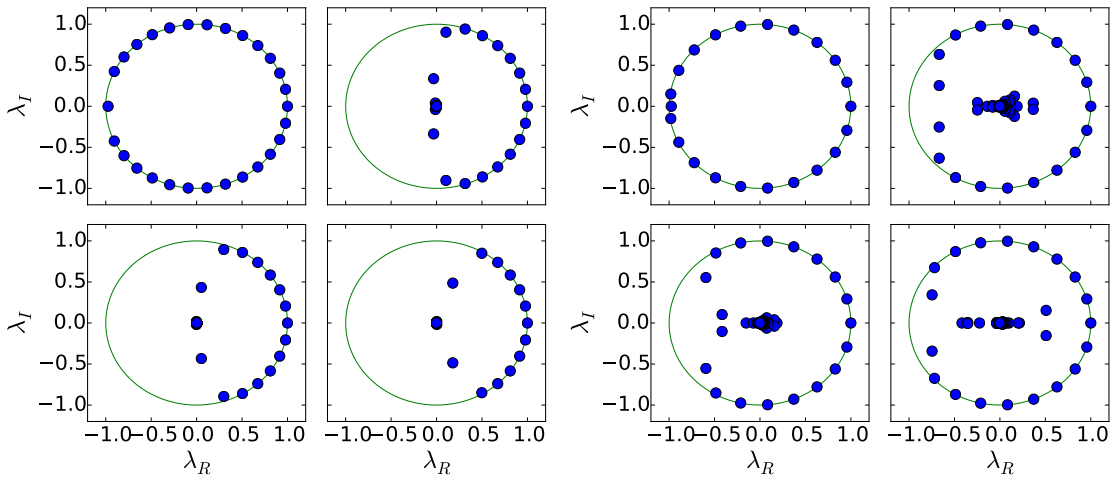
2.6 Results and Discussion

2.6.1 Koopman Operator Approximation

For the temporal regime considered, the wake of the cylinder shows the cyclical vortex shedding which represents a dynamical system operating on a limit cycle. In principle, such a system is expected to predicted accurately using a model due to its repetitive and periodic behavior. Such a cylinder wake flow is considered as a

baseline case to validate the sparse convolution framework. The data-set chosen for the analysis is at Reynolds numbers, $Re = 100$ and $Re = 1000$. The ideas of GP-k-convolution, GP-kv-convolution, and GP-sv-convolution are demonstrated by extracting the leading Koopman eigenvalues and eigenmodes. To prove that GP-convolutions approximate the Koopman operator, these results are compared with those from the POD-convolution.

The associated eigenvalues are shown in Figure 2.4a and 2.4b in the form of real and imaginary parts plotted with a unit circle for $Re = 100$ and $Re = 1000$, respectively.



(a) $Re = 100$ (b) $Re = 1000$

Figure 2.4: Eigenvalues computed from $\bar{A}$ for the periodic cylinder flow using POD-convolution(top left), GP-k-convolution(top right), GP-kv-convolution(bottom left), and GP-sv-convolution(bottom right)

From Figure 2.4a and 2.4b, it is observed that all of the GP-convolutions are capable of capturing the leading eigenvalues. For the case of $Re = 100$, the eigenvalues are located at the right plane of the unit circle and are accurately captured for the GP-convolutions as compared to POD-convolution. For the case of $Re = 1000$, the ones that reside close to the unit circle from GP-convolutions share similar identities as

the ones from POD-convolution in Figure 2.4a. The frequency for each eigenvalues can be computed as $f = \frac{log(\lambda_I)}{\Delta t}$. The corresponding Strouhal number(St) is computed using $St = \frac{fD}{U_\infty}$, where D is the diameter of the cylinder, and U_∞ is the inlet velocity. The resulting St number for $Re = 100$ and $Re = 1000$ match reasonbly well with the simulations from several studies (Roshko 1954, Tritton 1959, Jordan & Fromm 1972, Rajani et al. 2009). The first three dominant eigenvalues and St number in are tabulated in Table 2.1 and 2.2.

Table 2.1: The first three dominant eigenvalues extracted from POD-convolution(POD), GP-k-convolution, GP-kv-convolution, and GP-sv-convolution for the periodic cylinder flows at $Re = 100$.

Convolution	Eigenvalue 1	Eigenvalue 2	Eigenvalue 3
	$St = 0.1655$	$St = 0.3311$	$St = 0.4966$
POD	$0.9784 + 0.2065i$	$0.9147 + 0.4041i$	$0.8115 + 0.5843i$
GP-k	$0.9784 + 0.2065i$	$0.9146 + 0.4041i$	$0.8113 + 0.5841i$
GP-kv	$0.9784 + 0.2065i$	$0.9145 + 0.4040i$	$0.8112 + 0.5841i$
GP-s	$0.9784 + 0.2065i$	$0.9146 + 0.4041i$	$0.8114 + 0.5842i$

Table 2.2: The first three dominant eigenvalues extracted from POD-convolution(POD), GP-k-convolution, GP-kv-convolution, and GP-sv-convolution for the periodic cylinder flows at $Re = 1000$.

Convolution	Eigenvalue 1	Eigenvalue 2	Eigenvalue 3
	$St = 0.2367$	$St = 0.3311$	$St = 0.4966$
POD	$0.9561 + 0.2930i$	$0.8283 + 0.5603i$	$0.6277 + 0.7785i$
GP-k	$0.9561 + 0.2930i$	$0.8282 + 0.5604i$	$0.6277 + 0.7784i$
GP-kv	$0.9561 + 0.2930i$	$0.8282 + 0.5603i$	$0.6277 + 0.7785i$
GP-s	$0.9561 + 0.2930i$	$0.8282 + 0.5603i$	$0.6277 + 0.7785i$

By correlating the eigenvalues from POD-convolution and GP-convolutions, the first three leading eigenvalues are accurately extracted by all four approaches for both $Re = 100$ and $Re = 1000$. The first three dominant eigenmodes are also shown in Figure 2.5, 2.6, 2.7, and 2.8 for $Re = 100$, and Figure 2.9, 2.10, 2.11, and 2.12 show the first three dominant eigenmodes for the $Re = 1000$. The zero mode, which is the mean mode, is not shown here. It is observed that the eigenmodes computed from GP-convolutions with different center placements are qualitatively similar to the results computed from POD-convolution.

The relationship between the DMD algorithm and the Koopman operator has been well-established in Tu et al. (2013), i.e., POD-convolution approximates the non-zero eigenvalues and eigenmodes of the Koopman operator. This can be further extended to argue that the leading eigenmodes and eigenvalues obtained from GP-convolutions also approximate those of the Koopman operator. This observation strengthens the connection between generalized sparse convolution framework and the Koopman theory.

It is worth noting that the eigenmodes computed directly from GP-convolution produce aliasing errors because the convolution operator C is not exactly invertible. While inverting the operator C, Tikhonov regularization is applied as shown in section 2.4.5 to suppress the near-zero singular values. However, this aliasing effect can be minimized by choosing the center locations optimally. Out of all three GP-convolutions, the case(GP-kv-convolution) with centers learned from k-means with local variability yields the best eigenmodes, i.e., the mode shapes are less distorted as compared to GP-k and GP-sv-convolution. In Figure 2.2, the resulting centers are more concentrated in the region of the cylinder wake where the flows are most active. For POD-convolution, the truncation method for selecting modes with optimal energy capture yields efficient approximations.

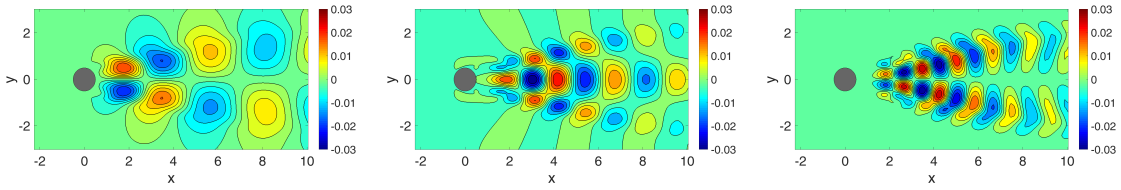


Figure 2.5: The first three eigenmodes(from left to right) using POD-convolution for the periodic cylinder flows at $Re = 100$

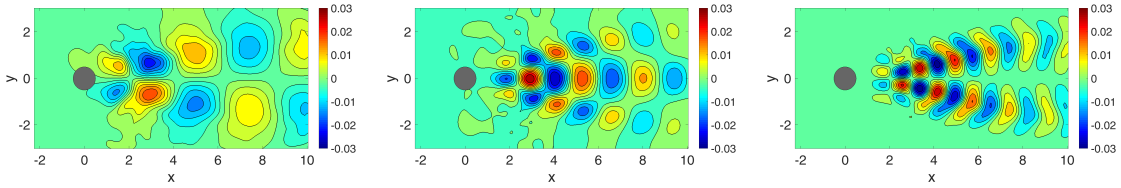


Figure 2.6: The first three eigenmodes(from left to right) using GP-k-convolution for the periodic cylinder flows at $Re = 100$.

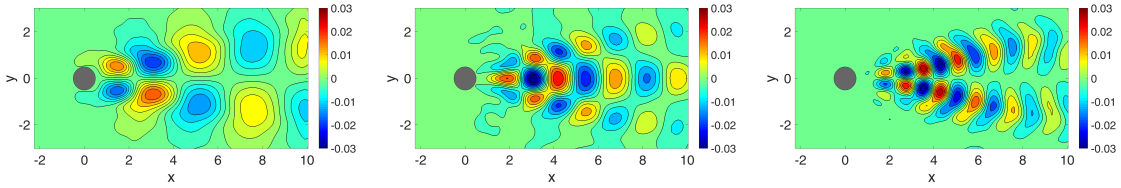


Figure 2.7: The first three eigenmodes(from left to right) using GP-kv-convolution for the periodic cylinder flows at $Re = 100$.

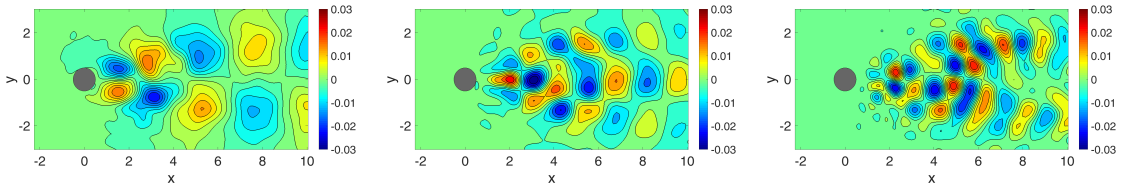


Figure 2.8: The first three eigenmodes(from left to right) using GP-sv-convolution for the periodic cylinder flows at $Re = 100$.

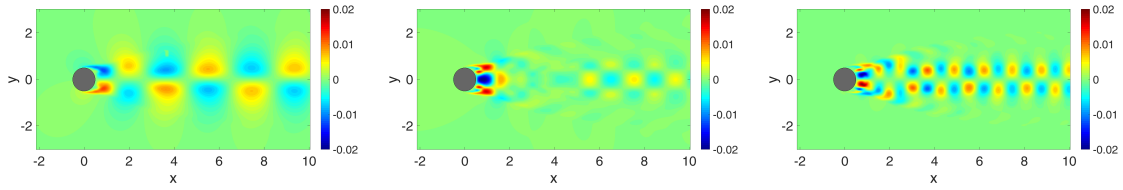


Figure 2.9: The first three eigenmodes(from left to right) using POD-convolution for the periodic cylinder flows at $Re = 1000$.

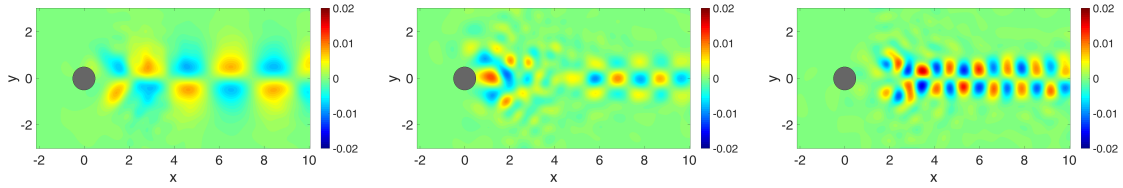


Figure 2.10: The first three eigenmodes(from left to right) using GP-k-convolution for the periodic cylinder flows at $Re = 1000$.

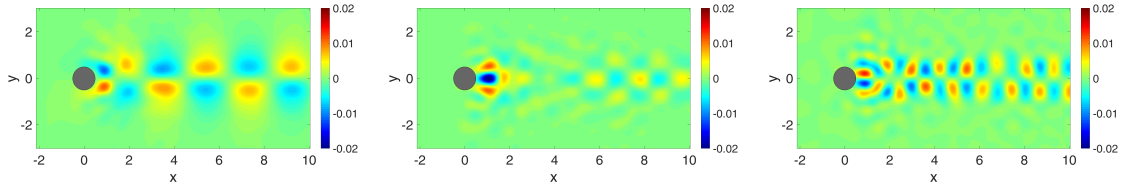


Figure 2.11: The first three eigenmodes(from left to right) using GP-kv-convolution for the periodic cylinder flows at $Re = 1000$.

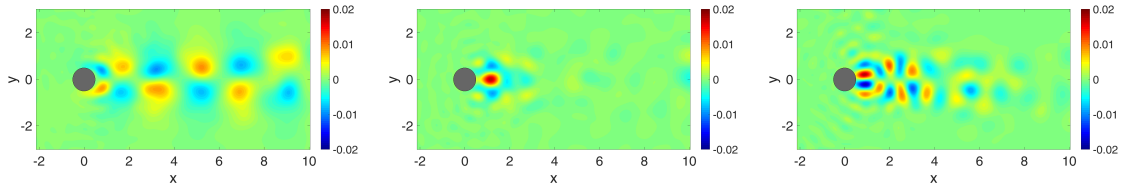


Figure 2.12: The first three eigenmodes(from left to right) using GP-sv-convolution for the periodic cylinder flows at $Re = 1000$.

2.6.2 Model Prediction

In this subsection, the focus is on investigating the effectiveness of the sparse convolution framework through model predictions for fluid flow systems with varying complexity. For this particular analysis, the periodic cylinder flow at $Re = 100$, transient cylinder flow at $Re = 100$, and Boussinesq flow at $Re = 1000$ are chosen. Individually, these flow problems exhibit different levels of non-linearity which can be utilized to examine the effectiveness of the data-driven modeling framework. The strategy is to first approximate the convolved system operator $\bar{A}$ with training data. Subsequently, the transition operator $\bar{A}$ is applied to the first snapshot of the convolved weights to predict and evolve the system in the weights space. As an illustrative example, the prediction for the future snapshot is: $\bar{x}_i = \bar{A}^{i-1}\bar{x}_1 \; i = 2, .., \text{inf}$ if $\bar{x}_1$ is denoted as the first convoluted weights and i is the snapshot index.

Periodic Cylinder Flow

For the periodic cylinder flow at $Re = 100$, a non-dimensional time unit as $T = \frac{tU_\infty}{D}$ is defined where t represents the physical time for the associated snapshot, U_∞ is the inlet velocity, and D is the cylinder diameter. In this analysis,300 snapshots ($T = 60$) of training data are chosen which corresponds to ten cycles of periodic limit-cycle behavior. To carry out an informative and comparative study, the different methods including POD-convolution, GP-k-convolution, GP-kv-convolution, GP-sv-convolution, GP-sv-POD-convolution are considered, and the resulting model is employed to predict up to 2000 snapshots ($T = 400$). An investigation of GP-convolutions with a different number of centers(300 and 600) is also conducted for this flow problem.

To illustrate predictive accuracy for the limit-cycle behavior, the projected weights and predicted weights for the cases of using GP-kv-convolution with 300 centers and POD-convolution are compared. The projected weights are computed from the full

2000 snapshots of data using the convolution operator. The evolution of the first three weights (in time) are shown in Figure 2.13 and 2.14, respectively. For brevity, the weights comparison for other GP-convolutions are not shown here, but they are also allowed to evolve periodically up to $T = 400$ as GP-kv-convolution.

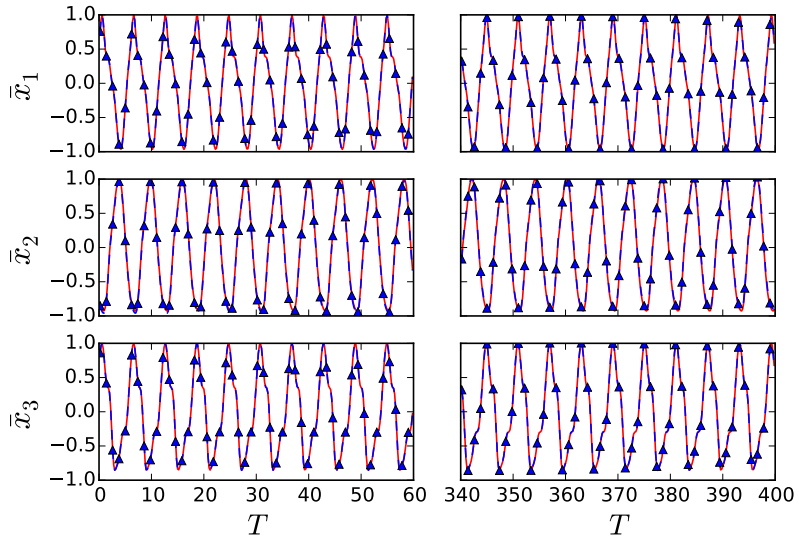


Figure 2.13: The first three projected(red line) and predicted(blue triangle) weights for the periodic cylinder flow at $Re = 100$ using GP-kv-convolution with 300 centers.

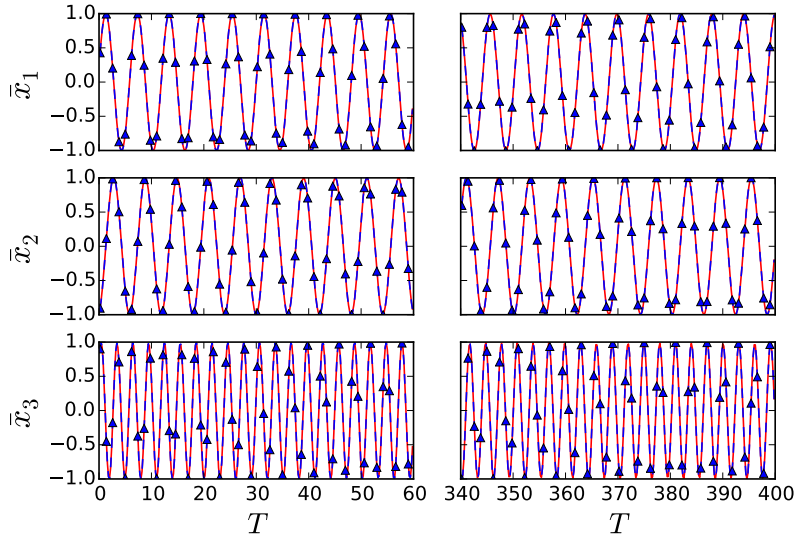


Figure 2.14: The first three projected(red line) and predicted(blue triangle) weights for the periodic cylinder flow at $Re = 100$ using POD-convolution.

To quantify the performance of the modeling framework, L^2 error norms are computed for the weight prediction as follows.

$$\phi_j = \sqrt{\frac{1}{K}\sum_{k=1}^{K}(\bar{x}_{k,j}^{Proj} - \bar{x}_{k,j}^{Pred})^2} \quad . \tag{2.31}$$

Similarly, to quantify the performance of using different convolutions, L^2 error norms for the prediction of each solution field is computed as:

$$\psi_j = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_{i,j}^{True} - x_{i,j}^{Pred})^2} \quad , \tag{2.32}$$

where i and j represent the number of dimension and snapshot for the data, respectively. From Equation 2.31, the projected weights are denoted as $\bar{x}_{k,j}^{Proj}$ whereas the predicted weights are denoted as $\bar{x}_{k,j}^{Pred}$. The true and predicted solution field are denoted as $x_{i,j}^{True}$ and $x_{i,j}^{Pred}$ in Equation 2.32. Subscripts $i = 1, 2, ..., N$, $k = 1, 2, ..., K$, and $j = 1, 2, ..., M$, where N, K, and M are the number of dimension, weights, and snapshot, respectively.

Table 2.3 and Table 2.4 summarize the numerical values of $\phi_{T=60}$, $\psi_{T=600}$, $\phi_{T=400}$, and $\psi_{T=400}$ for all the cases. The corresponding time series of the L^2 error norms for ϕ_j and ψ_j are shown in Figure 2.16 and 2.17, respectively. From both Table 2.3 and Table 2.4, it is observed that the L^2 error norms for both the weights($\phi_{T=60}$ and $\phi_{T=400}$) and field prediction($\psi_{T=60}$ and $\psi_{T=400}$) have the same order of magnitude irrespective of which center placement algorithm is chosen for GP-convolutions. The predicted solution field from POD-convolution have the highest accuracy among others as evidenced in Figure 2.17 and 2.15.

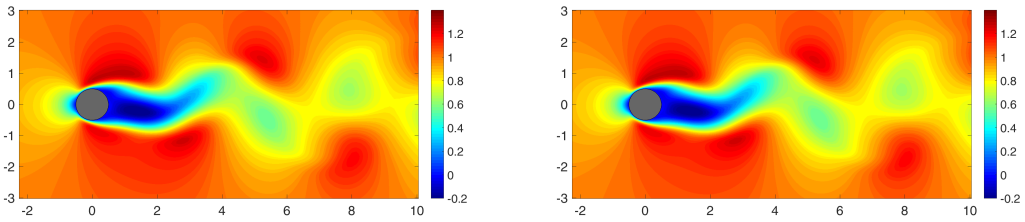


Figure 2.15: The stream-wise velocity contour at $T = 400$ for left: actual solution, right: predicted solution from POD-convolution.

Table 2.3: The L^2 error norms for the prediction of the weights and solution field from all the cases for $Re = 100$ periodic cylinder flow

Convolution[1]	$\phi_{T=60}$	$\psi_{T=60}$	$\phi_{T=400}$	$\psi_{T=400}$
GP-k	4.9562E-3	1.2920E-1	1.1207E-2	1.2970E-1
GP-kv	5.4338E-3	9.9607E-2	2.1339E-2	9.9893E-2
GP-sv	7.7917E-3	4.7397E-2	2.2470E-2	5.1750E-2
GP-kv-POD[2]	3.5863E-4	9.9604E-2	1.5987E-3	9.9894E-2
POD[3]	2.7619E-2	6.2501E-4	1.2729E-1	1.4183E-3

Table 2.4: The L^2 error norms for the prediction of the weights and solution field from all the cases for $Re = 100$ periodic cylinder flow.

Convolution[4]	$\phi_{T=60}$	$\psi_{T=60}$	$\phi_{T=400}$	$\psi_{T=400}$
GP-k	1.6181E-3	7.0818E-2	4.2919E-3	7.0842E-2
GP-kv	3.4440E-3	5.1194E-2	8.3614E-3	5.1272E-2
GP-sv	2.5959E-3	2.7470E-2	6.6308E-3	2.7593E-2

[1]GP convolutions performed in this table use 300 centers.

[2]Retain 100 modes.

[3]Retain 10 modes, greater than 99.99 percent energy.

[4]GP convolutions performed in this table use 600 centers.

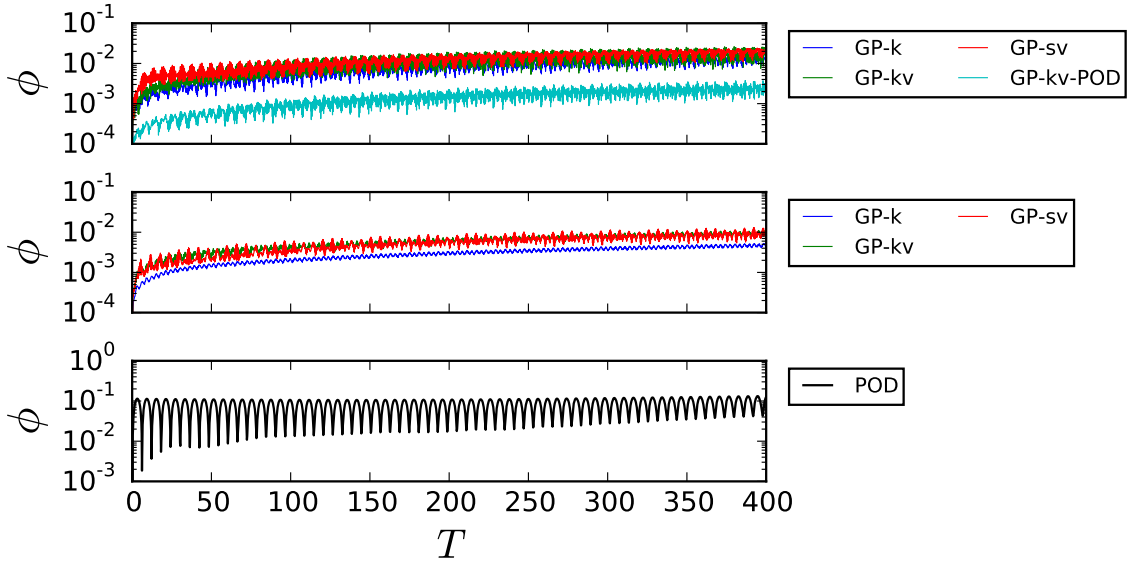


Figure 2.16: L^2 error norms of the weights prediction for the periodic cylinder flow at $Re = 100$. Top: GP convolutions with 300 centers. Middle: GP convolutions with 600 centers.

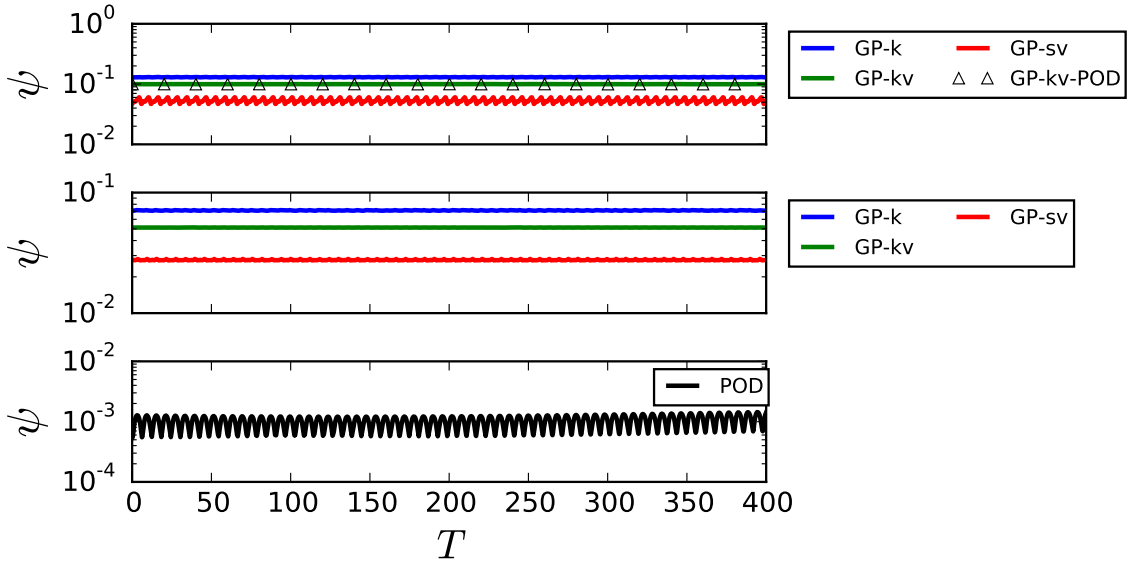
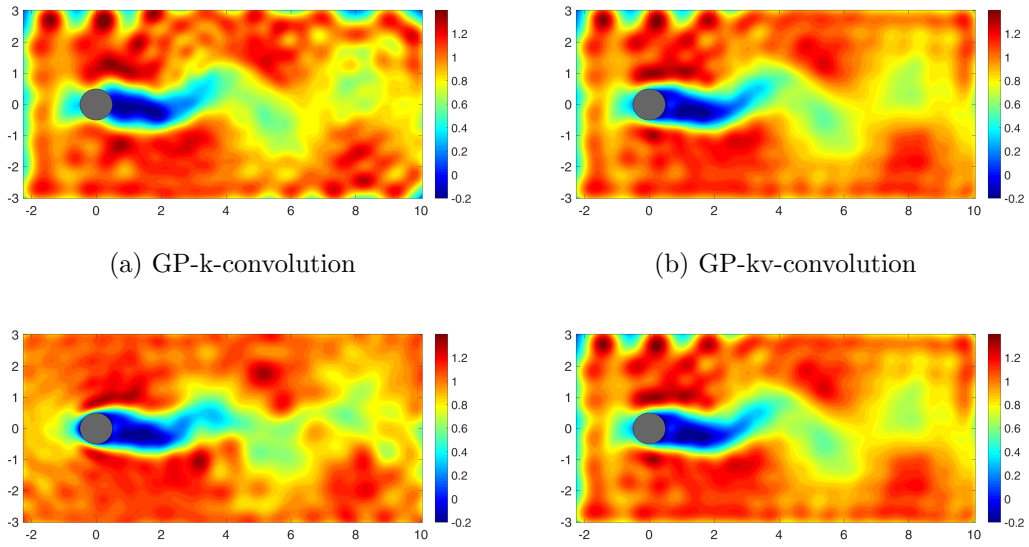


Figure 2.17: L^2 error norms of the solution field for the periodic cylinder flow at $Re = 100$. Top: GP convolutions with 300 centers. Middle: GP convolutions with 600 centers.



(a) GP-k-convolution

(b) GP-kv-convolution

(c) GP-sv-convolution

(d) GP-kv-POD-convolution

Figure 2.18: The stream-wise velocity contour of the predicted solution field at $T = 400$ for the periodic cylinder flow at $Re = 100$ from GP-convolutions with 300 centers.

For comparison purposes, the stream-wise velocity contour plots for the GP cases are shown in Figure 2.18. Not surprisingly, the aliasing effect occurs for all of the GP-convolution methods as the GP-convolution operator is not exactly invertible. On the other hand, the POD modes are orthogonal vectors and have an exact inverse which makes the deconvolution exact. Although the L^2 error norms for GP-sv-convolution is smaller as compared to GP-k-convolution and GP-kv-convolution, the wake of the cylinder is predicted poorly in Figure 2.18. However, both GP-kv and GP-kv-POD perform extremely well at the regions where the vortex shedding occurs but with small distortion at the edge of the domain. To support this argument, the L^2 error norms of the solution field shown in Figure 2.19 for a selected domain of $2 < x < 9$ and $-2 < y < 2$, is computed and denoted as ψ^S. It is observed that the performance of GP-kv-convolution and GP-k-convolution exceeds that of GP-sv-convolution. If the domain of interest is in the wake of cylinder regardless of the prediction near the edge of the domain, using GP-kv-convolution is preferable. While GP-kv-POD-

convolution produces fairly identical results as GP-kv-convolution, it is still worth noting that adding an extra layer of POD-convolution helps to reduce the rank of the system which reduces computational effort while solving the inverse problem for a smaller matrix.

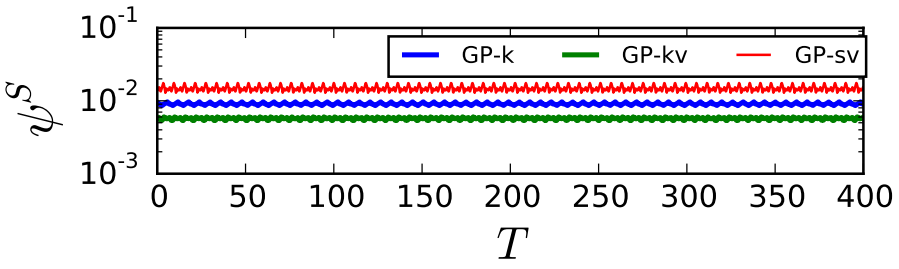
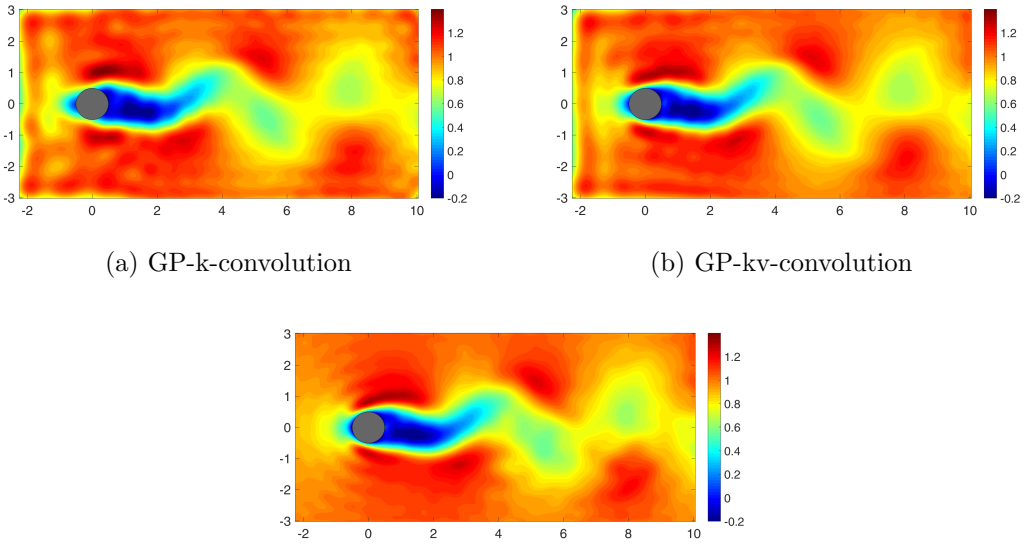


Figure 2.19: L^2 error norms of the solution field for the selected domain for periodic cylinder flow at $Re = 100$. 600 centers are used.



(a) GP-k-convolution

(b) GP-kv-convolution

(c) GP-sv-convolution

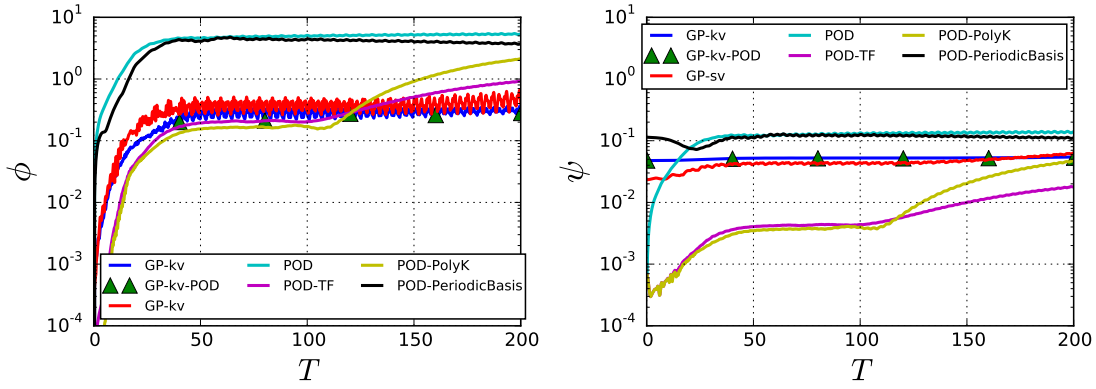
Figure 2.20: The stream-wise velocity contour of the predicted solution field at $T = 400$ for the periodic cylinder flow at $Re = 100$ from GP-convolutions with 600 centers.

As the number of centers used in GP-convolutions increase to 600, both the weights and field prediction improve notably as shown in Figure 2.16 and 2.17, and the cor-

responding stream-wise velocity fields are shown in Figure 2.20. In particular, the aliasing effect is less perceptible as compared to the cases using 300 centers. Comparing Figure 2.20 and 2.15, the predicted solution from GP-kv-convolution is nearly as good as POD-convolution with minor distortions at the left side of the domain, whereas GP-k-convolution and GP-sv-convolution have shown visible aliasing effect that covers the majority of the domain. Although GP-kv-convolution and GP-k-convolution share similar characteristics with respect to center placement, the former predicts less noisy and more accurate solution field than the latter. These findings show that the number and the location of the centers are the predominant factors that impact the performance of the GP-convolution-based model prediction. Ideally, one would like to place a center at every location where the flow information is available to capture the system dynamics, but the computational cost would be too high. Hence, the algorithm for learning centers to accurately and efficiently capture the flow physics is essential.

Transient Cylinder Flow

As has been pointed out already, the dynamics of a limit-cycle system is relatively straightforward to model. In this section, the goal is to model the more complex transient cylinder wake flow. As in the previous study, 340 ($T = 68$) snapshots from the cylinder data are used for training such that the first half of the data shows transient growth of the instability, while the second half shows periodic limit-cycle behavior. The analysis is performed using GP-kv-convolution, GP-sv-convolutions, GP-kv-POD-convolution, POD-convolution, GP-TF-convolution, and GP-PolyK to predict up to 1000 snapshots ($T = 200$). 600 center locations are chosen for the GP-convolutions keeping in mind the accuracy of the predictions from the prior study of periodic wake flow. The L^2 error norms for the predicted weights and the predicted field for each of the cases are shown in Figure 2.21a and Figure 2.21b, respectively.

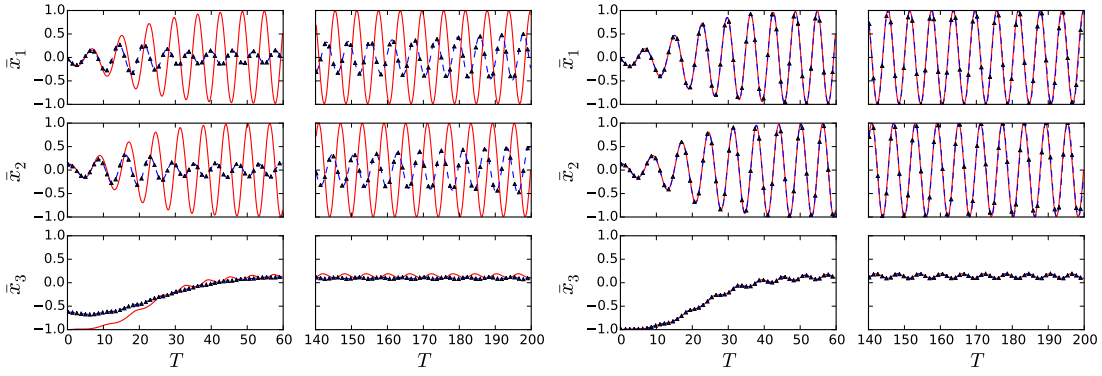


(a) (b)

Figure 2.21: L^2 error norms of the predicted weights(left) and solution field(right) for transient cylinder flow at $Re = 100$.

From Figure 2.21, it is observed that POD-convolution performs poorly in both weights and field prediction as compared to the other convolution methods. This result is not surprising and has previously been pointed out by Rowley & Dawson (2017). The transient to periodic bahavior of the cylinder flow is essentially nonlinear growth of the wake instability, and DMD(in our case, POD-convolution), being a linear method is not capable of predicting the correct dynamics. To illustrate this argument, the first three projected and predicted weights for POD-convolution, POD-TF-convolution, POD-PolyK-convolution, and GP-kv-convolution are compared in Figure 2.22. It is observed that the prediction from POD-convolution does not accurately capture the instability growth and the location of the limit cycle as observed in the evolution of the weights, whereas both POD-TF-convolution and GP-kv-convolution identifying both transient and periodic behavior with reasonable accuracy. The streamwise velocity contours at $T = 25$, $T = 68$, and $T = 200$ are shown in Figure 2.23, 2.24, and 2.25. Figure 2.26 shows the corresponding stream-wise velocity contour for the actual field. It is evident that the solution from POD-convolution does not predict the transitional behavior well. The approach of using POD-TF-

convolution performs well if prior knowledge of the dynamical system is available as in this case. In this case, it is well known that the accuracy of the prediction can be improved by using quadratic terms which is equivalent to constructing a non-linear model (Rowley & Dawson 2017). Thus, leveraging an user-defined transfer function is particularly effective for this flow problem.



(a) POD-convolution

(b) POD-TF-convolution

(c) POD-PolyK-convolution

(d) GP-kv-convolution

Figure 2.22: The first three projected(red) and predicted(blue) weights for the transient cylinder flow at $Re = 100$.

On the other hand, GP-kv-convolution displays comaprable accuracy to the POD-TF-convolution by capturing the transient evolution of the weights prediction as shown in Figure 2.21a. Even though the solution field still displays some aliasing effect from inaccurate deconvolution proceedure, this convolution operator does not rely on

prior knowledge of the system dynamics as compared to POD-TF-convolution. Similarly, layering a polynomial kernel function with POD-convolution, i.e., POD-PolyK-convolution, also performs reasonably well as evidenced in Figure 2.21a and 2.21b. This study highlights the effectiveness of the generalized sparse convolution framework for capturing the correct non-linear dynamics without prior knowledge of the dynamical system.

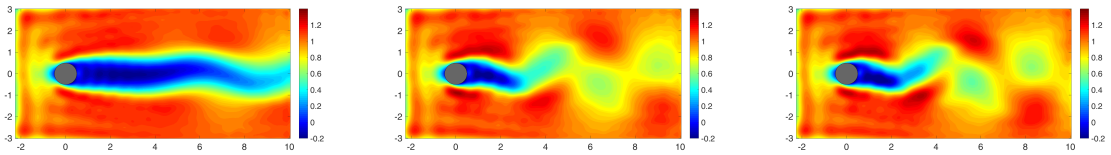


Figure 2.23: The predicted stream-wise velocity contour for the transient cylinder flow at $Re = 100$ using GP-kv-convolution. Left to right: $T = 25$, $T = 68$, and $T = 200$.

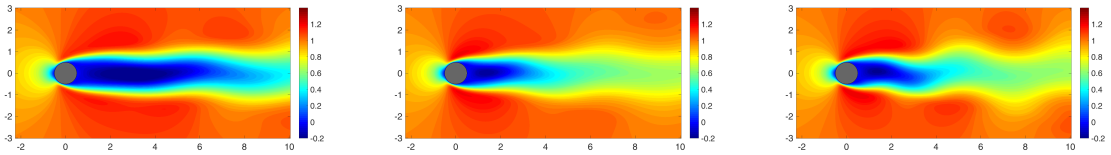


Figure 2.24: The predicted stream-wise velocity contour for the transient cylinder flow at $Re = 100$ using POD-convolution. Left to right: $T = 25$, $T = 68$, and $T = 200$.

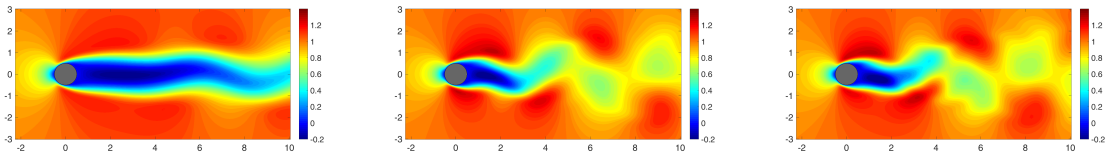


Figure 2.25: The predicted stream-wise velocity contour for the transient cylinder flow at $Re = 100$ using POD-TF-convolution. Left to right: $T = 25$, $T = 68$, and $T = 200$.

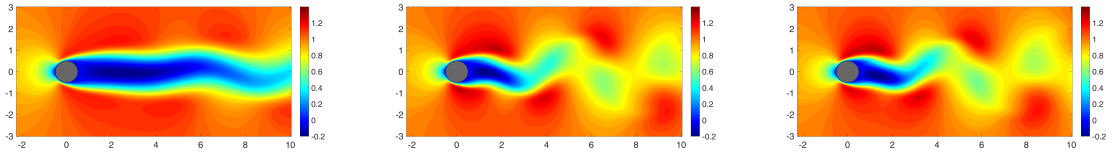


Figure 2.26: The actual stream-wise velocity contour for the transient cylinder flow at $Re = 100$. Left to right: $T = 25$, $T = 68$, and $T = 200$.

The predicted stream-wise velocity contour using GP-sv-convolution is shown in Figure 2.27. In comparison, both GP-kv-convolution and GP-sv-convolution perform reasonably well in predicting the transient and limit-cycle behavior. For the region of the cylinder wake, the predicted solution field is reasonably accurate and not as distorted as in the GP-kv-convolution method even though 600 center locations were used. This result indicates that center or sensor placement is indeed important for constructing an accurate and efficient predictive model.

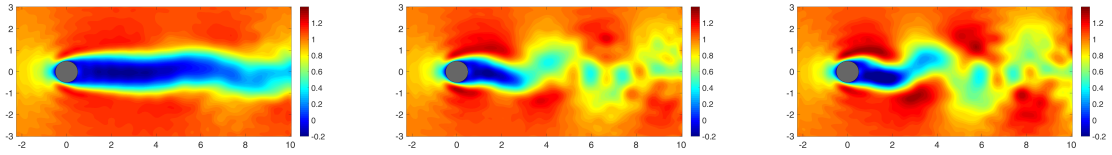


Figure 2.27: The predicted stream-wise velocity contour for the transient cylinder flow at $Re = 100$ using GP-sv-convolution. Left to right: $T = 25$, $T = 68$, and $T = 200$.

The L^2 error norms for both weights and field prediction are tabulated in Table 2.5 at $T = 68$ and $T = 200$. Instead of performing convolution on the transient cylinder data, this case sudy uses the POD modes from the periodic cylinder data for modeling the system. The corresponding case is termed as "POD-PeriBasis" as shown in Figure 2.28. It is observed that the behavior from transient to limit-cycle is not occurring within the training data, i.e., $T < 68$. This observation not only emphasizes that an appropriate basis or convolution operator is required to capture

the nonlinear system dynamics but also indicates that the periodic cylinder data does not contain the necessary growth mode for triggering the instability.

Table 2.5: The L^2 error norms for the prediction of the weights and solution field from all the cases for transient cylinder flow $Re = 100$.

Convolution[5] [6]	$\phi_{T=68}$	$\psi_{T=68}$	$\phi_{T=200}$	$\psi_{T=200}$
GP-kv	2.3826E-1	5.2452E-2	3.0967E-1	5.4405E-2
GP-kv-POD [7]	2.1352E-1	5.1747E-2	2.7648E-1	5.3024E-2
GP-sv	4.0995E-1	4.1658E-2	4.1864E-1	6.2778E-2
POD	4.7785E01	1.2272E-1	5.3785E01	1.3400E-1
POD-TF	2.0760E-1	4.2751E-3	9.1273E-1	1.7732E-2
POD-PolyK	1.6431E-1	3.6741E-3	2.1288E01	4.7402E-2
POD-PeriBasis	4.4731E01	1.2168E-2	3.7053E01	1.1148E-1

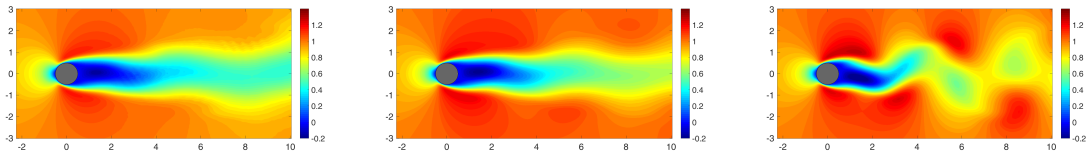


Figure 2.28: The predicted stream-wise velocity contour for the transient cylinder flow at $Re = 100$ using POD-PeriBasis-convolution. Left to right: $T = 25$, $T = 68$, and $T = 200$.

Boussinesq Buoyancy-Driven Mixing Flow

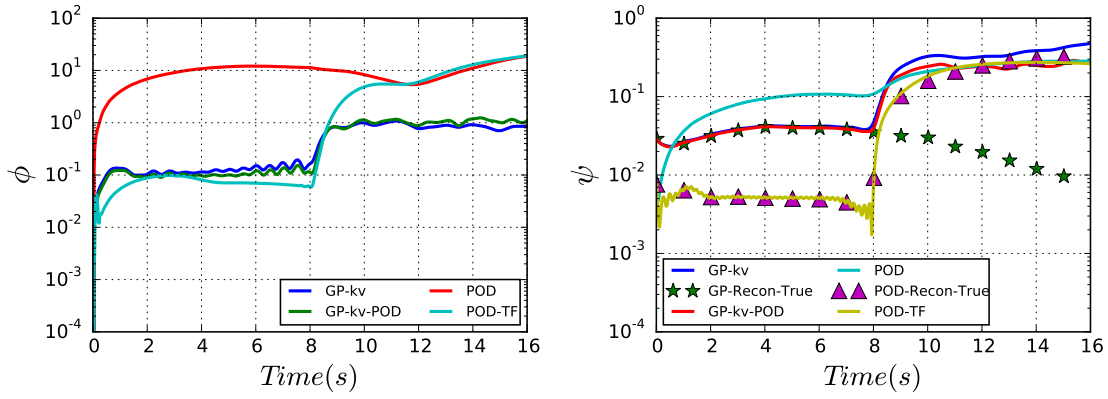
The third problem considered as part of our test bed is a Boussinesq buoyancy-driven mixing flow, which represents highly convective dynamics. Because Boussinesq flow does not settle into a limit-cycle (as against cylinder flow), predicting its evolution is

[5] GP convolutions performed in this table use 600 centers.

[6] POD convolutions performed in this table retain 50 modes.

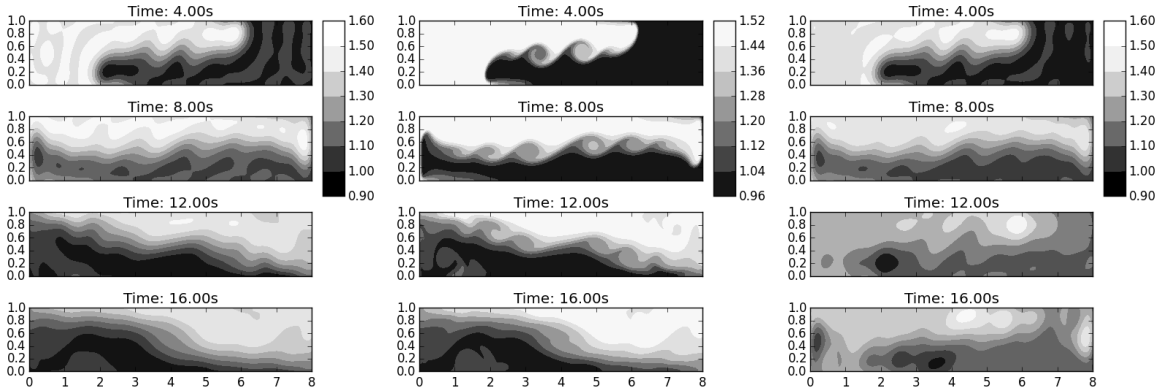
[7] 100 POD modes are retained.

rather challenging. In this analysis, different convolution operators such as GP-kv, GP-kv-POD, POD, and POD-TF-convolutions that performed the best for thge limit cycle case from the earlier study are explored. 400 snapshots($t = 8s$) of training data are used to predict up to 800 snapshots($t = 16s$). Since the optimal number of centers has not been explored, an ad hoc choice of 1500 centers is used. The L^2 error norms for the weights and field prediction are shown in Figure 2.29a and 2.29b, respectively. Similar to that observed for the cylinder wake flow, the POD-TF-convolution performs better than GP-convolutions and POD-convolution in the prediction of the weights. However, the L^2 error norms for both weights and field prediction increase substantially beyond the training region, i.e., $t > 8s$, for all the cases. To illustrate, the temperature fields are compared for GP-kv-convolution and POD-TF-convolution as shown in Fig 2.30 and Figure 2.31. This is because, the POD mdoes obtained from data in the training region and employed in the construction convolution operator is not optimal for the data to be predicted beyond the training set.



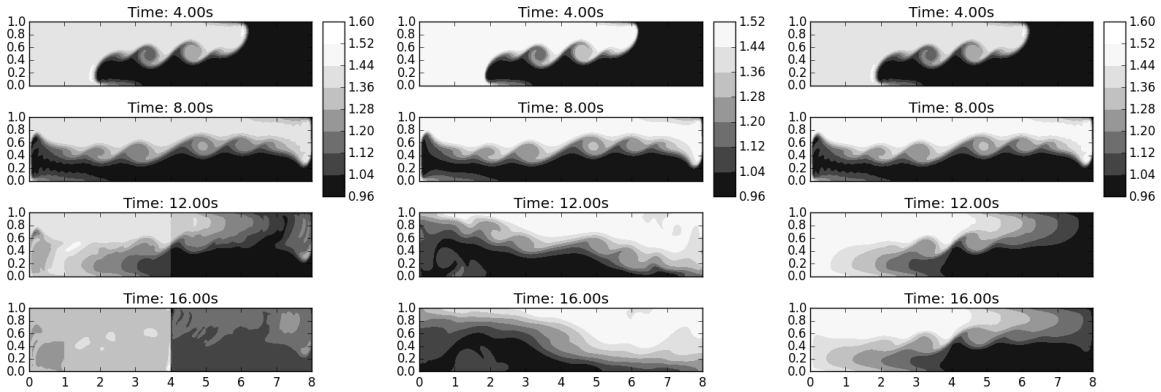
(a) (b)

Figure 2.29: L^2 error norms of the predicted weights(left) and solution field(right) for Boussinesq flow. Right: the predicted solution is compared with true solution.



(a) Reconstruted Solution (b) Actual Solution (c) Predicted Solution

Figure 2.30: Temperature field for Boussinesq flow using GP-kv-Convolution.

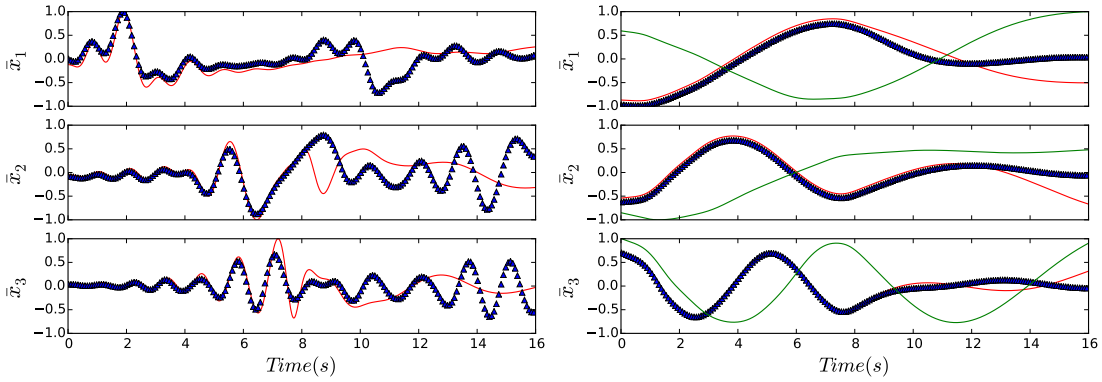


(a) Reconstruted Solution (b) Actual Solution (c) Predicted Solution

Figure 2.31: Temperature field for Boussinesq flow using POD-TF-Convolution.

Also shown are the reconstructed solution fields computed from the convolution operator and convolved weights. In principle, the best possible prediction with such a framework is the exact reconstruction from the convolution operator. For GP-kv-convolution, the field prediction performs well within the training region but deviates outside the training region. However, the reconstructed solution is close to the actual solution as expected. This implies that this convolution operator can potentially capture the dynamics. The predicted and projected weights are presented in Fig 2.32a. It is observed that the weights match well only within the training region. On the

other hand, the predicted and projected weights for POD-TF-convolution presented in Fig 2.32b show reasonable comparison even beyond the training region.



(a) (b)

Figure 2.32: The first three projected(red), true(green), and predicted(blue triangle) weights for Boussinesq flow using GP-kv(left) and POD-TF-convolution(right).

The issue with POD-TF-convolution is that the projected weights are not consistent with the true weights. Explaining further, the true weights for the POD-TF-convolution are computed by using the POD modes learned from all 800 snapshots of the data instead of 400 in Figure 2.32b. The reconstructed solution also appears to be a bad representation as compared to the actual solution in Figure 2.31. The reason is that the convolution operator(in this case, the POD modes) evolves dynamically as the system changes, and this complicates the modeling and prediction for such systems. However, it is worth mentioning that the dynamics within the training region are correctly predicted as evidenced in Figure 2.30 and 2.31. Table 2.6 and 2.7 summarize the L^2 error norms for both predicted weights and solutions at $t = 4s$ and $t = 8s$. For completeness, the predicted weights for POD and GP-kv-POD-convolution are shown in Figure 2.33a and 2.33b, respectively. The predicted solution field for POD and GP-kv-POD-convolution are shown in Figure 2.34 and 2.35, respectively.

In summary, the key to the sparse presented convolution framework capturing the

Table 2.6: The L^2 error norms for the prediction of the weights and solution field from different convolutions for Boussinesq flow at $t = 4s$.

Convolution	$\phi_{t=4s}$	$\psi_{t=4s}$
GP-kv	1.1440E-1	4.2218E-2
GP-kv-POD	9.7056E-2	4.1251E-2
POD	1.0828E01	9.4090E-2
POD-TF	8.0980E-2	5.2347E-3

Table 2.7: The L^2 error norms for the prediction of the weights and solution field from different convolutions for Boussinesq flow at $t = 8s$.

Convolution	$\phi_{t=8s}$	$\psi_{t=8s}$
GP-kv	1.6342E-1	4.6276E-2
GP-kv-POD	1.1282E-1	3.9037E-2
POD	1.1360E01	1.0789E-1
POD-TF	5.9375E-2	6.7019E-3

correct system dynamics is the identification of the appropriate convolution operator and sparse representation. To highlight this point, different POD-convolutions and GP-convolutions are investigated with various sparsification (center learning) algorithms(GP-k, GP-kv, and GP-sv) using different numbers (300 and 600) of centers. For the periodic cylinder flow, the favorable center placement(GP-kv with 600 centers) is identified for the most accurate model among all the cases using GP-convolution. Additionally, the aliasing effect observed in GP-convolutions can be related to the center placement algorithm. When the local temporal variability of the velocity field is exploited to determine the location of the centers, the predicted solution improves significantly and accurately captures the physics of the cylinder wake. Thus, the choice of the center placement is a critical component for accurate

model prediction in the GP-convolution.

When encountering more complex flows, e.g., the transient cylinder flow, the choice of the convolution operator is critical for predicting dynamical behavior. For example, the single-layer convolutions such as POD and POD-PeriBasis have failed to capture the transitional behavior. However, the idea of multi-layer convolution performs reasonably well as highlighted by the results from POD-TF, and POD-PolyK-convolution because the prior knowledge of this flow system has been utilized. As a result, the quadratic nonlinearity has been incorporated into these two convolutions for model prediction. On the other hand, GP-convolutions are also capable of predicting the transient dynamics correctly even without including transfer function or kernel function. The underlying reason has not been investigated yet, but this implies that GP convolution is advantageous than POD-convolution in case of unknown flow physics.

The computational time and L_2 error norms ψ of the predictive models for both periodic and transient cylinder flow are summarized in Table 2.8 as compared to the DNS data. For brevity, only the best cases are summarized below. For the periodic cylinder flow, the models are allowed to predict up to $T = 400$. The total time to compute the DNS data is approximately 204 minutes whereas both predictive models from POD and GP-kv-convolution require less than 2 minutes. This results in a ratio of more than 100 for the total time saved which is significantly remarkable. Similarly, the total time saved for the transient cylinder flow is comparable to the periodic cylinder flow, i.e., ratio of ~ 100. The models predicted in this case are allowed to evolve up to $T = 200$. It is worth noting that the amount of resources to collect high-fidelity DNS data is extremely demanding which the cylinder data in both cases are generated using 16 standard computer nodes in a parallel computing environment. On the other hand, the predictive models developed for the cylinder flows were performed on a regular desktop computer which saves a significant amount

of computing resources.

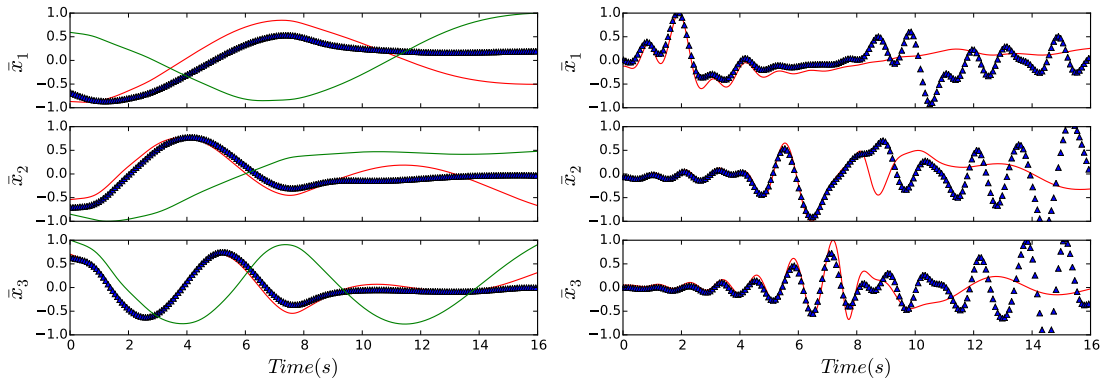
For both flow systems, POD-convolution(POD-TF-convolution) appears to outperform GP-kv-convolution regarding efficiency and accuracy due to three reasons as follows. Firstly, the inverse of the operator C from POD-convolution can be computed directly by taking the transpose which a significant amount of time can be saved. For GP-convolutions, the pseudo-inverse of C is computed to build the predictive models which require more computational effort. Secondly, determining the center locations is needed to build the convolution operator for GP-convolutions, and hence it is more computationally expensive than POD-convolution. Last but not least, operator C from GP-convolutions is not exactly invertible, and the resulting predictions will introduce the aliasing effect. This implies that GP-convolutions is inherently less accurate than POD-convolution even though an appropriate convolution is used. However, the predicted solutions from GP-kv-convolution appears to be more stable than POD-TF-convolution as evidenced in Figure 2.21b. The reason is still not fully understood at this time, and further investigation is needed in the future.

Table 2.8: The computational time(minutes) and error of the predictive models for both periodic and transient cylinder flows.

	Periodic cylinder flow		
	DNS	POD-convolution	GP-kv-convolution
Time (minutes)	204	0.3	1.5
L^2 error($\psi_{T=400}$)	-	1.42E-3	5.13E-2
	Transient cylinder flow		
	DNS	POD-TF-convolution	GP-kv-convolution
Time (minutes)	105	0.5	1.25
L^2 error($\psi_{T=200}$)	-	1.77E-2	5.44E-2

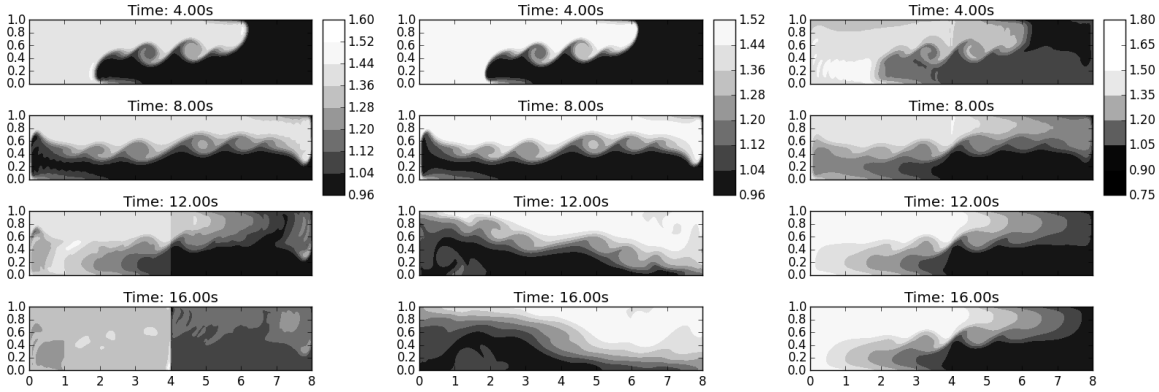
The dynamics of Boussinesq flow while challenging to predict from data, can be accurately captured within the training region using the correct convolution, e.g., GP-kv and POD-TF-convolutions. Beyond the training region, the predictions become erroneous. For POD type of convolution, the relevance of the POD basis to the data being predicted changes as the system evolves. In other words, the computed POD basis from 400 data snapshots is different from the ones computed using 800 data snapshots. For the system that evolves significantly in time, dynamic updates to POD basis may lead to accurate model. Hence, this becomes the primary motivation to explore the ideas of online updates in chapter 4. For GP-convolution, the various data-driven predictive models have failed to capture the evolving physics outside the training region although the convolution operator kernels themselves remain relevant. This indicates that the center locations may need to be modified as the physics evolves in time. The discussion of computational effort for this flow system is highlighted here because the flow physics has not been predicted correctly.

A disadvantage of such data-driven approaches is the need for ad-hoc regularization parameter for obtaining accurate and stable model predictions. This makes the modeling framework less self-consistent, but it is a common flaw in almost all data-driven approaches.



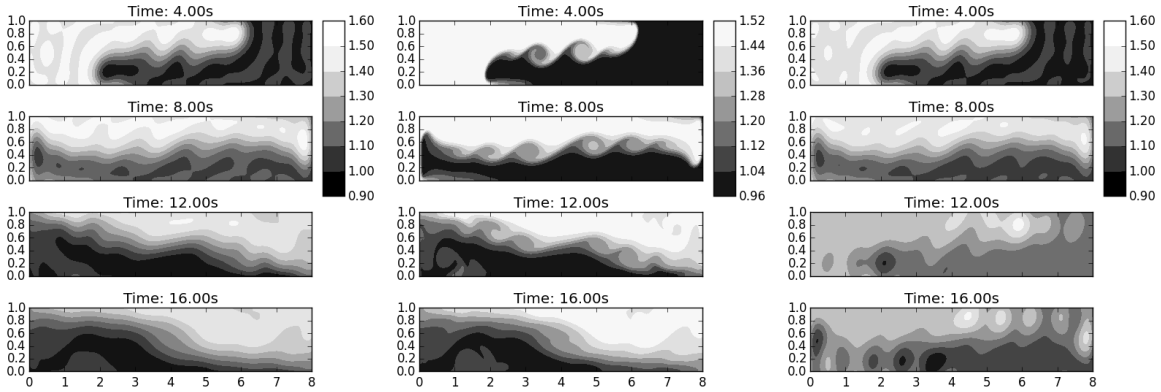
(a) (b)

Figure 2.33: The first three projected(red), true(green), and predicted(blue triangle) weights for Boussinesq flow using POD(left) and GP-kv-POD-convolution(right).



(a) Reconstruted Solution (b) Actual Solution (c) Predicted Solution

Figure 2.34: Temperature field for Boussinesq flow using POD-Convolution



(a) Reconstruted Solution (b) Actual Solution (c) Predicted Solution

Figure 2.35: Temperature field for Boussinesq flow using GP-kv-POD-Convolution

CHAPTER 3

SPARSE DATA RECONSTRUCTION FOR NON-LINEAR FLUID FLOWS

3.1 Motivation and Review

In many practical flow applications, sparse measurement data is often utilized to decipher the underlying system dynamics, especially when the underlying models are not known. An example is the case of data collected using unmanned aerial systems (UAS) over a spatial domain of interest. This data is invariably sparse, and UAS measurements are often leveraged to learn the structure of microscale geophysical flow phenomena such as pollutant/scalar transport, forest fires, and atmospheric turbulence. However, Billions of measurements are required to accurately measure atmospheric turbulence whereas hundreds of UAS is more realistic. In the laboratory, particle image velocimetry(PIV) (Adrian 2005, Kim et al. 2016) is one of the widely used methods to study complex flows. Although PIV can provide accurate quantitative flow measurement, the resolution of the data can be insufficient, or the data (image) quality can be unreliable in pockets for high fidelity analysis due to various experimental challenges including insufficient illumination, shadowing, obstructed view, and low seed density (Gunes & Rist 2008). As a consequence, the resulting data can be sparse, under-resolved, gappy and almost always noisy. Measuring the high dimensional state data for such flow systems is mostly impractical and too expensive. It is common that the resulting measurements are often limited in spatial resolution (Bui-Thanh et al. 2004). To correctly understand the flow system, the full state is often required to extract dominant mode structures such as POD and

DMD modes. Although recent advancements have focused on extracting dynamical modes from sparse measurement (Brunton et al. 2013, 2014), recovering the full high dimensional state from sparse data is still attractive (de Baar et al. 2014, Lee et al. 2015, Saini et al. 2016, Moreno et al. 2016). Consequently, sparse reconstruction is a critical component in data-driven approaches for analyzing non-linear fluid flow systems. Lastly, the aliasing effect resulting from the GP-convolutions from chapter 2 is associated with de-convolution not being exact, which again is a sparse reconstruction problem. The methods presented in this chapter will help to alleviate this issue.

Sparse reconstruction technique such as "gappy POD (GPOD)" based on L_2 minimization was originally proposed by Everson & Sirovich (1995) to recover marred faces in image processing. The fundamental idea of GPOD is to utilize the POD basis to approximate the coefficients that minimize the reconstruction error in L_2 sense. For sparse data with no knowledge of POD basis, GPOD can be extended to an iterative deconvolution procedure (iterative inverse filtering) that approximates both POD basis and coefficients. Naturally, this approach is less efficient and accurate compared to the non-iterative framework when the basis is known. Since its inception, GPOD has proven to be a beneficial method for data recovery as evidenced in many practical applications. For examples, Bui-Thanh et al. (2004) have utilized the knowledge of the computed aerodynamics or measured flow data to construct the entire flow field information, with successful extensions to inverse airfoil design from data. Similarly, Venturi & Karniadakis (2004) reconstructed accurate representations of velocity field for cylinder wake from data with different sparsity levels. Saini et al. (2016) have employed GPOD to recover missing data in PIV measurements for the studies of gas turbine combustor flows. The success of the GPOD lies in the POD basis being a sparse representation of the data.

In recent years, compressive sensing (Candès et al. 2006, Tropp & Gilbert 2007, Candès & Wakin 2008, Needell & Tropp 2009) has been popular as sparse reconstruc-

tion framework based on L_1-minimization. This approach has also been leveraged for fluid flow applications (Brunton et al. 2013, Bright et al. 2013, Bai et al. 2014, Brunton et al. 2014, Bai et al. 2017, Kramer et al. 2017). The goal of compressive sensing is to use a limited amount of data to identify a sparse set of coefficients using L_1 minimization. For a sparse system, one can throw away the majority of the coefficients and retain only a few to represent the system. Compressive sensing identifies the combination of sparse coefficients that work best. In contrast, GPOD tries to approximate with equal fidelity, all the coefficients used to model the system. Both these approaches have their inherent advantages and disadvantages, and their success is very problem dependent. Many studies have focused on integrating ideas of compressive sensing into fluid flow modeling. Using this L_1 minimization framework, there exist two strands of effort - compressive sampling and compressive sensing. The former refers to the existence of a higher dimensional data set from which a sparse-sampled data-set is generated for efficient computation. This is not that dissimilar to the sparse convolution ideas in the earlier chapter. Once the sparse data is generated, a sparse reconstruction method is employed to map back to the high-resolution flow field. In contrast, the compressive sensing approach involves purely sparse measurements (without knowledge of the high dimensional state vector) which are then used to reconstruct the full flow field. Bright et al. (2013) have utilized compressive sampling to characterize the flow around a cylinder with limited pressure measurements, and Bai et al. (2017) have exploited sparse sampling to provide efficient measurement and characterization of a fluid system which reduces the effort to obtain valuable data. Bai et al. (2014) have compressed and reconstructed turbulent airfoil data using strategies from compressive sampling. Kramer et al. (2017) proposed a sparse sensing framework based on DMD algorithm to identify flow regimes, ranging from steady to chaotic flow, and bifurcations in large-scale thermo-fluid systems.

Local kriging has also been an effective method to interpolate to the unknown

values using the known data, especially in the field of geology (Oliver & Webster 1990, Deutsch & Journel 1998, Marinoni 2003). Kriging has strong connections to GP regression. Over the past few decades, the use of kriging to recover sparse data has gradually emerged for modeling fluid flow systems as evidenced in Venturi & Karniadakis (2004), Gunes et al. (2006) and Gunes & Rist (2008). In particular, Gunes et al. (2006) have utilized kriging to reconstruct flow past a cylinder and suggested that kriging performs better with data that has high gappiness or low temporal resolution as compared to the results from L_2-based reconstruction technique. For a sensitive region, i.e., the region with absolute instability, Gunes et al. (2006) have shown that kriging-based interpolation is an effective method to repair the missing data. Also, Gunes & Rist (2008) have used kriging to eliminate or alleviate the incorrect and missing data in PIV measurements of a separated transitional flat-plate boundary layer. de Baar et al. (2014) have provided an improved reconstruction of the experimental velocity data obtained from PIV using kriging based regression as compared to the traditional methods.

In summary, L_2 minimization, L_1 minimization and kriging-based methods have all shown promising results for practical fluid flow applications. However, their effectiveness depends on the physics to be modeled. In this study, their performance will be compared for different classes of fluid flows.

3.2 Objective and Contribution

Insufficient data is often encountered in many practical applications, and sparse reconstruction methods help recover the full-field solution. The major goal of this study is to explore reconstruction techniques which can be potentially integrated into the convolution framework developed in chapter 2 for modeling non-linear fluid flow systems using sparse data. Additionally, the tools developed can be leveraged to analyze the physics contained in sparse measurement data for many practical flows. The primary

contributions of this work include the exploration, implementation, and evaluation of three techniques that were mentioned previously for sparse data reconstruction. The first method is L_2-based reconstruction with POD basis(GPOD). Secondly, a L_1-based reconstruction, which uses compressive sampling matching pursuit(CoSaMP), is exploited as an alternative method for data recovery. Thirdly, local kriging, a statistical method, is utilized to interpolate the unknown values from the known data. The detailed procedure for these methods are discussed in section 3.4 and section 3.5. The focus of this study is to evaluate the performance of the techniques mentioned above for three different classes of reconstructions, namely as:

1. Sparse reconstruction with known basis, i.e., the POD basis are learned from the full field data.

2. Sparse reconstruction with known but inexact basis, i.e., the POD basis are learned from the related flow data at different temporal regime or the low-resolution data.

3. Sparse reconstruction with unknown basis, i.e., the basis functions are learned from the sparse data.

The questions to be answered in this study are summarized as follows.

1. What is the minimum amount of sparse data needed to recover an accurate solution?

2. What is the optimal rank for sparse recovery of a particular system?

3. Which of the three methods performs the best?

4. For unknown basis, how many modes can be accurately captured?

To effectively answer these questions, the reconstruction techniques are applied to three fluid flow systems including (a) flow past a cylinder at $Re = 100$, which is a

low-dimensional system, (b) 2-D turbulent channel flow ($Re = 4200$ and $Re_\tau = 180$), which is a high-dimensional system, and (c) 2-D reduced channel flow ($Re = 4200$ and $Re_\tau = 180$), which zeroes in on the near-wall flow and is a moderate-dimensional system. A low-dimensional system is one which can be described effectively with a small number of basis (be it POD or other basis functions) whereas a high-dimensional system requires a large number of modes for accurate representation as the higher modes can have non-negligible energy content and important low-energy structures.

3.3 General Theory

The L_2 and L_1-based methods introduced in this study have conceptual similarities to inverse filtering ideas such as approximate deconvolution(AD) which require repeated application of the filter to obtain its inverse. However, in AD the exact form of the filter is not known and is usually guessed. As a result, the quality of the reconstruction depends on the choice of filter kernel employed. In the sparse reconstruction methods considered in this chapter, the filter kernel is mostly known from data in the form of POD basis. The primary focus here is the L_2 and L_1-based minimization framework which relies on manifold learning (POD bases) from data instead of adopting a functional filter kernel for sparse reconstruction.

The general theory of sparse reconstruction from data can be outlined as follows. Consider a set of sparse data $\tilde{X} \in \mathbb{R}^{P\times M}$, it can be described by a filter or a measurement matrix $C \in \mathbb{R}^{P\times N}$ as:

$$\tilde{X} = CX, \tag{3.1}$$

where $X \in \mathbb{R}^{N\times M}$ is the full field representation that is needed to be recovered. P, N, and M are the number of measurements, full field grid dimension, and snapshots of data respectively. The left pseudo-inverse of C can be applied to Equation 3.1 to

recover the reconstructed field as:

$$C^{+}\tilde{X} = X. \tag{3.2}$$

The challenge in Equation 3.2 is to find the pseudo-inverse of the measurement matrix C that is usually ill-conditioned. This results in solving an ill-posed and underdetermined problem for $P << N$. Consequently, C^{+} is non-unique and requires regularization which is mostly *ad hoc*. To improve the conditioning of C, the full field data can be represented using a low-dimensional basis $\Phi \in \mathbb{R}^{N\times K}$ and coefficient $a \in \mathbb{R}^{K\times M}$ as:

$$X = \Phi a. \tag{3.3}$$

Equation 3.1 can also be rewritten as:

$$\tilde{X} = C\Phi a. \tag{3.4}$$

Then the sparse reconstruction problem becomes:

$$a = (C\Phi)^{+}\tilde{X}. \tag{3.5}$$

In comparison to matrix C, the conditioning of $(C\Phi)$ is greatly improved when K is in order of P, i.e., $(C\Phi)$ is closer to a square matrix. Therefore, solving for the coefficient a is more effective and robust. Equation 3.4 is a generalized framework that is applicable to both L_2-based and L_1-based techniques(compressive sampling). In this study, POD basis represent the only sparse basis used in both approaches. Other possible choices of basis include Fourier functions or wavelets. In particular, the L_2-based method with POD basis is also known as GPOD, and was developed by Everson & Sirovich (1995) to solve a minimization problem in Equation 3.6. The details of the GPOD procedure are discussed in section 3.4.1.

$$\min \left|\left|\tilde{X} - C\Phi a\right|\right|_2^2. \tag{3.6}$$

Alternatively, in the compressive sensing frameworks, the goal is to solve the sparse coefficient a with L_1 minimization as shown in Equation 3.7.

$$\min \left|\left| \tilde{X} - C\Phi a \right|\right|_2^2 \quad \& \quad \min ||a||_1 . \tag{3.7}$$

In this study, the CoSaMP algorithm (Needell & Tropp 2009) is employed to solve the above minimization problem which requires a combinatorial search. This method is an improved version of the orthogonal matching pursuit algorithm(OMP) (Tropp & Gilbert 2007) with several modifications that accelerate the convergence and provide numerical stability (Needell & Tropp 2009). The details of the CoSaMP algorithm are discussed in section 3.4.2.

3.4 Sparse Reconstruction with Known Basis

3.4.1 L_2-based Minimization

In this subsection, the mathematical steps of the original L_2 minimization algorithm using POD basis or GPOD procedure (Everson & Sirovich 1995) are introduced for reconstructing sparse data. The reader is assumed to possess knowledge of POD and SVD as well as their computations. Given a complete data vector, one can obtain the POD basis $\phi_k(z)$ and coefficient a_k as:

$$x(z) = \sum_{k=1}^{K} a_k \phi_k(z). \tag{3.8}$$

The masking of data refers to blanking out elements at the chosen spatial locations. As a result, the masked data $\tilde{X}(z)$ and mask vector $m(z)$ are related as:

$$\tilde{x}(z) = m(z)x(z). \tag{3.9}$$

The elements of the mask vector are either 1 or 0 depends on the availability of the data. Relating with the general theory from previous section, the mask vector is equivalent to the measurement matrix C, but of a different dimension. In fact, one

can compute the full measurement matrix C for the given mask vector in GPOD if memory storage permits. For practical implementation, point-wise computation using the mask vector is much more computationally efficient than computing matrix C. The goal of the GPOD procedure is to recover the masked (inaccesible) data by approximating the POD coefficient $\bar{a}$ from the unmasked data:

$$\tilde{x}(z) \approx m(z) \sum_{k=1}^{K} \bar{a}_k \phi_k(z). \tag{3.10}$$

Unlike the traditional POD framework, the coefficient vector $\bar{a}$ cannot be computed directly from the inner product, e.g., from equation 3.8, $a_k = \langle u, \phi_k \rangle$. To obtain the "best" approximation of the coefficient $\bar{a}$, one can minimize the error in an L_2 sense:

$$\text{error} = \left|\left| \tilde{x}(z) - m(z) \sum_{k=1}^{K} \bar{a}_k \phi_k(z) \right|\right|_2^2, \tag{3.11}$$

By denoting the gappy basis function as $\tilde{\phi}(z) = m(z)\phi(z)$, Equation 3.11 can be re-written as:

$$\text{error} = \left|\left| \tilde{x}(z) - \sum_{k=1}^{K} \bar{a}_k \tilde{\phi}_k(z) \right|\right|_2^2. \tag{3.12}$$

To minimize the error, Equation 3.12 can be rearranged using the property of Forbenius norm with matrix notation in Equation 3.13.

$$\text{error} = tr[(\tilde{x} - \tilde{\Phi}\bar{a})^T (\tilde{x} - \tilde{\Phi}\bar{a})]. \tag{3.13}$$

By expanding Equation 3.13 and applying the basic property of trace $tr(a + b) = tr(a) + tr(b)$:

$$\text{error} = tr(\tilde{x}^T\tilde{x}) - tr(\bar{a}^T\tilde{\Phi}^T\tilde{x}) - tr(\tilde{x}^T\tilde{\Phi}\bar{a}) + tr(\bar{a}^T\tilde{\Phi}^T\tilde{\Phi}\bar{a}). \tag{3.14}$$

By applying additional trace property for a product $tr(ab) = tr(ba)$:

$$\text{error} = tr(\tilde{x}^T\tilde{x}) - tr(\tilde{x}\bar{a}^T\tilde{\Phi}^T) - tr(\tilde{\Phi}\bar{a}\tilde{x}^T) + tr(\tilde{\Phi}^T\tilde{\Phi}\bar{a}\bar{a}^T). \tag{3.15}$$

Differentiate Equation 3.15 respect to $\bar{a}$:

$$\frac{\partial}{\partial \bar{a}}[\text{error}] = \frac{\partial[tr(\tilde{x}^T\tilde{x}) - tr(\tilde{x}\bar{a}^T\tilde{\Phi}^T) - tr(\tilde{\Phi}\bar{a}\tilde{x}^T) + tr(\tilde{\Phi}^T\tilde{\Phi}\bar{a}\bar{a}^T)]}{\partial \bar{a}}. \tag{3.16}$$

Using matrix calculus (Traa n.d.), each term in Equation 3.16 can be expressed as:

$$\frac{\partial[tr(\tilde{x}\bar{a}^T\tilde{\Phi}^T)]}{\partial\bar{a}} = \tilde{\Phi}^T\tilde{x},\ \frac{\partial[tr(\tilde{\Phi}\bar{a}\tilde{x}^T)]}{\partial\bar{a}} = \tilde{\Phi}^T\tilde{x},\ \text{and}\ \frac{\partial[tr(\tilde{\Phi}^T\tilde{\Phi}\bar{a}^T\bar{a})]}{\partial\bar{a}} = 2\tilde{\Phi}^T\tilde{\Phi}\bar{a}. \quad (3.17)$$

Consequently, Equation 3.16 is simplified by substituting Equation 3.17:

$$\frac{\partial}{\partial\bar{a}}(\text{error}) = -2\tilde{\Phi}^T\tilde{x} + 2\tilde{\Phi}^T\tilde{\Phi}\bar{a}. \quad (3.18)$$

By setting Equation 3.18 to zero, a linear system of equations are obtained as:

$$\mathbf{M}\bar{a} = f, \quad (3.19)$$

where $\mathbf{M}_{i,j} = \langle\tilde{\phi}_i, \tilde{\phi}_j\rangle$ and $f_i = \langle\tilde{x}, \tilde{\phi}_i\rangle$. Then the full field data isreconstructed as:

$$\bar{x}(z) = \sum_{k=1}^{K} \bar{a}_k\phi_k(z). \quad (3.20)$$

It is important to note that the final form of the repaired data requires one to substitute the original known data. Algorithm 3 summarizes the procedure of the GPOD with the knowledge of full data ensemble, i.e, the access to the POD basis function (ϕ_k) is available.

Algorithm 3: L_2-minimization with known basis

input : Full data ensemble $X \in \mathbb{R}^{N\times M}$

Incomplete data vector $\tilde{x} \in \mathbb{R}^N$

the mask vector $m \in \mathbb{R}^N$.

output: Approximated full data vector $\bar{x} \in \mathbb{R}^N$

1 Option: take out the temporal mean of the ensemble X.

2 Compute SVD of X to obtain the POD basis function Φ.

3 Decide on number of modes to retain.

4 Build a least square problem: $\mathbf{M}\bar{a} = f$.

5 Compute gappy basis function: $\tilde{\Phi} = m\Phi$ with point-wise multiplication.

6 Compute matrix $\mathbf{M} = \tilde{\Phi}^T \cdot \tilde{\Phi}$.

7 Compute vector: $f = \tilde{\Phi}^T \cdot \tilde{x}$.

8 Solve $\bar{a}$ from the least squares problem: $\mathbf{M}\bar{a} = f$.

9 Reconstruct the approximated solution $\bar{x} = \Phi\bar{a}$.

10 Substitute the gappy data back to $\bar{u}$:

(a) $\bar{x}_i = \bar{x}_i \quad \text{if} \quad m_i = 0$

(b) $\bar{x}_i = \tilde{x}_i \quad \text{if} \quad m_i = 1$

11 Output the approximated full data vector $\bar{x}$.

3.4.2 L_1-based Minimization

In compressive sensing methods that leverage L_1 minimization ideas, the approach is to employ a combinatorial matrix $\Theta \in \mathbb{R}^{P\times K}$, which is mathematically equivalent to $C\Phi \in \mathbb{R}^{P\times K}$ in Equation 3.4.

$$\tilde{X} = \Theta a. \tag{3.21}$$

Different from the GPOD procedure introduced in previous section, the compressive sampling strategy is to solve for coefficient a subject to a specified sparsity level that

best fits Equation 3.21 in L_1 sense as shown in Equation 3.22.

$$\min \left|\left| \tilde{X} - C\Phi a \right|\right|_2^2 \quad \& \quad \min ||a||_1 . \tag{3.22}$$

There are two restrictions for this method. First, the data should be adequately represented in a low-dimensional space, i.e., most of the elements in a are zero. Secondly, to form an appropriate matrix Θ, the measurement matrix C and the basis function Φ must be incoherent to each other. In other words, $\Theta(C\Phi)$ needs to satisfy the restricted isometry principle (RIP) for sparse coefficient a (Brunton et al. 2013):

$$(1-\delta)\,||a||_2^2 \leq ||C\Phi a||_2^2 \leq (1+\delta)\,||a||_2^2 , \tag{3.23}$$

where δ is a restricted isometry constant, and s is the sparsity level of coefficient a. Sparsity level s indicates the number of the elements in coefficient a are non-zero.

Solving Equation 3.22 requires a solution of a combinatorial problem that is a non-deterministic polynomial time problem. Greedy algorithms such as OMP (Tropp & Gilbert 2007) and CoSaMP (Needell & Tropp 2009) have advantages of requiring less computational time and samples as compared to the convex relaxation (Daubechies et al. 2004) and combinatorial algorithms(Gilbert et al. 2002, 2007, Needell & Tropp 2009). In this study, the CoSaMP algorithm from Needell & Tropp (2009) is chosen. CoSaMP is a greedy iterative procedure, which is a modified version of the OMP method to identify the columns of Θ that are strongly correlated with the sparse measurements $\tilde{X}$. With the combined matrix Θ, sparse data $\tilde{X}$, and target sparsity level s, each iteration of CoSaMP consists of the following five steps (Needell & Tropp 2009):

1. Identification: compute the sample proxy, which is a vector that is highly correlated with the current sample, and locate the largest components of the proxy.

2. Support Merger: unite the set of identified components with the set of components in the current approximation.

3. Estimation: solve a least squared problem to approximate the targeted sparse coefficient.

4. Pruning: produce a new approximation by retaining only the largest components of the newly solved sparse coefficients.

5. Sample update: update the samples by computing the residual between the current samples and approximated samples.

The detailed algorithm for sparse reconstruction with the known basis is summarized in Algorithm 4 and 5. To recover the sparse coefficient $\bar{a}$, a sparsity level s is required as one of the inputs for computation. According to Needell & Tropp (2009), a simple strategy is to deduce s from the number of measurements K: $s \approx P/(2\log K)$, which is reasonable in particular for high dimension data, i.e., $K >> P$. It is important to note that the transformed basis and coefficient in low-dimensional space are utilized. Therefore, the size of the coefficient a is equivalent to the number of measurements in this context, and it is different than how the measurements are normally viewed as spatial points or degree of freedoms. The second approach is rather brutal but could be easy in some situations: running CoSaMP with a range of sparsity levels and selecting the best approximation obtained, say $s = 1, 2, 4, 8, ..., P$.

A most straightforward way to stop the iterative process of the CoSaMP algorithm is to specify a maximum number of iterations. Alternatively, the normalized residual $\frac{||v||_2}{\left|\left|\tilde{X}\right|\right|_2}$ (Needell & Tropp 2009) can be computed to see whether it is smaller than the tolerance ϵ, which is pre-specified before the algorithm. In this study, the POD basis functions are chosen for the L_1 method. Fourier or wavelet basis can also be used to construct sparse samplings (Candès et al. 2006, Candès & Wakin 2008)

Algorithm 4: L_1-minimization with known basis part I

input : Full data ensemble $X \in \mathbb{R}^{N\times M}$

Incomplete data vector $\tilde{x} \in \mathbb{R}^N$.

Mast vector $m \in \mathbb{R}^N$.

Sparsity level: s.

halting tolerance: ϵ.

output: Approximated full data ensemble $h \in \mathbb{R}^{N\times M}$.

Compute the POD basis Φ from X.

Compute matrix Θ from mask matrix m and Φ.

Set iterative variable $k = 0$ and coefficient $\bar{a}^k = 0$.

Compute the current sample, $v = \tilde{X}$, where $v \in \mathbb{R}^{P\times 1}$.

while *Halting is false* **do**

  Form the sample proxy: $y = \Theta^* v$, where * is complex conjugate.

  Identify the large components from the proxy: $\Omega = \text{supp}(y_{2s})$.

  Merge supports: $T = \Omega \cup \text{supp}(\bar{a}^{k-1})$.

  Solve a least squares problem: $b|_T = \Theta_T^+ \tilde{X}$.

  Find the first s largest and nonzero elements: $\bar{a}^k = b_s$.

  Update the samples: $v = \tilde{X} - \Theta\bar{a}^k$ and check halting criterion.

end

Algorithm 5: L_1-minimization for known basis part II

Compute the approximated solution: $\bar{h} = \Phi\bar{a}$.

for *i=1* **to** N **do**

  $h_{i,j} = \bar{h}_i$ if $m_{i,j} = 0$ or $h_{i,j} = \tilde{x}_i$ if $m_{i,j} = 1$

end

3.5 Sparse Reconstruction with Unknown Basis

3.5.1 L_2-based Minimization

In the case of unknown basis function, both Φ and a are solved using an iterative framework (Everson & Sirovich 1995). The lack of knowledge of the basis puts much more strains on the quality of the sparse data to identify the correct POD modes. The complete high-dimensional data is needed to compute the initial guess for the POD basis which requires one to fill in the missing elements with some regression procedure. The incomplete data ensemble $\tilde{X} \in \mathbb{R}^{N\times M}$ and the corresponding mask vector ensemble $n \in \mathbb{R}^{N\times M}$ can be expressed as:

$$n(z,t) = \begin{cases} 0, & \text{if } \tilde{X}(z,t) \text{ is missing or incorrect.} \\ 1, & \text{if } \tilde{X}(z,t) \text{ is known.} \end{cases} \tag{3.24}$$

A possible approach to fill in the missing elements is to use the temporal mean of the sparse data. For instance, the initial complete data for starting iteration is denoted as h^0:

$$h^0(z,t) = \begin{cases} \tilde{X}(z,t), & \text{if } n(z,t) = 1. \\ x_{avg}(z), & \text{if } n(z,t) = 0. \end{cases} \tag{3.25}$$

$$\bar{x}_{avg,i} = \frac{1}{Q_i}\sum_{j=1}^{M} \tilde{X}_{i,j}, \qquad Q_i = \sum_{j=1}^{M} n_{i,j}. \tag{3.26}$$

However, this method breaks down if the data information is missing at all time as pointed out by Gunes et al. (2006). Another suggestion for the initial guess is to use Kriging as discussed in section 3.5.4. Once the initial guess of the full data is constructed, the iterative procedure can be started by computing the POD basis:

$$h^0(z,t) = \sum_{k=1}^{K} a_k(t)\phi_k(z). \tag{3.27}$$

The gappy basis function is denoted as $\tilde{\phi}_k(z) = n(z,t)\phi_k(z)$ for each data snapshot at time t. Similar to before, a least-squares problem is solved to approximate the

coefficient $\bar{a}$ by minimizing Equation 3.28:

$$\text{error} = \left|\left| \tilde{X}(z,t) - \sum_{k=1}^{K} \bar{a}_k(t)\tilde{\phi}_k(z) \right|\right|_2^2. \tag{3.28}$$

Subsequently, it leads to the linear system of equations:

$$\mathbf{M}\bar{a} = f, \tag{3.29}$$

where $\mathbf{M}_{i,j} = \langle \tilde{\phi}_i, \tilde{\phi}_j \rangle$ and $f_i = \langle \tilde{U}, \tilde{\phi}_i \rangle$. Important to note, if M numbers of sparse data snapshots are used, Equation 3.29 is to be solved for M times in each iteration, one for each data snapshot. The repaired solution at the first iteration for each snapshot can be computed as:

$$h^1(z,t) = \sum_{k=1}^{K} \bar{a}_k(t)\phi_k(z). \tag{3.30}$$

Furthermore, $h^1(x,t)$ can be used for improving the solutions in the next iteration by repeating Equation 3.27-3.30. The iterative process should be repeated until the maximum number of iterations is reached or until the algorithm has converged for the eigenvalues and eigenvectors. The iterative procedures for repairing incomplete data are summarized in Algorithm 6 and 7.

Algorithm 6: L_2-minimization with unknown basis part I

input : Incomplete data ensemble $\tilde{X} \in \mathbb{R}^{N\times M}$.

Mask matrix $m \in \mathbb{R}^{N\times M}$.

Maximum number of iteration: $iter$.

output: Approximated full data ensemble $h \in \mathbb{R}^{N\times M}$.

1 Initialize the iterative variable: $l = 0$.

2 Compute the initial iterative guess h by filling the missing elements with average values

(a)Sum the mask matrix for each dimension i: $Q_i = \sum_{j=1}^{M} m_{i,j}$.

(b)Compute the temporal mean for the incomplete data:

$x_{avg,i} = \{\sum_{j=1}^{M} \tilde{X}_{i,j}\}/Q_i$.

(c)Compute the initial guess: $h^l_{i,j} = \tilde{X}_{i,j}$ if $m_{i,j} = 1$ or $h^l_{i,j} = x_{avg,i}$ if $m_{i,j} = 0$.

Algorithm 7: L_2-minimization with unknown basis part II

while $l < iter$ **do**
  Compute the POD basis Φ from h^l
  Decide K numbers of modes to retain.
  $l = l + 1$
  for *j=1* **to** M **do**
    for *k=1* **to** K **do**
      for *i=1* **to** N **do**
        Compute gappy basis function: $\tilde{\Phi}_{i,k} = m_{i,j}\Phi_{i,k}$.
      end
    end
    Compute matrix : $\mathbf{M} = \tilde{\Phi}^T\tilde{\Phi}$.
    Compute vector: $f = (\tilde{\Phi})^T h^{l-1}_{:,j}$.
    Solve $\bar{a}$ from a least squares problem: $\mathbf{M}\bar{a} = f$.
    Compute the approximated solution: $\bar{h} = \Phi\bar{a}$.
    Update the repaired data for only missing elements:
    for *i=1* **to** N **do**
      $h^l_{i,j} = \bar{h}_i$ if $m_{i,j} = 0$ or $h^l_{i,j} = h^{l-1}_i$ if $m_{i,j} = 1$
    end
  end
end

3.5.2 L_1-based Minimization

The iterative procedure from the L_2-based approach can be adopted for the L_1-minimization framework. The algorithm is summarized in Algorithm 8 and 9. The complete data is still required to compute the POD basis for initiating the iterative process. As the first solution is approximated, it can be used as the new condition

to compute the POD basis. The difference between L_2 and L_1-based techniques is that the former constructs a least-squares problem for solving $\bar{a}$ as shown in Algorithm 7, and the latter employs greedy matching pursue algorithm to find the sparse coefficients as shown in Algorithm 9. For both approaches, a maximum number of iterations is specified to stop the iterative procedure, e.g., $iter = 100$. Once again, the choice of using L_1 or L_2 minimization depends on whether one expects to find a realistic sparse solution or not.

Algorithm 8: L_1-minimization with unknown basis part I

input : Incomplete data ensemble $\tilde{X} \in \mathbb{R}^{N\times M}$.

Mask matrix $m \in \mathbb{R}^{N\times M}$.

Maximum number of iteration: $iter$.

Sparsity level: s.

halting tolerance: ϵ.

output: Approximated full data ensemble $h \in \mathbb{R}^{N\times M}$.

1 Initialize the iterative variable: $l = 0$.

2 Compute the initial iterative guess h by filling the missing elements with average values.

(a)Sum up the mask matrix for each dimension i: $Q_i = \sum_{j=1}^{M} m_{i,j}$.

(b)Compute the temporal mean for the incomplete data:

$x_{avg,i} = \{\sum_{j=1}^{M} \tilde{X}_{i,j}\}/Q_i$.

(c)Compute the initial guess h: $h^l_{i,j} = \tilde{X}_{i,j}$ if $m_{i,j} = 1$ or $h^l_{i,j} = x_{avg,i}$ if $m_{i,j} = 0$.

3 Choose number of mode retained.

Algorithm 9: L_1-minimization with unknown basis part II

while $l < iter$ **do**
  Compute the POD basis Φ from h^l and $l = l + 1$.
  for *j=1* **to** M **do**
    Compute matrix Θ from mask matrix m and Φ.
    Set iterative variable $k = 0$ and coefficient $\bar{a}^k = 0$.
    Compute the current sample, $v = \tilde{X}$, where $v \in \mathbb{R}^{P\times 1}$.
    while *Halting is false* **do**
      $k = k + 1$
      Form the sample proxy: $y = \Theta^* v$, where * is complex conjugate.
      Identify the large components from the proxy: $\Omega = \text{supp}(y_{2s})$.
      Merge supports: $T = \Omega \cup \text{supp}(\bar{a}^{k-1})$.
      Solve a least squares problem: $b|_T = \Theta_T^+ \tilde{X}$.
      Find the first s largest and nonzero elements: $\bar{a}^k = b_s$.
      Update the samples: $v = \tilde{X} - \Theta\bar{a}^k$ and check halting criterion.
    end
    Compute the approximated solution: $\bar{h} = \Phi\bar{a}$.
    for *i=1* **to** N **do**
      $h_{i,j}^l = \bar{h}_i$ if $m_{i,j} = 0$ or $h_{i,j}^l = h_i^{l-1}$ if $m_{i,j} = 1$
    end
  end
end

3.5.3 Improvement: Progressive Method

The original GPOD method developed by Everson & Sirovich (1995) has several downsides as discussed by Gunes et al. (2006) and Venturi & Karniadakis (2004). The

method breaks down if the matrix $\mathbf{M}$ is singular or if the data information is missing at all time as mentioned previously. The optimal number of modes for repairing the sparse data depends on the sparse measurements, initial guess, and dimensionality of the underlying physics. To overcome the dependency on the initial guess and improve the accuracy of the repaired solution, Venturi & Karniadakis (2004) have extended the original algorithm to a progressive framework. The novelty is to apply the iterative procedure recursively with increasing number of modes. For example, the first iteration procedure will be initiated using only two POD modes. Until the process has converged with two modes, a second iteration procedure is performed using three modes. The number of modes retained is progressively increased for a new iteration process. Although it increases the computational expense significantly, the progressive method has been shown to improve the accuracy of the reconstructed solution as compared to the GPOD procedure (Venturi & Karniadakis 2004). The success of the progressive method is associated with the improved prediction of the higher energy modes from data as compared to the generic L_2 method.

In this study, the progressive method is applied to both L_2 and L_1-based algorithm. The extended version for L_2 minimization with the progressive method is shown in Algorithm 10 and 11. Similarly, the extended version for L_1 minimization with the progressive method is shown in Algorithm 12 and 13. In section 3.6, the corresponding results are presented for both with and without progressive methods.

Algorithm 10: Progressive L_2-minimization with unknown basis part I

input : Incomplete data ensemble $\tilde{X} \in \mathbb{R}^{N\times M}$.

Mask matrix $m \in \mathbb{R}^{N\times M}$.

Maximum number of iteration: $iter$.

output: Approximated full data ensemble $h \in \mathbb{R}^{N\times M}$.

1 Initialize the iterative variable: $l = 0$.

2 Compute the initial iterative guess h by filling the missing elements with average values

(a)Sum up the mask matrix for each dimension i: $Q_i = \sum_{j=1}^{M} m_{i,j}$.

(b)Compute the temporal average for the incomplete data:

$x_{avg,i} = \{\sum_{j=1}^{M} \tilde{X}_{i,j}\}/Q_i$.

(c)Compute the initial guess h: $h_{i,j}^l = \tilde{X}_{i,j}$ if $m_{i,j} = 1$ or $h_{i,j}^l = x_{avg,i}$ if $m_{i,j} = 0$.

3 Choose number of mode retained $K = 2$.

Algorithm 11: Progressive L_2-minimization with unknown basis part II

while $K < Max.\ mode$ **do**
  while $l < iter$ **do**
    Compute the POD basis Φ from h^l
    $l = l + 1$
    for *j=1* **to** M **do**
      for *k=1* **to** K **do**
        for *i=1* **to** N **do**
          Compute gappy basis function: $\tilde{\Phi}_{i,k} = m_{i,j}\Phi_{i,k}$.
        end
      end
      Compute matrix : $\mathbf{M} = \tilde{\Phi}^T\tilde{\Phi}$.
      Compute vector: $f = (\tilde{\Phi})^T h^{l-1}_{:,j}$.
      Solve $\bar{a}$ from a least squares problem: $\mathbf{M}\bar{a} = f$.
      Compute the approximated solution: $\bar{h} = \Phi\bar{a}$.
      Update the repaired data for only missing elements:
      for *i=1* **to** N **do**
        $h^l_{i,j} = \bar{h}_i$ if $m_{i,j} = 0$ or $h^l_{i,j} = h^{l-1}_i$ if $m_{i,j} = 1$
      end
    end
  end
  $l = 0$
  $K = K + 1$
end

Algorithm 12: Progressive L_1-minimization with unknown basis part I

input : Incomplete data ensemble $\tilde{X} \in \mathbb{R}^{N \times M}$.

Mask matrix $m \in \mathbb{R}^{N \times M}$.

Maximum number of iteration: $iter$.

Sparsity level: s.

halting tolerance: ϵ.

output: Approximated full data ensemble $h \in \mathbb{R}^{N \times M}$.

1 Initialize the iterative variable: $l = 0$.

2 Compute the initial iterative guess h by filling the missing elements with average values.

(a)Sum up the mask matrix for each dimension i: $Q_i = \sum_{j=1}^{M} m_{i,j}$.

(b)Compute the temporal average for the incomplete data:

$x_{avg,i} = \{\sum_{j=1}^{M} \tilde{X}_{i,j}\}/Q_i$.

(c)Compute the initial guess h: $h_{i,j}^l = \tilde{X}_{i,j}$ if $m_{i,j} = 1$ or $h_{i,j}^l = x_{avg,i}$ if $m_{i,j} = 0$.

3 Choose number of mode retained $K = 2$.

Algorithm 13: Progressive L_1-minimization with unknown basis part II

while $K < Max.\ mode$ **do**
  while $l < iter$ **do**
    Compute the POD basis Φ from h^l and $l = l + 1$.
    for *j=1* **to** M **do**
      Compute matrix Θ from mask matrix m and Φ.
      Set iterative variable $k = 0$ and coefficient $\bar{a}^k = 0$.
      Compute the current sample, $v = \tilde{X}$, where $v \in \mathbb{R}^{P\times 1}$.
      while *Halting is false* **do**
        $k = k + 1$
        Form the sample proxy: $y = \Theta^* v$, where * is complex conjugate.
        Identify the large components from the proxy: $\Omega = \mathrm{supp}(y_{2s})$.
        Merge supports: $T = \Omega \cup \mathrm{supp}(\bar{a}^{k-1})$.
        Solve a least squares problem: $b|_T = \Theta_T^+ \tilde{X}$.
        Find the first s largest and nonzero elements: $\bar{a}^k = b_s$.
        Update the samples: $v = \tilde{X} - \Theta \bar{a}^k$ and check halting criterion.
      end
      Compute the approximated solution: $\bar{h} = \Phi \bar{a}$.
      for *i=1* **to** N **do**
        $h^l_{i,j} = \bar{h}_i$ if $m_{i,j} = 0$ or $h^l_{i,j} = h^{l-1}_i$ if $m_{i,j} = 1$
      end
    end
  end
  $l = 0$
  $K = K + 1$
end

3.5.4 Statistical Method: Kriging

Statistical estimation such as local kriging can be exploited to interpolate the unknown values using known data (Gunes et al. 2006). The correlogram version of kriging from Lophaven et al. (2002) and Gunes et al. (2006) is adopted and presented as follows. Consider a snapshot of the available data is $\tilde{X} = [\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_{N_1}] \in \mathbb{R}^{N_2 \times N_1}$ at spatial location $\tilde{Z} = [\tilde{\mathbf{z}}_\mathbf{1}, \tilde{\mathbf{z}}_\mathbf{2}, ..., \tilde{\mathbf{z}}_{\mathbf{N_1}}] \in \mathbb{R}^{S \times N_1}$, where N_1 is the number of known data, and N_2 is the number of scalar for the known data. The number of S equals to the dimensionality of the data, i.e., $S = 2$ in this case. The desired unknown data is assumed to be $\bar{x}$ at a particular spatial location $\bar{z} \in \mathbb{R}^S$. The formulation of kriging method is shown in Equation 3.31.

$$\bar{x} = f(\bar{z})\mu + r(\bar{z})\gamma, \tag{3.31}$$

where $f(\bar{z})$ and μ are the regression function and parameters, respectively. $r(\bar{z})$ and γ are the correlation function and parameters. Equation 3.31 is the general form for the kriging estimation with correlation model.

For the regression model, function $f(\bar{z})$ is computed using linear and quadratic terms of $\bar{z}$ as:

$$f_1(\bar{z}) = 1, \quad f_2(\bar{z}) = z_1, \quad ... \quad f_{S+1}(\bar{z}) = z_S, \tag{3.32a}$$

$$f_{S+2}(\bar{z}) = z_1^2, \quad f_{S+3}(\bar{z}) = z_1 z_2, \quad ... \quad f_{2S+1}(\bar{z}) = z_1 z_S, \tag{3.32b}$$

$$f_{2S+2}(\bar{z}) = z_2^2, \quad f_{2S+3}(\bar{z}) = z_2 z_3, \quad ... \quad f_L(\bar{z}) = z_S^2, \tag{3.32c}$$

where $L = (S+1)(S+2)/2$. The regression parameter μ can be computed as:

$$\mu = (F^T R^{-1} F)^{-1} F^T R^{-1} \tilde{Z}^T, \tag{3.33}$$

where the interpolation matrix $F_{i,j} = f_j(\tilde{\mathbf{z}}_\mathbf{i})$, and the correlation matrix is:

$$R_{i,j} = \prod_{k=1}^{S} exp(-\theta(\tilde{Z}_{k,i} - \tilde{Z}_{k,j})^2), \tag{3.34}$$

where $i = 1, 2, ..., N_1$ and $j = 1, 2, ..., N_1$. θ is the bandwidth for Gaussian correlation model. The correlation function $r(\bar{z})$ is computed as:

$$r_{1,i}(\bar{z}) = \prod_{k=1}^{S} exp(-\theta(\tilde{Z}_{k,i} - \bar{z}_k)^2), \tag{3.35}$$

where $i = 1, 2, ..., N_1$. The correlation parameter γ can be computed as:

$$\gamma = R^{-1}(Y^T - F\mu). \tag{3.36}$$

The dimensions for each quantity can be summarized in Table 3.1.

Table 3.1: The Kriging Quantities

Name	Symbol	Dimension
Known data	$\tilde{X}$	$N_2 \times N_1$
Spatial points for known data	$\tilde{Z}$	$S \times N_1$
Unknown data	$\bar{x}$	$N_2 \times 1$
Spatial point for known data	$\bar{z}$	$S \times 1$
Regression function	$f(\bar{z})$	$N_2 \times L$
Regression parameter	μ	$L \times 1$
Correlation matrix	R	$N_1 \times N_1$
Correlation function	$r(\bar{z})$	$N_2 \times N_1$
Correlation parameter	γ	$N_1 \times 1$

The above formulation is basically a least-squares fit with Gaussian correlation, and the detailed discussion of its derivation is referred to Lophaven et al. (2002).Inspired from Gaussian Processes convolution in section 2.4.3, a regression model that is based on GP function, which is referred as GP kriging, is also presented. The basic formulation is similar to Equation 3.31 but without the correlation model:

$$\bar{X} = f(\tilde{Z}, \bar{Z})\mu(\tilde{X}), \tag{3.37}$$

where $\bar{X} \in \mathbb{R}^{N_3 \times 1}$, $F(\bar{Z}) \in \mathbb{R}^{N_3 \times N_1}$, $\bar{Z} \in \mathbb{R}^{N_3 \times S}$, and $\mu \in \mathbb{R}^{N_1 \times 1}$. In this formulation, our regression model is computed as:

$$f(\tilde{Z}, \bar{Z}) = f_{i,j} = \exp \frac{-||\tilde{z}_i - \bar{z}_j||^2}{2\sigma^2}. \tag{3.38}$$

The regression parameter is computed as:

$$\mu(\tilde{X}) = \tilde{f}(\tilde{Z}, \tilde{Z}))^{-1} \tilde{X}. \tag{3.39a}$$

$$\tilde{f}(\tilde{Z}, \tilde{Z}) = \tilde{f}(i,j) = \exp \frac{-||\tilde{z}_i - \tilde{z}_j||^2}{2\sigma^2}. \tag{3.39b}$$

The hyper-parameter σ is user-specified. The difference between the above formulations and GP-convolution is that the unknown data points are assumed to be the centers in Equations 3.38, and the centers in Equation 3.39 are assumed to be the known data measurement points. In comparison to the kriging methods shown in Lophaven et al. (2002), Gunes et al. (2006), the proposed GP kriging employs a least-squares fit, but much straightforward to understand. In the following section, the feasibility of GP kriging as compared to the regular kriging is examined for sparse interpolation of fluid flow data.

3.6 Results and Discussion

3.6.1 Sparse Reconstruction with Known Basis

Flow past a cylinder at $Re = 100$

The first analysis is to reconstruct sparse data with known basis functions that are learned from the full-field data. Note that in most real-world problems, the full field data is not available and so are the POD bases. However, this analysis is useful to quantify the performance of the reconstruction techniques with the basis that are known exactly. In this study, flow past a cylinder at $Re = 100$, which is a low-dimensional system, is investigated. The POD basis functions are extracted using

300 snapshots of data, and the associated energy captured for a different number of POD modes(rank) is shown in Figure 3.1. The number of modes required to capture 95% of the energy content is 4, i.e., $K_{95\%} = 4$.

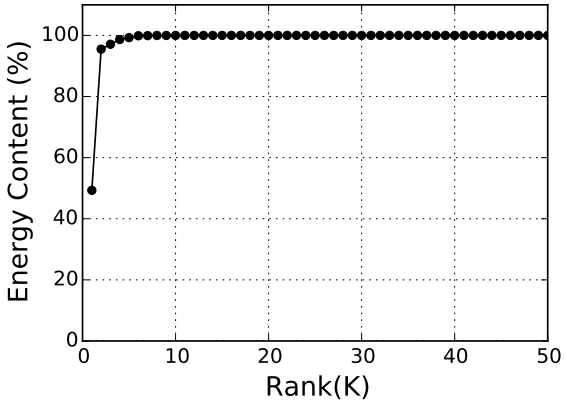


Figure 3.1: Energy captured for cylinder flow at $Re = 100$.

The sparse data is randomly measured from the velocity field with different gappiness level, which is denoted as p. For instance, $p = 0.01$ indicates that 1% of the data is available. The total number of spatial points for the cylinder flow is about 24,000, and the gappiness level p and the corresponding number of measurements are summarized in Table 3.2. The reconstruction is performed by retaining a number of modes ranging from 2 to 70. For L_1 minimization, the sparsity level is chosen as half of the modes retained.

Table 3.2: The gappiness level p and their corresponding number of measurements for reconstruction with known basis for cylinder flow at $Re = 100$.

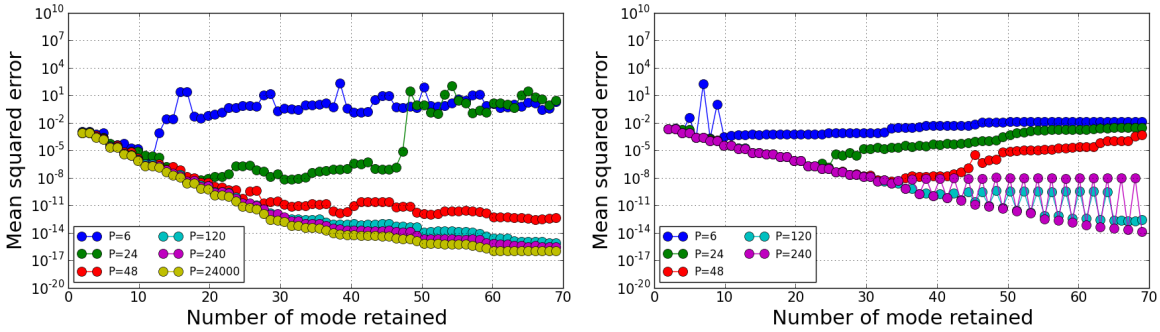
Gappiness level p	0.00025	0.001	0.002	0.005	0.01	1.0
No. of measurements P	6	24	48	240	480	24000

The 30^{th} snapshot from the data ensemble is arbitrarily selected for evaluating the performance of reconstruction techniques. The accuracy of the repaired snapshot

is quantified by defining a mean squared error(MSE) as:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(u_i^{\text{actual}} - u_i^{\text{repair}})^2,$$

where u is a snapshot vector, in this case, the 30^{th} snapshot from the data ensemble. i is the index for the spatial points N, and $N = 24000$. The MSE for the reconstruction from both L_2 and L_1-based methods with known basis is shown in Figure 3.2. For

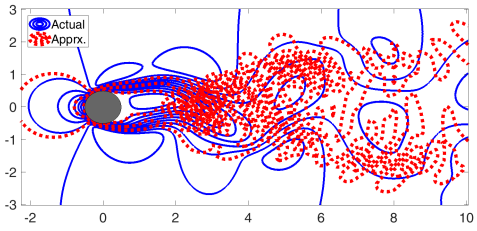


(a) L_2 Minimization

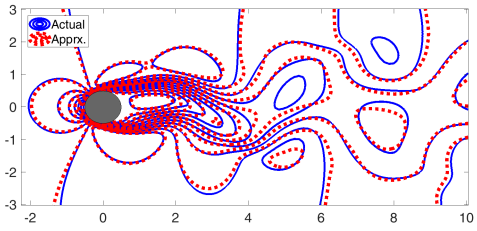
(b) L_1 Minimization

Figure 3.2: The MSE of reconstruction with known basis for cylinder flow at $Re = 100$.

both L_2 and L_1-based methods, the number of measurements for reconstruction must be chosen carefully with respect to the number of modes retained, i.e., the chosen P should be greater than maximum of K and $K_{95\%}$. For example, the MSE appears to be high when $K = 15$ and $P = 6$, and it is low for $K = 15$ and $P = 24$ for L_2 method. On the other hand, the MSE appears to be reasonable when $K = 15$ and $P = 6$ for L_1 method. To further support these observations, the repaired solution fields for $K = 15$ and $P = 6$ are shown in Figure 3.3 for both methods. For comparison, the constructed solutions for $K = 15$ and $P = 24$ are shown in Figure 3.4. L_2 method fails to reconstruct the data when $K = 15$ and $P = 6$ because $P < \max(K, K_{95\%})$ whereas L_1 can outperform L_2 method because the number of P is close to the sparsity level chosen($s = 7$). The reconstructed solutions match with the actual solution perfectly for both cases when $K = 15$ and $P = 24$.

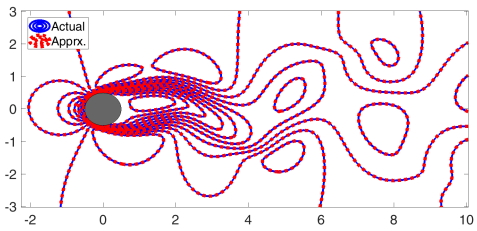


(a) L_2 Method

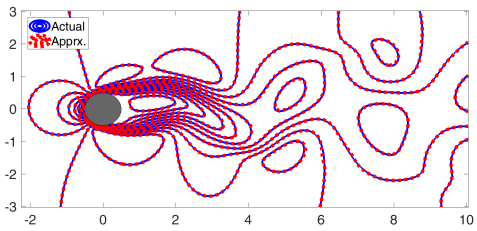


(b) L_1 Method

Figure 3.3: The repaired solution with known basis for cylinder flow at $Re = 100$. $K = 15$ and $P = 6$.



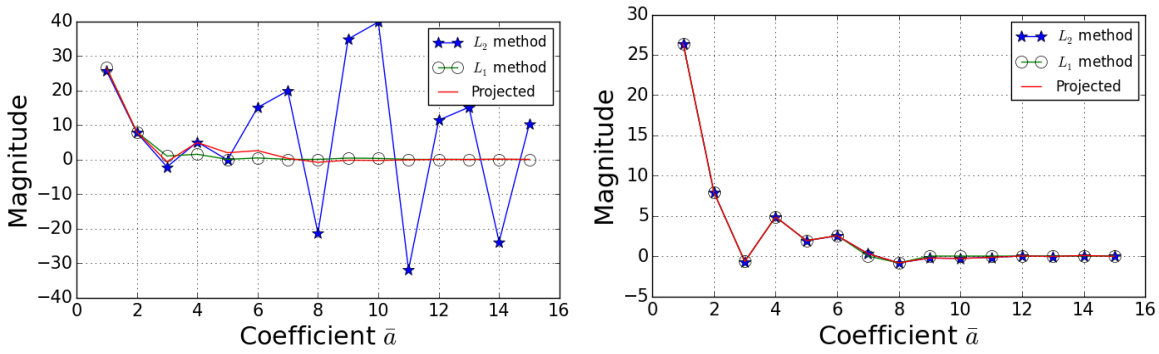
(a) L_2 Method



(b) L_1 Method

Figure 3.4: The repaired solution with known basis for cylinder flow at $Re = 100$. $K = 15$ and $P = 24$.

The approximated coefficients are also presented to further illustrate that they are accurately captured for the L_1-based method in Figure 3.5a. The projected coefficient can be directly computed from the actual solution and the POD basis. When $K = 15$ and $P = 24$, both methods perform reasonably well by capturing all the coefficients. The corresponding results are shown in Figure 3.5b. This analysis proves that $P >= \max(K, K_{95\%})$ is a sufficient condition to obtain an accurate solution for cylinder flow. In general, L_2-based technique performs better than the L_1 case. However, exceptions can be identified when $P < \max(K, K_{95\%})$ but $P \sim s$ for L_1 method.



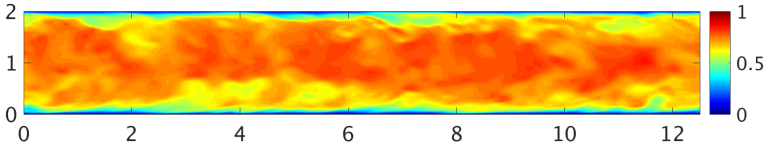
(a) $K = 15$ and $P = 6$

(b) $K = 15$ and $P = 24$

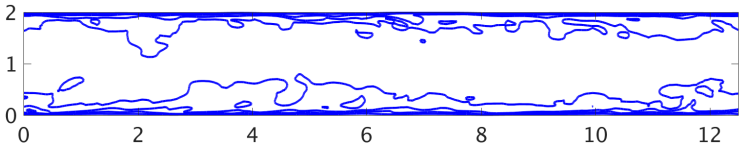
Figure 3.5: The coefficient $\bar{a}$ approximated from known basis for cylinder flow at $Re = 100$.

Channel Flow

In this study, a high-dimensional system, e.g., the channel flow at $Re = 4200$ and $Re_\tau = 180$, is investigated using both L_2 and L_1-based methods for the reconstruction. The 2-D velocity fields at one of the span-wise plane are chosen. 300 snapshots$(651 - 950^{th})$ of data are utilized to build the POD basis. Different than the previous analysis, the reconstruction is performed on the entire data ensemble with $P = 0.005, 0.01, 0.1, 0.25$, and 1.0. The corresponding stream-wise velocity for the 651^{st} snapshot is shown in Figure 3.6. The horizontal direction represents the x-



(a) Colored Contour



(b) Lined Contour

Figure 3.6: The stream-wise velocity for channel flow at 651^{st} snapshot.

coordinate, whereas the vertical direction is given the y-coordinate. The z-coordinate is not shown since this is a 2D extraction from the 3-D data. Otherwise, the z-coordinate should be in/out of the page. The length and height of the channel are 12.6 and 2 unit length, respectively. The total number of spatial points are $N = 16512$ for the 2D plane and the gappiness p is tabulated with the associated number of measurements in Table 3.3.

Table 3.3: The gappiness level p and their corresponding number of measurements for reconstrcution with known basis for channel flow.

Gappiness level p	0.005	0.01	0.1	0.25	1.0
No. of measurements P	83	165	1651	4128	16512

The energy captured for different the number of modes is shown in Figure 3.7. In observation, the higher modes contain a significant amount of energy, and the number of modes required to capture 95% of the energy content is approximately 200, i.e., $K_{95\%} = 200$.

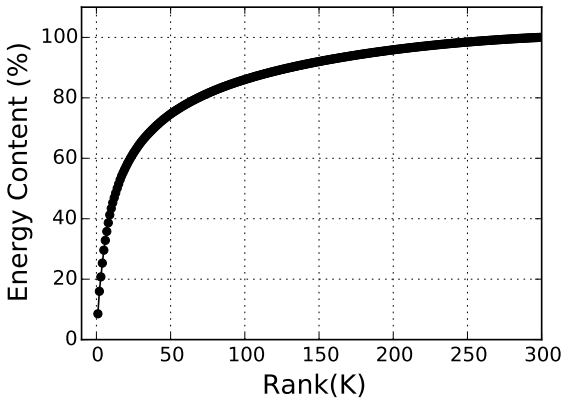
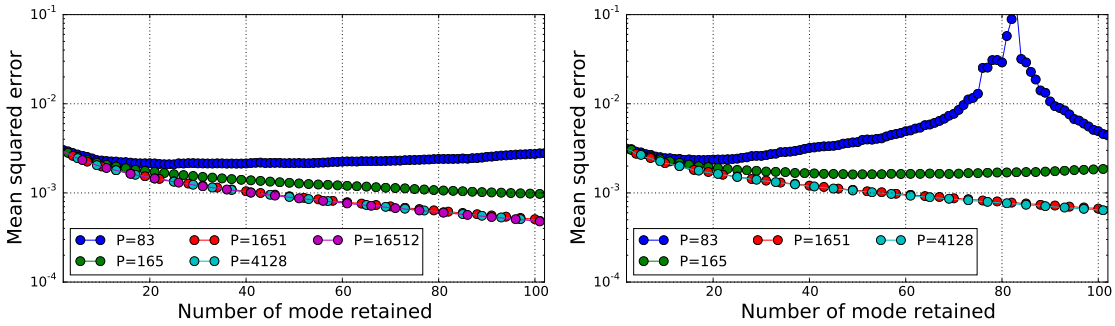


Figure 3.7: Energy captured for channel flow.

In this analysis, the number of modes retained range from 2 to 100. The corresponding MSE for both L_2 and L_1 methods are shown in Figure 3.8. The relation between the number of measurements P and rank K is straightforward. Both methods tend to



(a) L_2 Minimization (b) L_1 Minimization

Figure 3.8: The MSE of reconstruction with known basis for channel flow.

reconstruct better solution as the number of modes increases, and the errors are saturated when $P = 1651$. In particular, when $P = 165$, $K = 100$, and $K_{95\%} = 200$, L_2-based method performs better than the L_1 case. As P increases to 1651, the accuracy resulting from both approaches are comparable.

The approximated solutions for $K = 100$ with $P = 165$ and $P = 1651$ are shown in Figure 3.9 and 3.10 and compared to the actual solution. The POD reconstructed solution and the actual solution are shown in Figure 3.11. Similar to MSE, the reconstructed solution from L_2 method seems to be more accurate as compared to L_1 method for $K = 100$ and $P = 165$. When $K = 100$ with $P = 1651$, both resonstructed solutions appear to have similar level of accuracy. This is because the sparsity level in the L_1 method is half of K, i.e. $s = 50$. Consequently, even if L_1 predicts the first 50 coefficients better, the quality of the reconstruction is not good enough when sufficient amount of energy resides in modes higher than fifty.

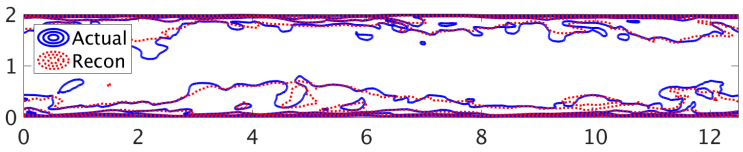
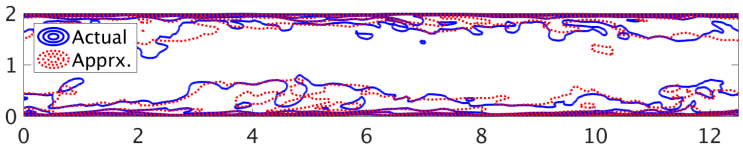
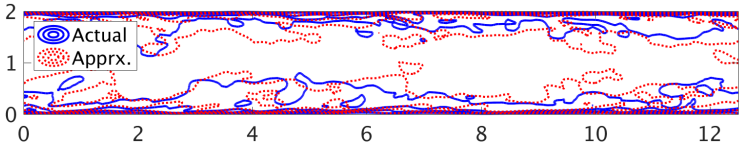


Figure 3.11: The POD reconstruction for channel flow. $K = 100$

Comparing the line contour for one snapshot cannot establish the accuracy trends.

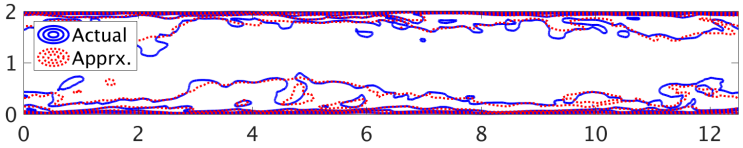


(a) L_2 Method

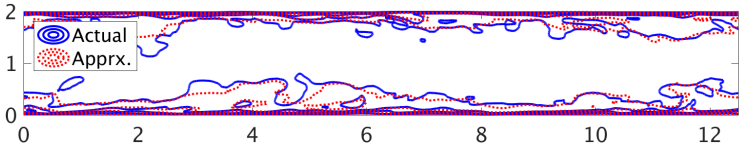


(b) L_1 Method

Figure 3.9: The approximated solution with known basis for channel flow. $K = 100$ and $P = 165$.



(a) L_2 Method



(b) L_1 Method

Figure 3.10: The approximated solution with known basis for channel flow. $K = 100$ and $P = 1651$.

Hence, the turbulent statistics such as ensemble mean, variance, and co-variance are computed. The ensemble mean of the stream-wise and wall-normal velocity are computed as:

$$\bar{u}(y) = \frac{1}{M}\frac{1}{N}\sum_{t=1}^{M}\sum_{x=1}^{N} u(x,y,t), \tag{3.40a}$$

$$\bar{v}(y) = \frac{1}{M}\frac{1}{N}\sum_{t=1}^{M}\sum_{x=1}^{N} v(x,y,t). \tag{3.40b}$$

and the associated variances are computed as:

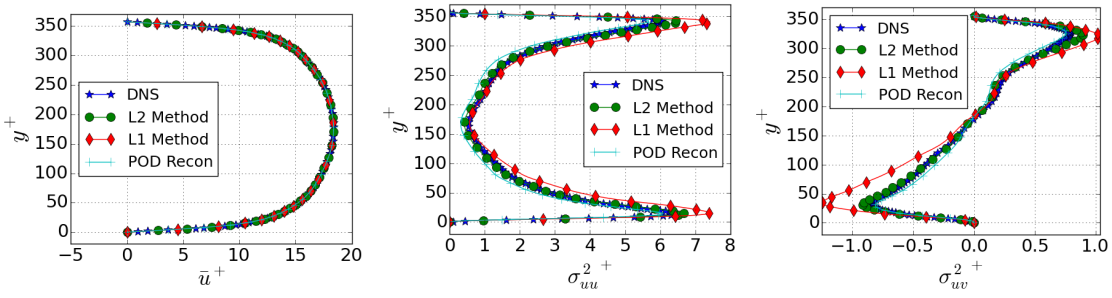
$$\sigma_{uu}^2(y) = \frac{1}{M}\frac{1}{N}\sum_{t=1}^{M}\sum_{x=1}^{N}\big\{u(x,y,t) - \bar{u}(y,t)\big\}^2, \tag{3.41a}$$

$$\sigma_{vv}^2(y) = \frac{1}{M}\frac{1}{N}\sum_{t=1}^{M}\sum_{x=1}^{N}\big\{v(x,y,t) - \bar{v}(y,t)\big\}^2. \tag{3.41b}$$

The covariance is computed as:

$$\sigma_{uv}^2(y) = \frac{1}{M}\frac{1}{N}\sum_{t=1}^{M}\sum_{x=1}^{N}\big\{u(x,y,t) - \bar{u}(y,t)\big\}\big\{v(x,y,t) - \bar{v}(y,t)\big\}. \tag{3.42}$$

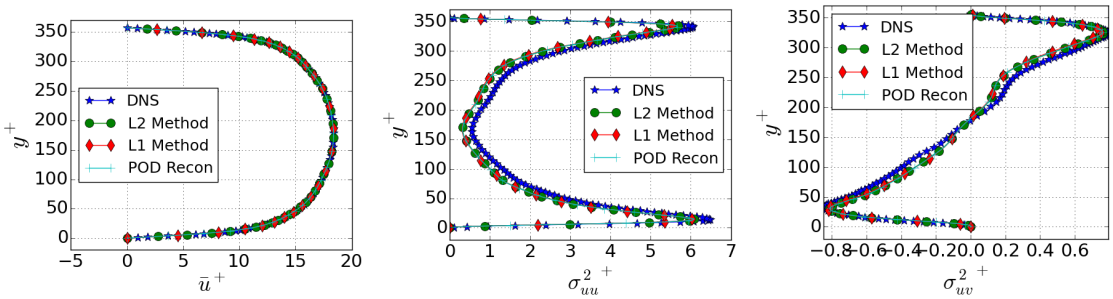
The ensemble mean is normalized as $\bar{u}^+ = y/\tau_u$. The ensemble variance is normalized as ${\sigma_{uu}^2}^+ = \sigma_{uu}^2/\tau_u^2$, and the ensemble co-variance is normalized as ${\sigma_{uv}^2}^+ = \sigma_{uv}^2/\tau_u^2$. The spatial variable y is normalized as $y^+ = y\tau_u/\nu$, where $\tau_u = 0.0425$ is the wall shear stress and $\nu = 1/4200$ is the viscosity. The normalized mean, variance, and co-variance computed from the actual solution(DNS), L_2 and L_1 approximation, and POD reconstruction are shown in Figure 3.12 and Figure 3.13 for $K = 100$ with $P = 165$ and $K = 100$ with $P = 1651$, respectively. From Figure 3.12 and Figure 3.13,



(a) Mean (b) Variance (c) Co-variance

Figure 3.12: The turbulent statistics of reconstruction with known basis for channel flow. $K = 100$ and $P = 165$.

using 165 measurements results in reasonably good predictions of the variance and co-variance for the L_2-based method. This implies that P is not required to be greater than $K_{95\%}$, but P needs to be greater than K for obtaining a reasonably good



(a) Mean (b) Variance (c) Co-variance

Figure 3.13: The turbulent statistics of reconstruction with known basis for channel flow. $K = 100$ and $P = 1651$.

reconstruction of the statistics. On the other hand, the turbulent statistics of the solution approximated from L_1 method are not as good as the L_2 case as the flow does not lend itself into a sparse solution. All the modes are essentially important for reconstructing the flow field, and L_1 method only identify s sparse coefficients. The statistics from both methods have improved when $P = 1651$. For channel flow, satisfying condition of $P > \max(K, K_{95\%})$ produces accurate reconstruction.

Reduced Channel Flow

The previous case was challenging for sparse reconstruction methods due to the high dimensionality of the system. In this study, a small localized region from the channel flow or the *reduced channel*$(0 < y < 0.1)$, is analyzed using L_2 and L_1 reconstruction. This system is considered to be a moderate-dimensional system. The energy capture with a different number of modes retained is shown in Figure 3.14. From observation, the number of modes required to capture 95% of the energy content is approximately 48, i.e., $K_{95\%} = 48$. Hence, this system can be characterized as moderate dimensional.

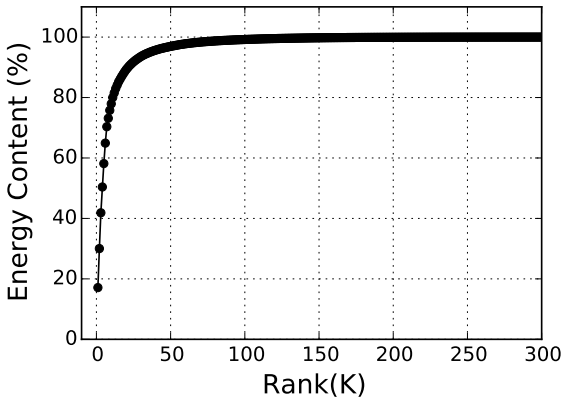
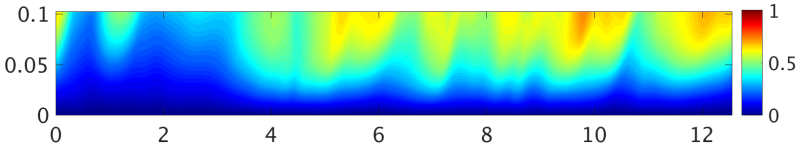
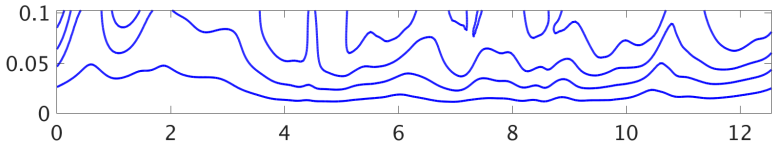


Figure 3.14: Energy captured for reduced channel flow.

The length and the heights are chosen to be 12.6 unit length and 0.1 unit length, respectively. Because the domain of interest is reduced, highly refined resolution data is used for this analysis. The total number of spatial points is 9218, and 300 snapshots $(801 - 1100^{th})$ of data are used to generate the POD basis and reconstruct the sparse data. The corresponding stream-wise velocity for the 801^{st} snapshot is shown in Figure 3.15. In this study, reconstructing sparse data with gappiness levels



(a) Colored Contour



(b) Lined Contour

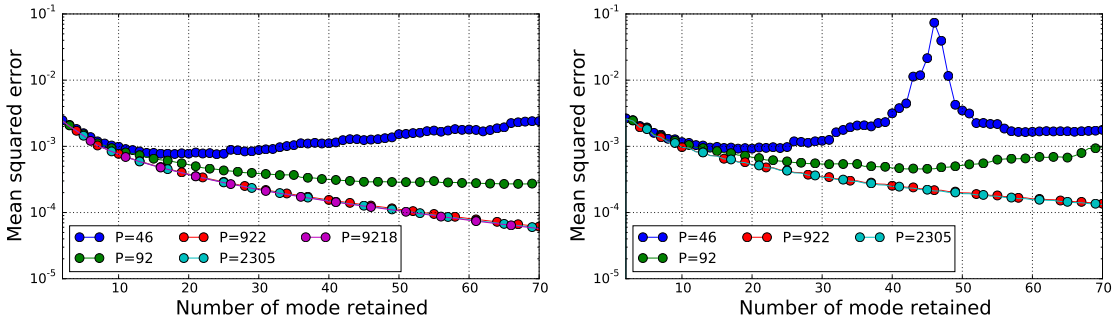
Figure 3.15: The stream-wise velocity for reduced channel flow at 801^{st} snapshot.

of $p = 0.005, 0.01, 0.1, 0.25$, and 1.0 are investigated. The number of p and the associated number of measurements are summarized in Table 3.4. The number of modes retained is ranging from 2 to 70 for recovering the sparse data. The MSE

Table 3.4: The gappiness level p and their corresponding number of measurements for the reconstruction with known basis for reduced channel flow.

Gappiness level p	0.005	0.01	0.1	0.25	1.0
No. of measurements P	46	92	922	2305	9218

for both L_2 and L_1 methods are shown in Figure 3.16. L_2-based technique tends



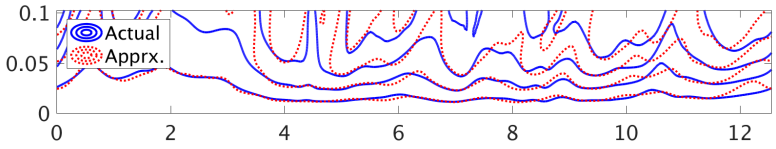
(a) L_2 Minimization

(b) L_1 Minimization

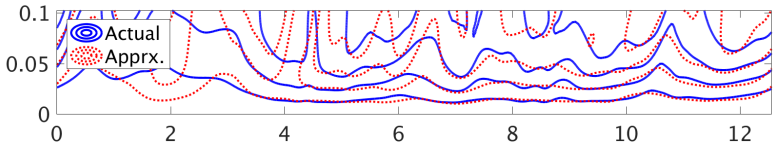
Figure 3.16: The MSE of reconstruction with known basis for reduced channel flow.

to reconstruct accurate and stable solutions when $P = 92$. On the other hand, L_1-based techniques require more measurements to obtain stable solutions, e.g., when $P = 922$. In comparison to the results from channel flow, the order magnitude for MSE has reduced exceptionally from 10^{-3} to 10^{-4}. This improvement is expected since the reduced channel system is lower-dimensional system as compared to the full channel flow. The error become saturated when $P > 922$ for both methods. The comparison between the reconstructed and actual solutions for $K = 50$ with $P = 46$ and $K = 50$ with $P = 92$ are shown in Figure 3.17 and 3.18. The reconstructed solutions from both L_2 and L_1 method are not matching correctly with the actual solution for $K = 50$ and $P = 46$ (i.e. P is less than both K and $K_{95\%}$). From the analysis for both cylinder and channel flow, reconstructing good solution requires $P > \max(K, K_{95\%})$. When $K = 50$ with $P = 92$, both resonstructed solutions have

improved siginificantly and match qualitatively with the acutal solution.

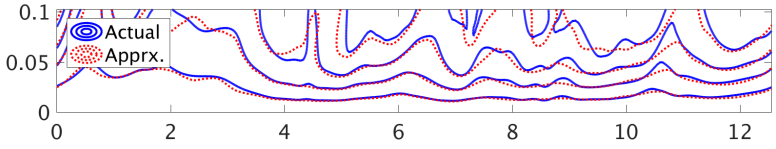


(a) L_2 Method

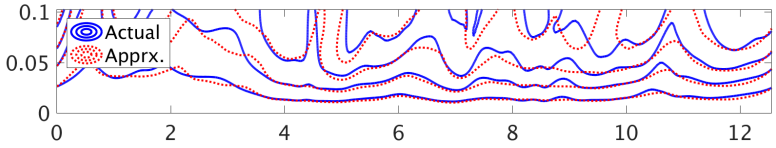


(b) L_1 Method

Figure 3.17: The approximated solution with known basis for reduced channel flow. $K = 50$ and $P = 46$.



(a) L_2 Method



(b) L_1 Method

Figure 3.18: The approximated solution with known basis for reduced channel flow. $K = 50$ and $P = 92$.

The normalized mean, variance, and co-variance computed from the actual solution(DNS), L_2 and L_1-based methods, and direct POD reconstruction are shown in Figure 3.20 and Figure 3.21 for $K = 50$ with $P = 46$and $K = 50$ with $P = 92$, respectively. From the statistics, both methods have improved significantly for capturing variance and co-variance when $P = 92$ instead of $P = 46$. Based on the reconstructed solution fields and statistical results, $P > \max(K, K_{95\%})$ is also a required condition

for reconstructing accurate solutions for reduced channel flow.

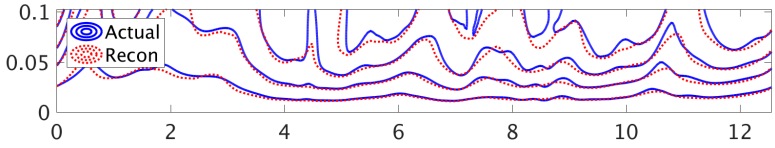
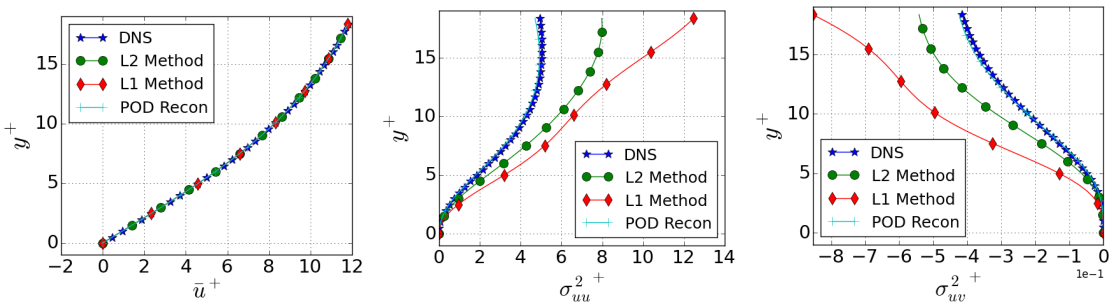
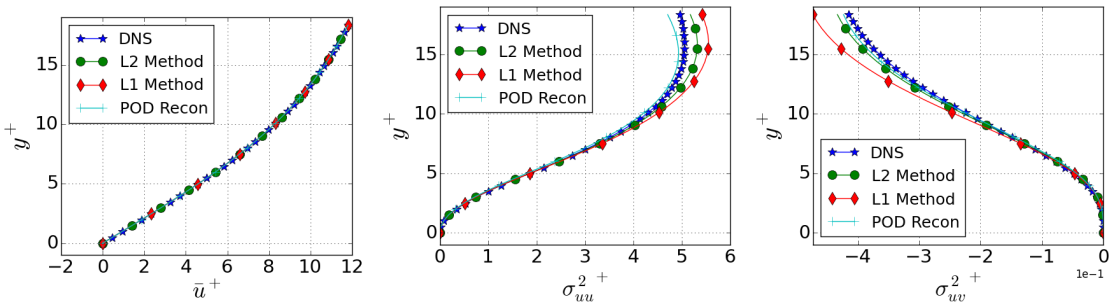


Figure 3.19: The POD reconstruction for reduced channel flow. $K = 50$



(a) Mean (b) Variance (c) Co-variance

Figure 3.20: The turbulent statistics of reconstruction with known basis for reduced channel flow. $K = 50$ and $P = 46$.



(a) Mean (b) Variance (c) Co-variance

Figure 3.21: The turbulent statistics of reconstruction with known basis for reduced channel flow. $K = 50$ and $P = 92$.

In summary, the number measurements P should be chosen wisely for obtaining accurate reconstruction, i.e., $P > \max(K, K_{95\%})$, and $K_{95\%}$ is the number of modes retained so that a sufficient amount of energy content is captured. In general, L_2

method performs better than L_1 when P is chosen sufficiently large as compared to K and $K_{95\%}$. However, L_1-based method can be effective than L_2-based. For instance, when $P < K$ but $P \sim s$, L_1 can identify s sparse coefficients and reconstruct qualitative good solution whereas L_2 cannot. In case of the high-dimensional and moderate-dimensional system, L_2 method is inherently better than L_1 because the low-energy modes are important for reconstructing the physics. L_1 reconstruction loses a significant amount of energy while capturing only s sparse coefficients for the system.

3.6.2 Sparse Reconstruction with Inexact Basis

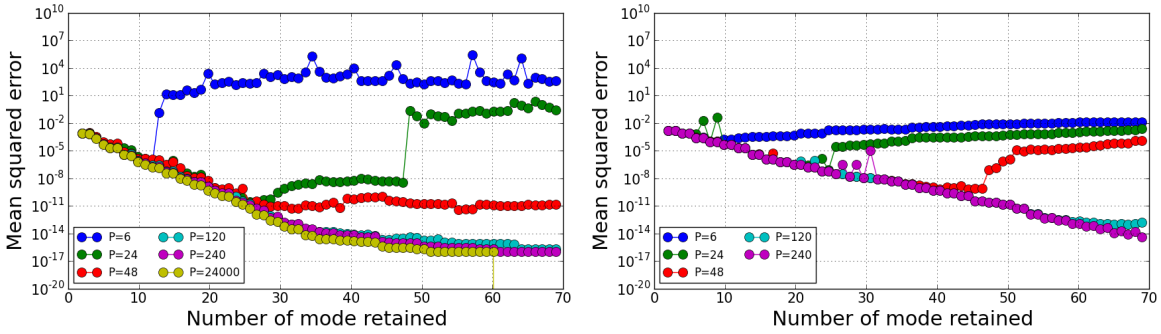
Flow past a cylinder at $Re = 100$

As discussed earlier, the knowledge of the exact POD basis will not be available in most practical situations and alternate approaches need to be identified. In this study, the prospects of using known, but 'inexact' basis that are extracted from the related flow data at different temporal regime or the low-resolution data. In this analysis, the performance of L_2 and L_1 is examined with basis that are learned at different temporal regimes for cylinder flow. 300 snapshots$(1 - 300^{th})$ of complete data are used to extract the POD basis. From the analysis with known basis, $K_{95\%}$ is approximately 4. The sparse data vector is selected to be the 340^{th} snapshot, which is outside the training data. The location of the measuring points are randomly chosen with a number of gappiness level p. The corresponding number of measurements P are tabulated in Table 3.5. The sparse reconstruction is performed by retaining a

Table 3.5: The gappiness level p and their corresponding number of measurements for reconstruction with inexact basis for cylinder flow at $Re = 100$.

Gappiness level p	0.00025	0.001	0.002	0.005	0.01	1.0
No. of measurements P	6	24	48	240	480	24000

number of modes ranging from 2 to 70. For L_1 minimization, the sparsity level is chosen as half of the modes retained. The MSE for the reconstruction from both L_2 and L_1-based methods with inexact basis is shown in Figure 3.22. By comparing

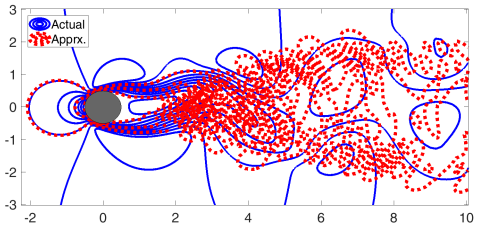


(a) L_2 Minimization

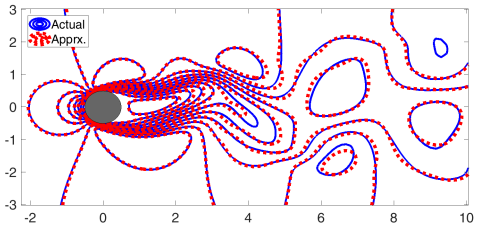
(b) L_1 Minimization

Figure 3.22: The MSE of reconstruction with inexact basis for cylinder flow at $Re = 100$.

Figure 3.2 and 3.22, the MSE are almost identical whether the known basis or inexact basis are used. The number of P should be chosen greater than maximum of K and $K_{95\%}$ to ensure a good reconstruction. The reconstructed solution fields for $K = 15$ and $P = 6$ are shown in Figure 3.23 for both L_2 and L_1 cases. Similar to the case with known basis, L_2-based method does not perform well when P is is less than K. For L_1 method, the reconstructed solution is qualitatively matching with the actual solution because of $P \sim s$. The reconstructed solution fields for $K = 15$ and $P = 24$ are shown in Figure 3.24. Both methods perform reasonably well in recovering the flow fields which is also observed in case of known basis. In conclusion, the choice of temporal regime does not necessarily impact the performance of sparse reconstruction for cylinder flow, which is a low-dimensional and limit-cycle system. Satisfying the condition $P > \max(K, K_{95\%})$ is still required for obtaining accurate reconstruction.

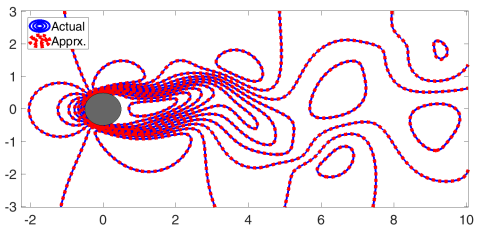


(a) L_2 Method

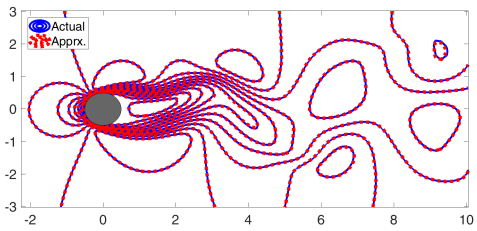


(b) L_1 Method

Figure 3.23: The repaired solution with inexact basis for cylinder flow at $Re = 100$. $K = 15$ and $P = 6$.



(a) L_2 Method



(b) L_1 Method

Figure 3.24: The repaired solution with inexact basis for cylinder flow at $Re = 100$. $K = 15$ and $P = 24$.

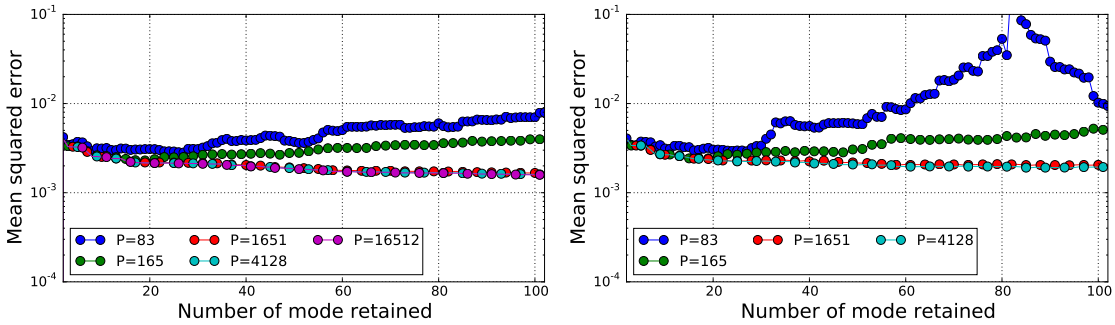
Channel Flow

In this analysis, the performance of L_2 and L_1 is examined with basis that are learned at different temporal regime for channel flow, and $300(651 - 950^{th})$ snapshots of data are utilized to build POD basis. The incomplete data is chosen as the 990^{th} snapshot, which is outside the training data. The location of the measuring points are randomly chosen with a number of gappiness level p. From the study with known basis, $K_{95\%} = 200$. The corresponding number of measurements P are tabulated in Table 3.6. In this analysis, the sparse reconstruction is performed by retaining a number of modes ranging from 2 to 100. The MSE for both L_2 and L_1 methods are shown in Figure 3.25. The level of accuracy has decreased as compared to the analysis with known basis, i.e., from $\mathcal{O}(10^{-3})$ to $\mathcal{O}(5^{-2})$. Similar to the results with known basis, both methods seem

Table 3.6: The gappiness level p and their corresponding number of measurements for reconstrcution with inexact basis for channel flow.

Gappiness level p	0.005	0.01	0.1	0.25	1.0
No. of measurements P	83	165	1651	4128	16512

to reconstruct better solution as the number of modes increases, and the errors tend to be sataurated when $P = 1651$. Because the reconstructed solution fields do not give

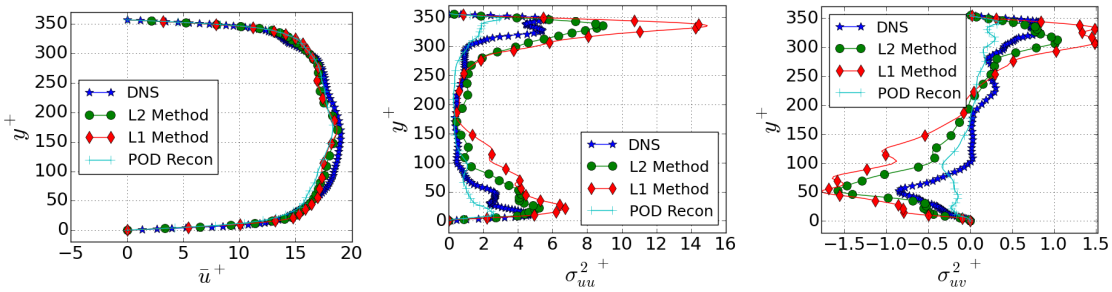


(a) L_2 Minimization

(b) L_1 Minimization

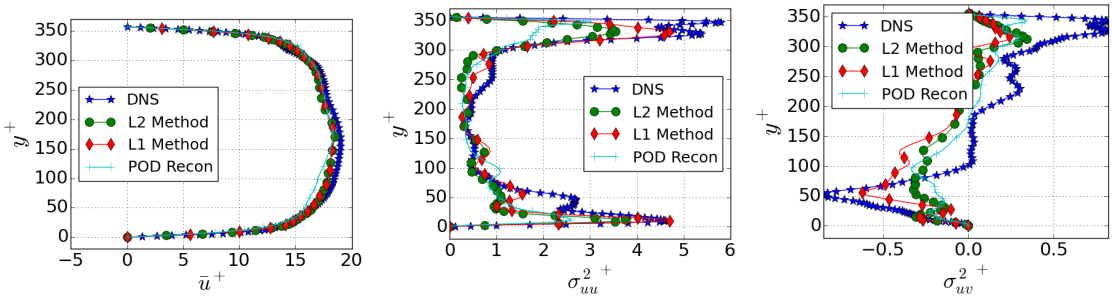
Figure 3.25: The MSE of reconstruction with inexact basis for channel flow.

meaningful interpretation for the channel flow, the normalized mean, variance, and co-variance are shown in Figure 3.26 and 3.27 for $K = 100$ with $P = 165$ and $K = 100$ with $P = 1651$, respectively. Since this analysis recovers only a single snapshot, the statisics shown in both Figure 3.26 and 3.27 are not smooth. For $P = 100$ and $P = 165$, both the variance and co-variance are poorly reconstructed as compared to the results from known basis. As the number of measurements increase to $P = 1651$, both statistical values have improved significantly. Satifying $P > \max(K, K_{95\%})$ becomes important when using the inexact basis that are learned from different temporal region for channel flow. Both L_2 and L_1 methods fail to reconstruct accurate solutions if P is not chosen correctly.



(a) Mean (b) Variance (c) Co-variance

Figure 3.26: The turbulent statistics of reconstruction with inexact basis for channel flow. $K = 100$ and $P = 165$.



(a) Mean (b) Variance (c) Co-variance

Figure 3.27: The turbulent statistics of reconstruction with inexact basis for channel flow. $K = 100$ and $P = 1651$.

Reduced Channel Flow

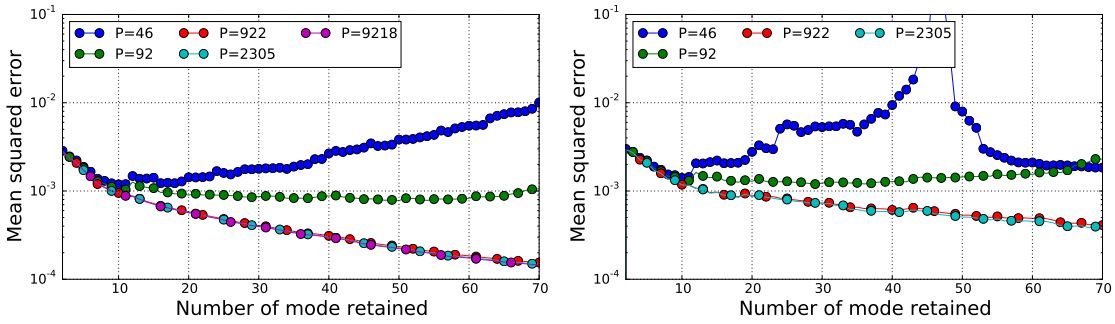
For the reduced channel flow, the low-resolution data is utilized to recover sparse measurements that are collected from fine-resolution data. The goal of this study is to explore how the accuracy of the solution can impact the reconstruction. GP kriging is exploited to map the low-resolution data into fine grid solution, and 300 snapshots of the low-resolution data are used to build POD basis. The number of modes required to capture 95% of the energy content is approximately 48, i.e., $K_{95\%} = 48$. The total number of spatial points for the low and high-resolution data are 2304 and 9218, respectively.

The number of gappiness p and measurements P are summarized in Table 3.7. The number of modes retained is ranging from 2 to 70 for data recovery, and the

Table 3.7: The gappiness level p and their corresponding number of measurements for the reconstruction with known basis for reduced channel flow.

Gappiness level p	0.005	0.01	0.1	0.25	1.0
No. of measurements P	46	92	922	2305	9218

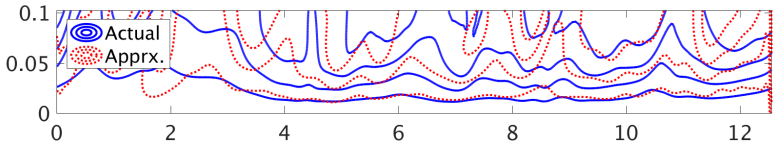
MSE for both L_2 and L_1 methods are shown in Figure 3.28. The results show that L_2 performs better than L_1 when $P > 92$. As P increases to 922, the errors appear to be saturated for both methods. The comparison between the reconstructed and actual solutions for $K = 50$ with $P = 46$ and $K = 50$ with $P = 92$ are shown in Figure 3.29 and 3.30, respectively. When $P < K$, the reconstructed solutions from both L_2 and L_1 methods do not match the actual solution. When $P = 92$, the reconstructed solution from L_2 method has improved siginificantly. Although L_1 method seems to improve the reconstruction, the line contour shown in Figure 3.30 does not match very well with the actual solution. Furthermore, the reconstructed solutions for $K = 50$



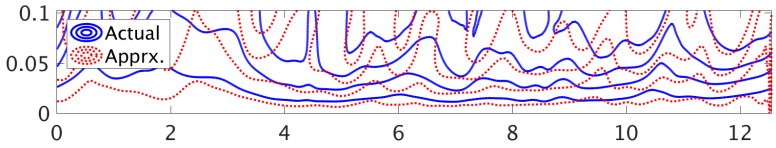
(a) L_2 Minimization (b) L_1 Minimization

Figure 3.28: The MSE of reconstruction with inexact basis for reduced channel flow.

with $P = 922$ are also shown in Figure 3.31. Both L_2 and L_1 methods does not show percetible improvements. The normalized mean, variance, and co-variance are



(a) L_2 Method

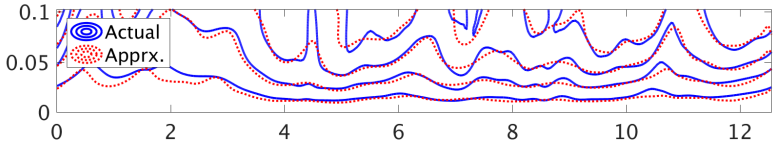


(b) L_1 Method

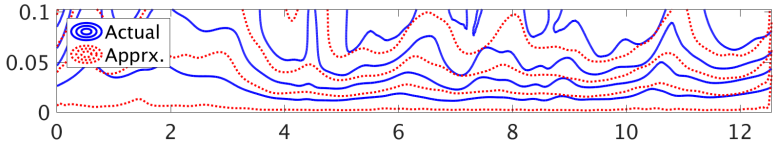
Figure 3.29: The approximated solution with inexact basis for reduced channel flow. $K = 50$ and $P = 46$.

also computed from the actual solution(DNS), L_2 and L_1 approximation, and POD reconstruction in Figure 3.26, 3.27, and 3.27 for $K = 50$ with $P = 46$, $K = 50$ with $P = 92$, and $K = 50$ with $P = 822$, respectively.

It is observed that for all three cases, the reconstructed statistics are not as good as the results using the known basis, and the MSE also increased drastically with the inexact basis. For this last study that employs low-resolution data, the reconstruction from L_2 method for $P = 92$ improves the variance and co-variance as compared to the

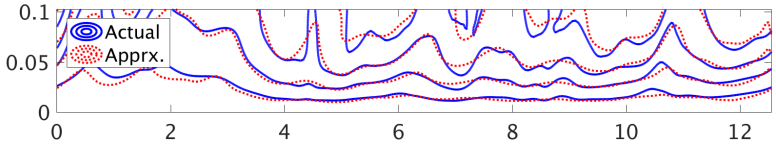


(a) L_2 Method

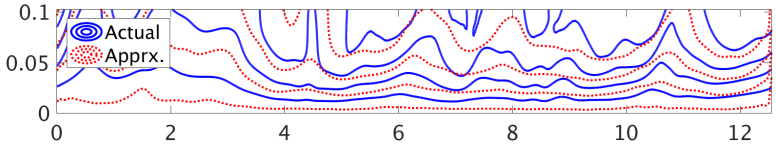


(b) L_1 Method

Figure 3.30: The approximated solution with inexact basis for reduced channel flow. $K = 50$ and $P = 92$.



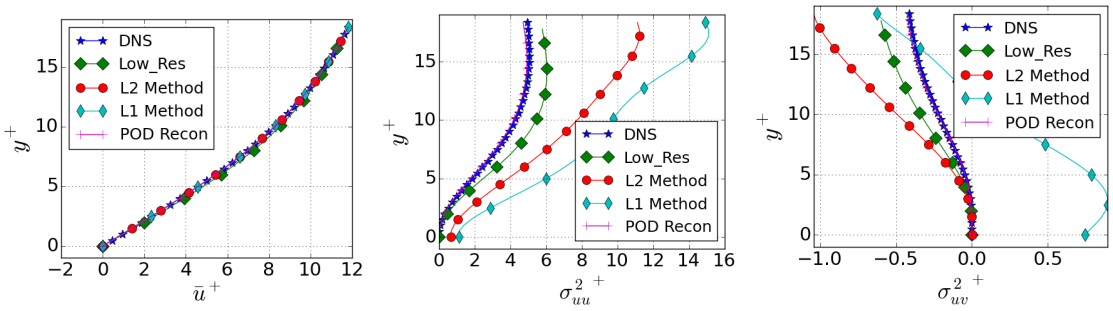
(a) L_2 Method



(b) L_1 Method

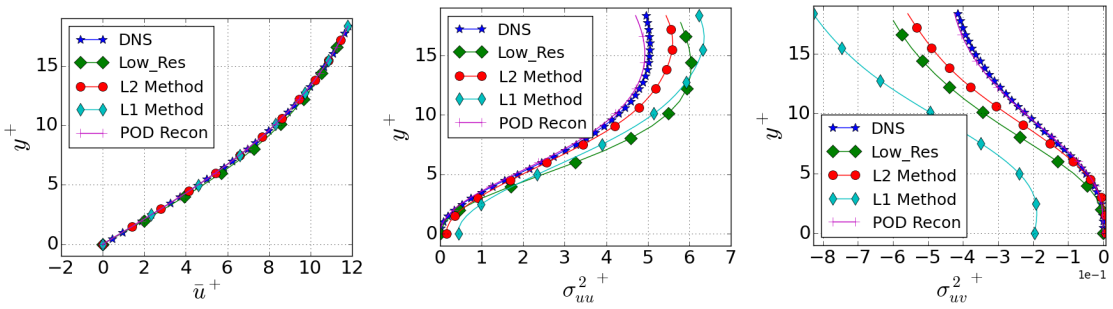
Figure 3.31: The approximated solution with inexact basis for reduced channel flow. $K = 50$ and $P = 922$.

statistics from low-resolution data. When $P = 922$, both variance and co-variance are reconstructed reasonably accurately as compared to the DNS data. For the L_1 method, the variance and co-variance are improved when the number of measurements has been increased to 922, but they are not as accurate as the statistics from L_2 method. This analysis has shown that L_2 method is advantageous than L_1 in case of inexact basis for systems that are not truly sparse. The reason is that L_1 identifies only sparse coefficients which the important modes cannot be captured. Another



(a) Mean (b) Variance (c) Co-variance

Figure 3.32: The turbulent statistics of reconstruction with inexact basis for reduced channel flow. $K = 50$ and $P = 46$.



(a) Mean (b) Variance (c) Co-variance

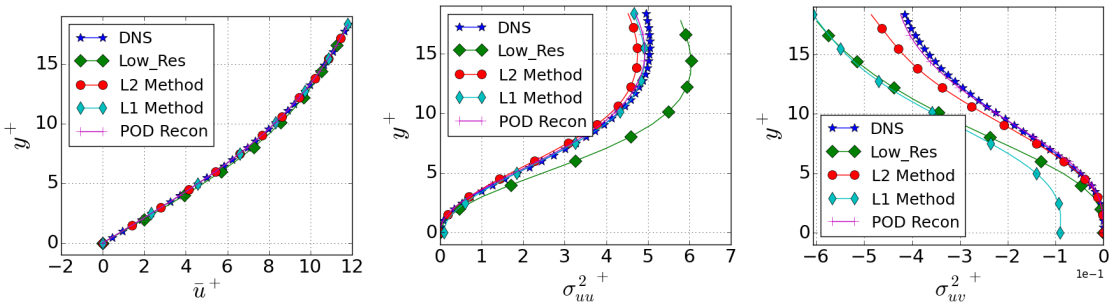
Figure 3.33: The turbulent statistics of reconstruction with inexact basis for reduced channel flow. $K = 50$ and $P = 92$.

significant remark is that L_2 method can improve the solution even with the inexact basis functions that are learned from low-resolution data. The required condition is that $P > \max(K, K_{95\%})$.

3.6.3 Sparse Reconstruction with Unknown Basis

Flow past a cylinder at $Re = 100$

Sparse reconstruction with the unknown basis is explored in this section. Different from the earlier analysis in section 3.6.1 and 3.6.2, the basis functions are treated as completely unknown. Such a system is harder to reconstruct. In this analysis,



(a) Mean (b) Variance (c) Co-variance

Figure 3.34: The turbulent statistics of reconstruction with inexact basis for reduced channel flow. $K = 50$ and $P = 922$.

the sparse reconstruction is performed with given number of measurements, and the effectiveness of iterative methods and kriging is investigated. For cylinder flow, a gappiness level $p = 0.25$ has been previously reported in Gunes et al. (2006) for obtaining accurate reconstruction with unknown basis. Therefore, 6000 measurements are randomly collected from DNS data for sparse reconstruction, and the measured points for one such snapshot in Figure 3.37a. Since the cylinder flow is a low-dimensional system, 100 snapshots of data are sufficient for this study. The number of modes retained is 15, which captures 99.999% of the energy content.

In the absence of prior knowledge of the basis functions, both POD basis and coefficients are solved iteratively. The detailed procedures can be referred to section 3.5. In this analysis, the iterative methods such as L_2, L_1, progressive L_2, progressive L_1, kriging with L_2, kriging with L_1, GP kriging with L_2, and GP kriging with L_1 are employed for sparse reconstruction. The iterative processes for all the cases stop at 100 which represents convergence for almost all methods. For the progressive method, the procedure starts from mode 2 to mode 15 with an increment of 1. Each mode proceeds 100 iterations. The kriging methods are used to generate the initial condition for starting the iterative procedure. Alternatively, regular ensemble average is exploited to fill the missing elements for the sparse data.

Firstly, the performance and convergence of each iterative solution is evaluated by computing the eigenvalues and eigenvectors from the co-variance matrix of the solution (Everson & Sirovich 1995). The solution is denoted as $X \in \mathbb{R}^{N \times M}$ which can be the true solution, reconstructed solution from the POD procedure, or the reconstructed solution from iterative procedures. The co-variance matrix $Z \in \mathbb{R}^{M \times M}$ is computed as:

$$Z = X^T X. \tag{3.43}$$

The eigenvalus and the corresponding eigenvectors of the covariant matrix Z satisfy:

$$Z\psi_K = \lambda_K \psi_K. \tag{3.44}$$

The inner product between the true eigenvectors should be either 1 or 0 if they are orthogonal. Hence, the summation of the inner product Ψ_K for the exact eigenvectors and reconstructed eigenvectors is computed in Equation 3.45. Ψ_K allows to see whether the repaired eigenvectors are close to the exact eigenvectors. In the event of perfect recovery, Ψ_K should equals to number of modes retained, i.e., $\Psi_K = K$.

$$\Psi_K = \sum_{i=1}^{K} \left| \psi_i^{\text{repair}} \cdot \psi_i^{\text{exact}} \right| \tag{3.45}$$

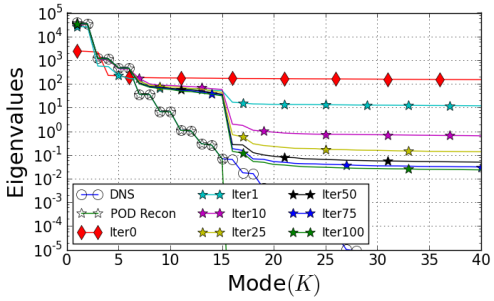
The eigenvalues for all the cases with unknown basis for cylinder flow are shown in Figure 3.35, and the corresponding Ψ_K are shown in Figure 3.36.

From both Figure 3.35 and 3.36, the L_1 method captures about 9 modes correctly whereas the L_2 method captures only about 6 modes. The progressive method has significantly improved the performance of the L_2 method, i.e., all 15 modes are perfectly captured. When kriging methods are used to obtain initial condition, the number of eigenvalues and eigenmodes is increased for L_2 method. For the L_1 method, both kriging methods do not impact the reconstruction performance. The number of eigenvalues and eigenvectors captured is summarized in Table 3.8 for all the cases. The reconstructed stream-wise velocity field at iteration 100 are shown in Figure 3.37, and

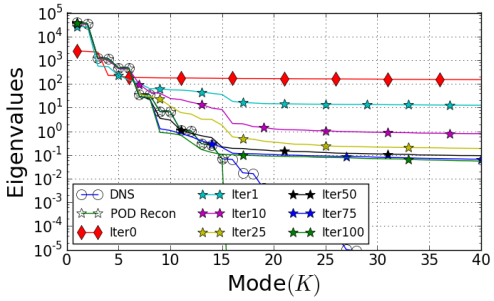
Table 3.8: The number of eigenvalues and eigenvectors captured with unknown basis for cylinder flow.

Method	Eigenvector	Eigenvalue
L_2	6	6
L_1	8	8
Progressive L_2	15	15
Progressive L_1	9	9
Kriging + L_2	10	10
Kriging + L_2	10	10
GP kriging + L_2	8	8
GP kriging + L_1	8	8

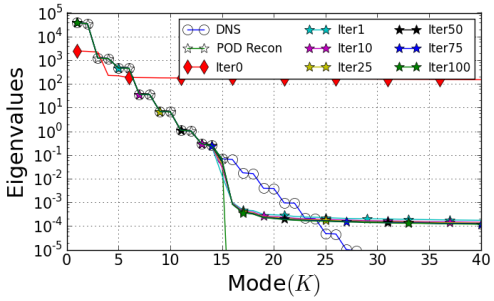
all the methods perform well for repairing the cylinder flow except for small discrepancies between L_2 and L_1 methods. For unknown basis, the pure L_2 and L_1 methods can only capture about half of the modes correctly even though a large number of measurements are obtained. However, capturing six modes results in fairly accurate solutions because it contains more than 99% of the energy content. The progressive L_2 method works the best, i.e., it almost captures fifteen modes exactly, but the computational effort becomes expensive since each mode requires nearly 100 iterations.



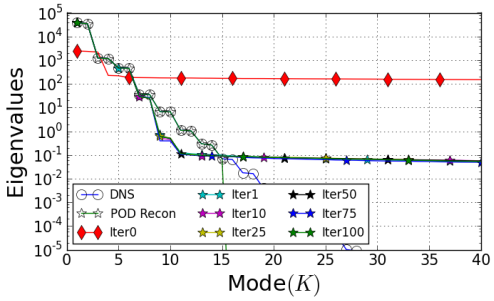
(a) L_2 method



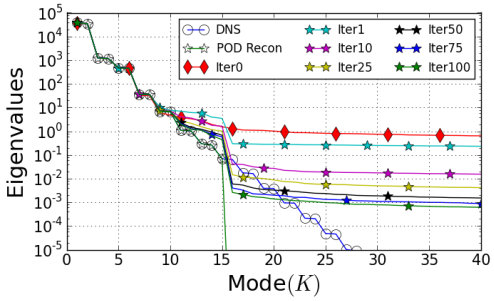
(b) L_1 method



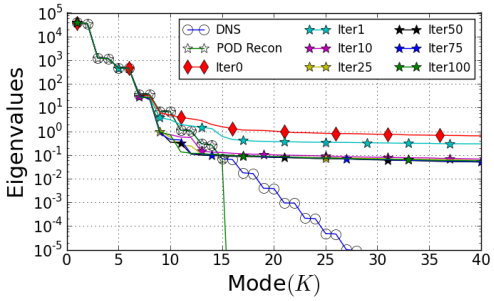
(c) Progressive L_2 method



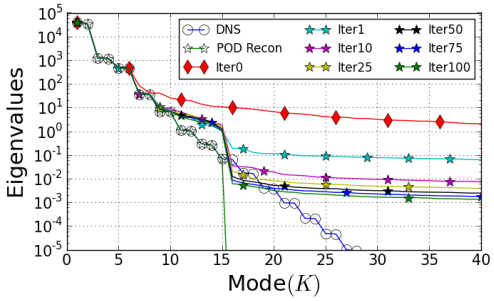
(d) Progressive L_1 method



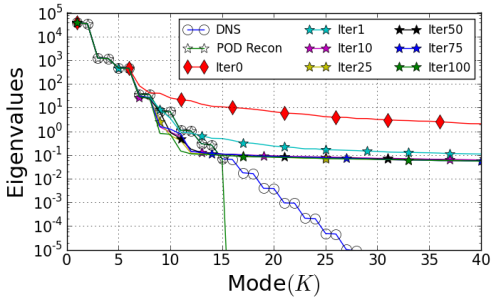
(e) Kriging L_2 method



(f) Kriging L_1 method

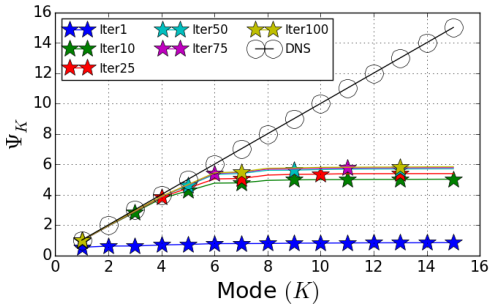


(g) GP Kriging L_2 method

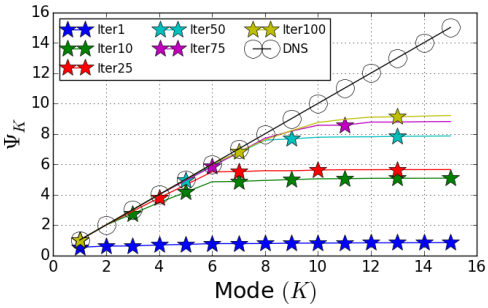


(h) GP Kriging L_1 method

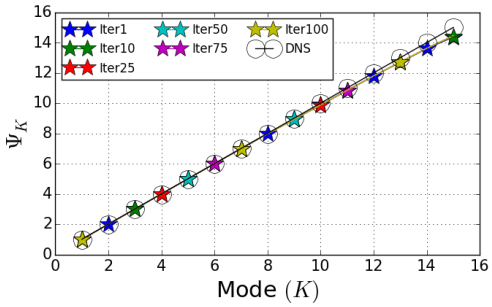
Figure 3.35: The eigenvalues the eigenfunction of the reconstructed solution for all the cases with unknown basis for cylinder flow at $Re = 100$.



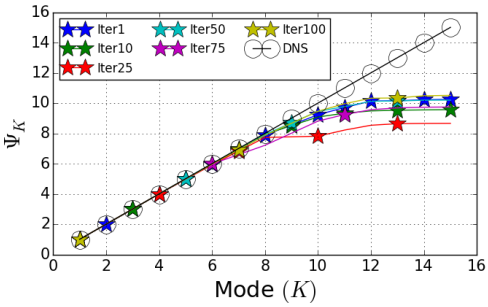
(a) L_2 method



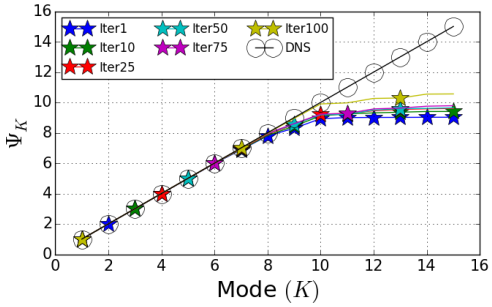
(b) L_1 method



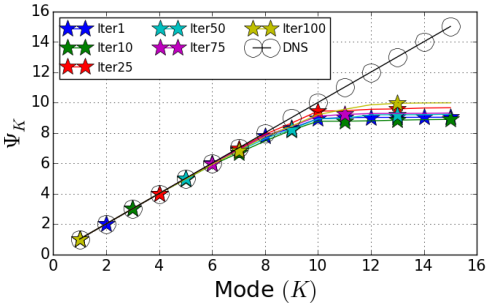
(c) Progressive L_2 method



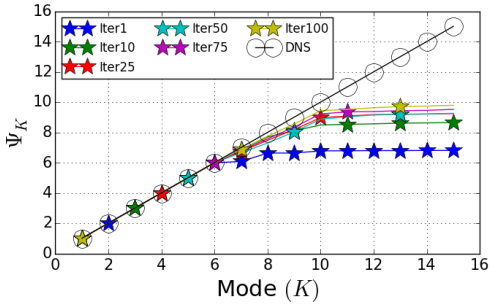
(d) Progressive L_1 method



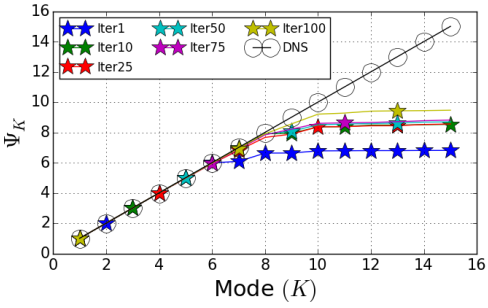
(e) Kriging L_2 method



(f) Kriging L_1 method

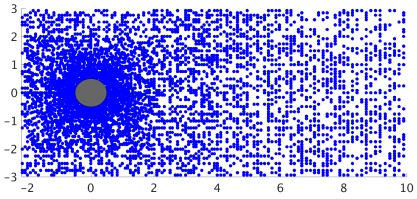


(g) GP Kriging L_2 method

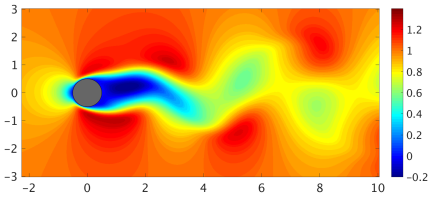


(h) GP Kriging L_1 method

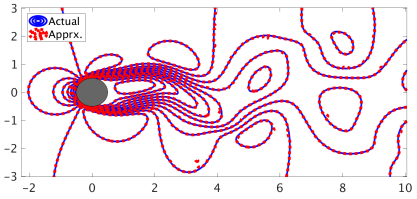
Figure 3.36: The computed Φ_K of the reconstructions for all the cases with unknown basis for cylinder flow at $Re = 100$.



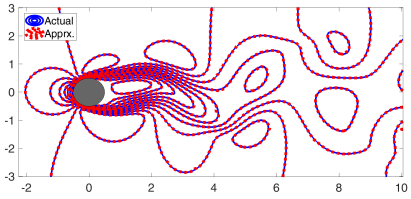
(a) Measured Field



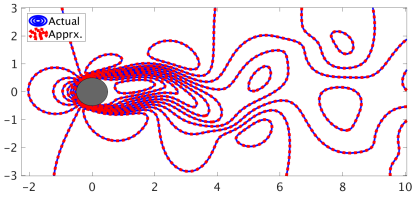
(b) Actual Solution



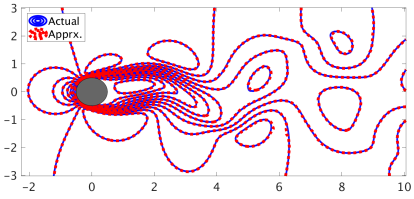
(c) L_2 method



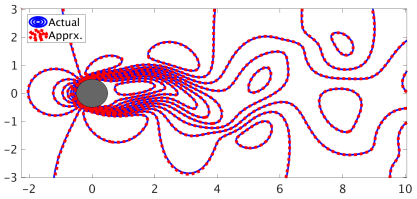
(d) L_1 method



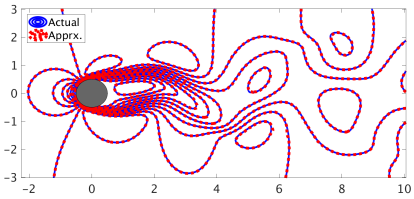
(e) Progressive L_2 method



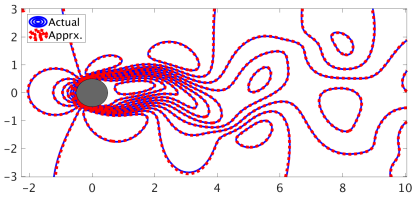
(f) Progressive L_1 method



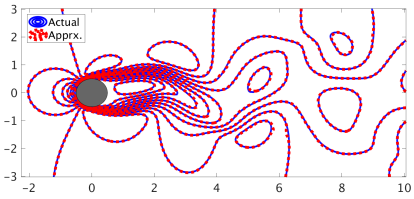
(g) Kriging + L_2 method



(h) Kriging + L_1 method



(i) GP Kriging + L_2 method



(j) GP Kriging + L_1 method

Figure 3.37: The reconstructed stream-wise velocity field for all the cases with unknown basis for cylinder flow at $Re = 100$.

Channel Flow

For the channel flow, the appropriate choice of gappiness level p is not known, so $p = 0.25$ is chosen based on observing the cylinder flow. The sparse measurements are collected from the low-resolution channel flow data. Similar to the reconstruction with known basis, 300 snapshots of data are utilized to obtain sparse measurements, and 100 modes are retained. Because a large number of modes are retained, the progressive method is not computationally feasible as a serial process. Efforts are being made to address this situation. From prior reconstruction efforts for channel flow, the L_2 method was observed to be advantageous than L_1 for a high-dimensional system. Therefore, L_2, statistical kriging with L_2, and GP kriging with L_2 are used for the sparse reconstruction. Kriging is employed to generate the initial solution field that initializes the iterative procedure.

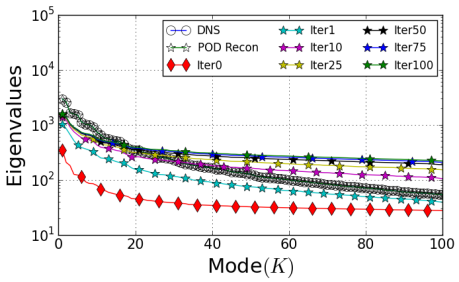
The computed eigenvalues and Φ_K from the co-variance matrix of the solution are shown in Figure 3.38 and 3.39, respectively. The pure L_2 method performs poorly because only about 10 eigenvectors and 5 eigenvalues are captured correctly. In comparison, the kriging with L_2 method captures about 25 eigenvectors and 10 eigenvalues whereas GP kriging with L_2 is reconstructing 20 eigenvectors and 10 eigenvalues correctly. The number of eigenvalues and eigenvectors captured is summarized in Table 3.9 for all three cases. This analysis emphasizes that retaining 100 modes can only reconstruct up to 25 modes correctly for a high-dimensional system. In this case, 100 modes represent 86% of the energy for the system, and only 50% of the energy content are captured. Naturally, this will lead to poor quality of solution reconstruction. Further, the choice of the initial guess from Kriging plays a critical role in the quality of the reconstruction.

Even though the initial condition from kriging methods have improved the results

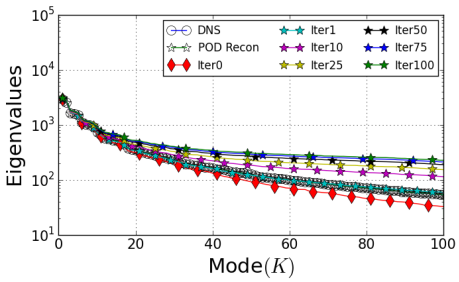
Table 3.9: The number of eigenvalues and eigenvectors captured with unknown basis for channel flow.

Method	Eigenvector	Eigenvalue
L_2	10	5
Kriging + L_2	25	10
GP kriging + L_2	20	10

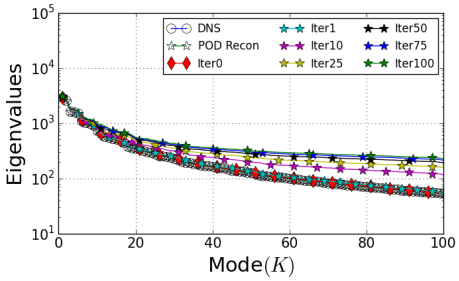
as compared to the pure L_2 method, both eigenvalues and Φ_K from iteration 100 are less accurate than iteration 1. The initial solutions from the kriging methods are shown in Figure 3.40. This indicates that the iterative method does not always converge into the correct solution and can degrade the reconstruction. Particularly, the reconstructed solutions appear to introduce noises for all the cases. This is because the low energy modes that carry significant energy are not accurately captured. The stability characteristics of this iterative procedure need to be further investigated. The corresponding turbulent statistics are shown in Figure 3.41, 3.42, and 3.43. On comparison, the mean, variance, and covariance from all three cases are comparable with the statistical analysis with the case with known basis. Because the solutions are very noisy, the statistics from the reconstruction are not as accurate as the DNS data.



(a) L_2 method

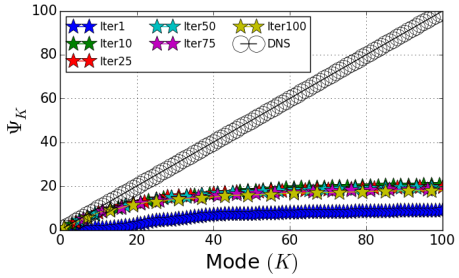


(b) Kriging + L_2 method

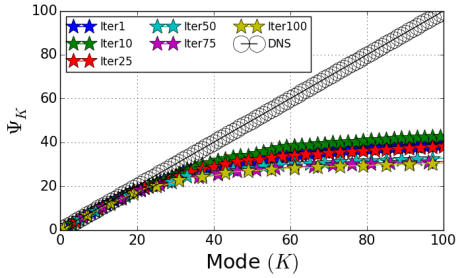


(c) GP Kriging + L_2 method

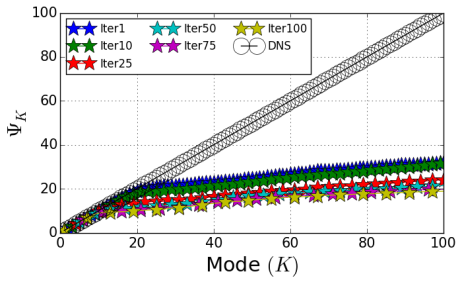
Figure 3.38: The eigenvalues of the reconstructions for all the cases with unknown basis for Channel flow.



(a) L_2 method

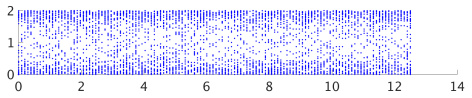


(b) Kriging L_2 method

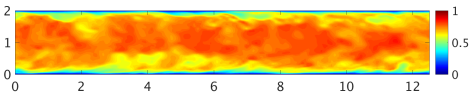


(c) GP Kriging L_2 method

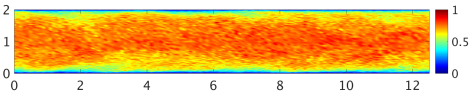
Figure 3.39: The computed Φ_K of the reconstructions for all the cases with unknown basis for Channel flow.



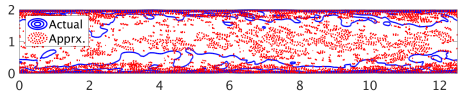
(a) Measured Field



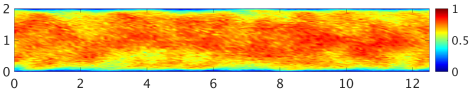
(b) Actual Solution



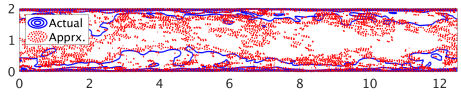
(c) L_2 method(colored)



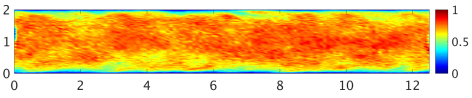
(d) L_2 method(lined)



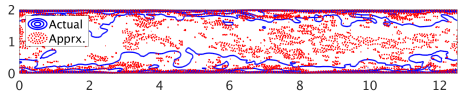
(e) Kriging L_2 method(colored)



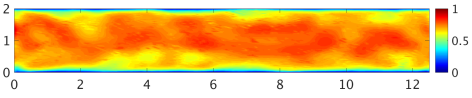
(f) Kriging L_2 method(lined)



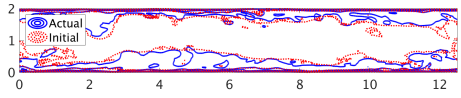
(g) GP Kriging L_2 method(colored)



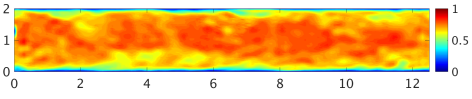
(h) GP Kriging L_2 method(lined)



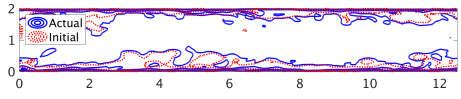
(i) Initial guess from kriging(colored)



(j) Initial guess from kriging(lined)



(k) Initial guess from GP kriging(colored)



(l) Initial guess from GP kriging(lined)

Figure 3.40: The reconstructed field for all the cases with unknown basis for channel flow.

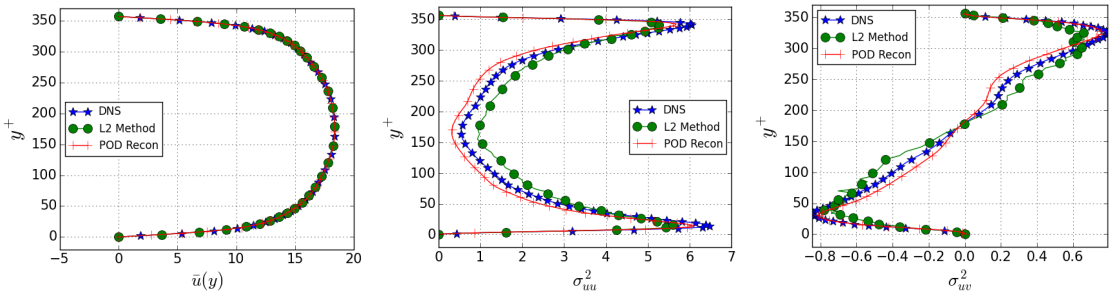


Figure 3.41: The turbulent statistics of reconstruction using L_2 method with unknown basis for channel flow. (Left to right: mean, variance, co-variance)

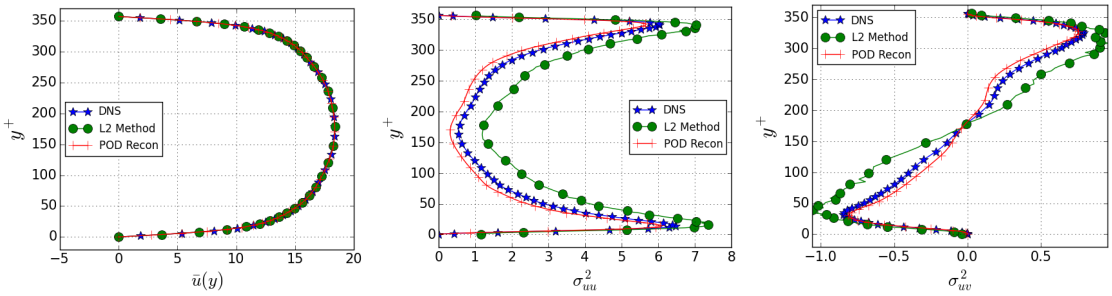


Figure 3.42: The turbulent statistics of reconstruction using Kriging + L_2 method with unknown basis for channel flow. (Left to right: mean, variance, co-variance)

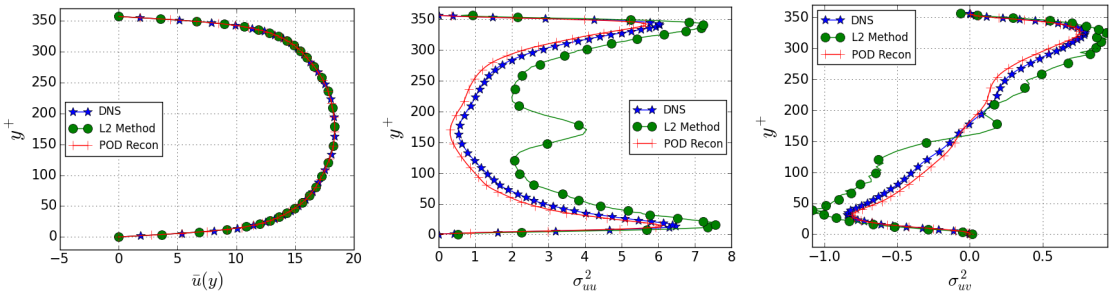


Figure 3.43: The turbulent statistics of reconstruction using GP Kriging + L_2 method with unknown basis for channel flow. (Left to right: mean, variance, co-variance)

Reduced Channel Flow

For the reduced channel flow, a gappiness level is once again chosen to be $p = 0.25$ to obtain sparse measurements from the high-resolution channel flow data. Similar to the analysis with known basis, 300 snapshots of data are used, and 50 modes are retained which should correspond to 95 percent energy capture. The progressive method is also too expensive to perform for this case. Therefore, the L_2 and L_1, kriging with L_2 and L_1, and GP kriging with L_2 and L_1 methods are explored for sparse reconstruction of this data. Kriging is utilized to obtain the initial reconstruction to initialize the iterative procedure.

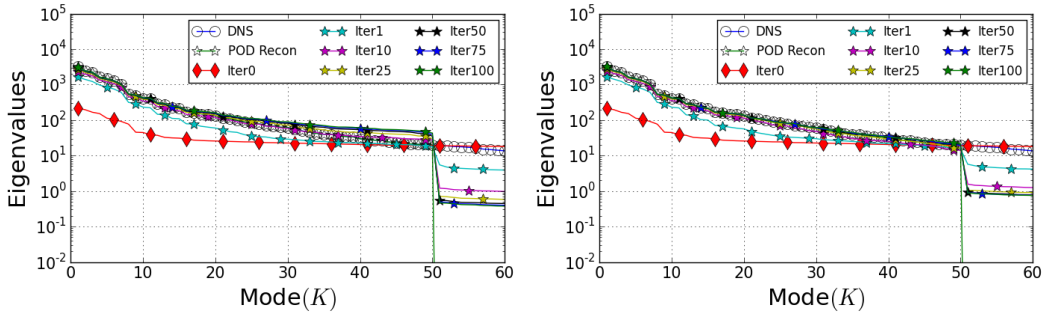
The eigenvalues and Φ_K from the covariance matrix are summarized in Table 3.10. The pure L_2 method captures about 20 eigenvectors and eigenvalues correctly. In comparison, both kriging and GP kriging with L_2 method reconstruct about 25 eigenvectors and eigenvalues. For a moderate-dimensional system such as this, retaining 50 modes(96%) can capture up to 25 modes(96%) with unknown basis. Fifty modes represent 96% and twenty five modes represent nearly 90% of the energy content for this system. In comparison to the full channel flow, the sparse construction performs better for the reduced channel flow because 90% of the energy content is captured. The corresponding eigenvalues and Φ_K are shown in Figure 3.44 and 3.45, respectively.

Table 3.10: The number of eigenvalues and eigenvectors captured with unknown basis for reduced channel flow.

Method	Eigenvector	Eigenvalue
L_2	20	20
Kriging + L_2	25	25
GP kriging + L_2	25	25

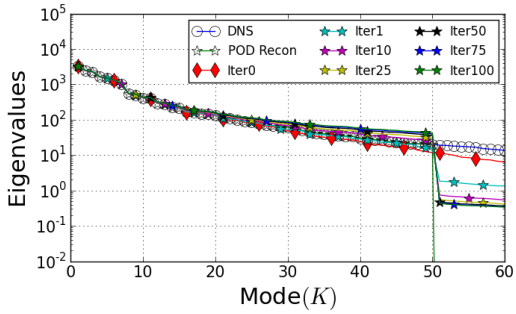
Similar to the channel flow, the iterative procedure does not converge and results in less accurate solution at later iterations as compared with iteration 1 for all the cases with kriging. This indicates that the initial condition outperforms both the L_2 and L_1 methods. For further demonstration, the initial solutions from the kriging methods are shown in Figure 3.47. All the reconstructed solutions capture the shape of large-scale structure correctly while containing some noise. In comparison to the channel flow, the reconstructions are less noisy. Kriging, even in its simpler forms, provide reasonably accurate reconstruction as compared to the actual solution. However, because the low-energy modes cannot be accurately captured, the reconstructions from the iterative procedure introduces noise for all the cases as shown in 3.46. The corresponding turbulent statistics are shown in Figure 3.48, 3.49, 3.50, 3.51, 3.52, and 3.53.

In summary, the L_2 and L_1 methods are capable of reconstructing sparse data accurately with known basis. From this analysis, an optimal rank should be chosen to capture at least 95% of the energy content to ensure accurate reconstructions. Obviously, the 95% is an *ad hoc* choice for the flow systems considered and could depend on the system dynamics. The number of modes retained should be chosen as large as possible to capture the majority of the energy content for the system. However, retaining a higher number of modes does not allow constructing an efficient model. The number of measurements P should be chosen such that it is higher than the maximum of K and $K_{95\%}$. This tends to make such reconstruction methods successful for low-dimensional systems. In most situations, the L_2 method performs better than the L_1. However, L_1 can reconstruct sparse data with fewer measurements for a system that is highly compressible in the basis space, i.e., $p \sim s$. In case of channel flow turbulence where the low energy modes cannot be neglected, L_2 minimization results in better reconstruction for both solution fields and turbulent statistics than the L_1 minimization.

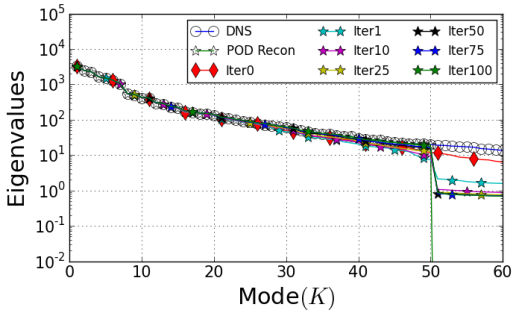


(a) L_2 method

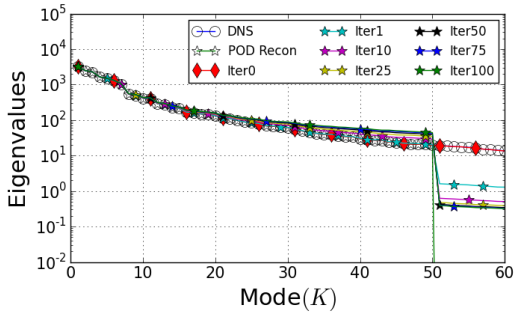
(b) L_2 method



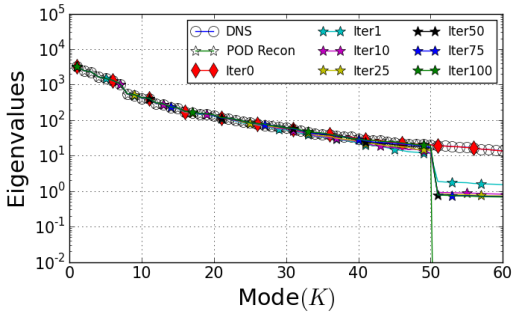
(c) Kriging + L_2 method



(d) Kriging + L_1 method



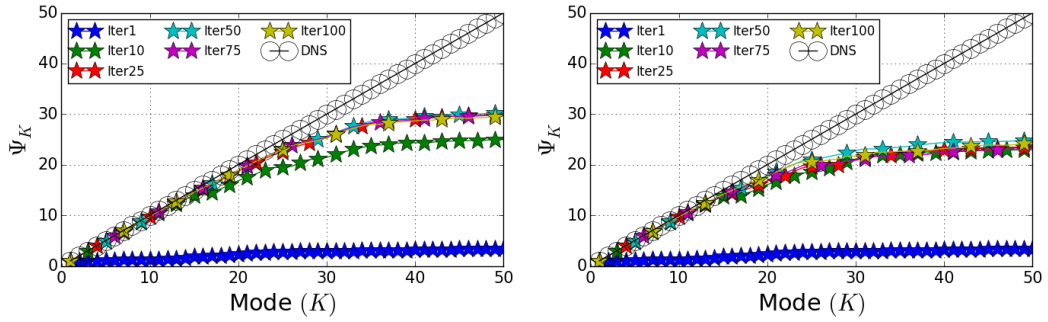
(e) GP Kriging + L_2 method



(f) GP Kriging + L_1 method

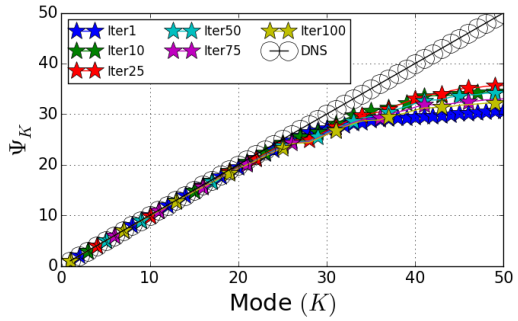
Figure 3.44: The eigenvalues of the reconstructions for all the cases with unknown basis for reduced channel flow.

For sparse reconstruction with known but inexact basis, the results for cylinder flow does not depend on whether the basis functions are learned from the different temporal region. Both MSE and reconstruction quality are consistent with the results from the cases with known basis. However, the reconstruction of the channel flow do not perform as well, and the number of measurements should be chosen such that

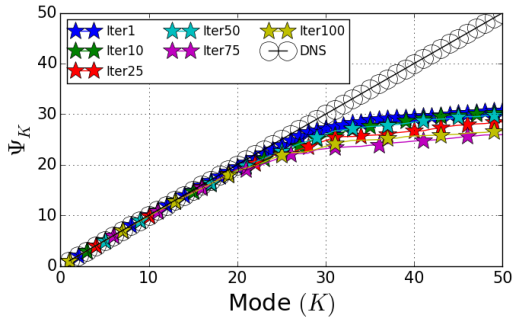


(a) L_2 method

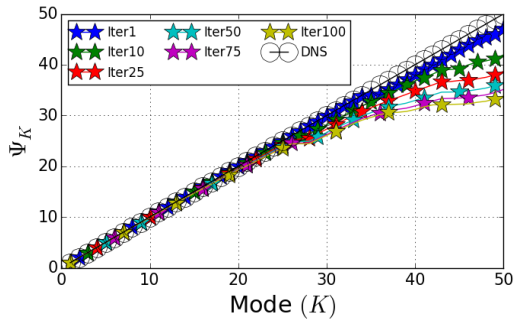
(b) L_1 method



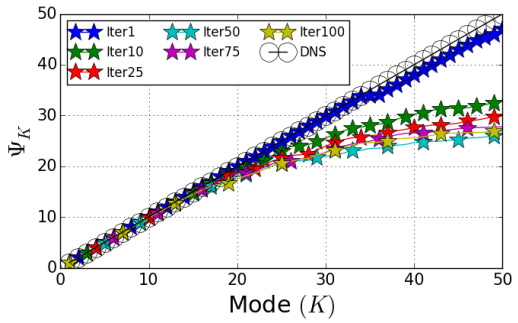
(c) Kriging + L_2 method



(d) Kriging + L_1 method



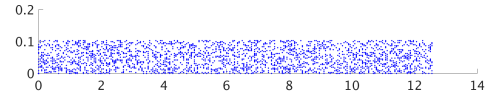
(e) GP Kriging + L_2 method



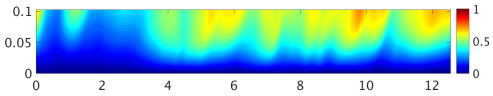
(f) GP Kriging + L_1 method

Figure 3.45: The computed Φ_K of the reconstructions of the reconstructions for all the cases with unknown basis for reduced channel flow.

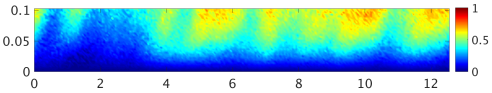
$P > (K, K_{95\%})$. In particular, the errors have increased as compared to the known basis, but the turbulent statistics are reconstructed within a reasonably accurate range. For reduced channel flow, L_2 performs reasonably well, but the L_1 method struggles to repair the solution and capture the turbulent statistics accurately.



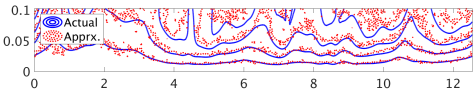
(a) Measured Field



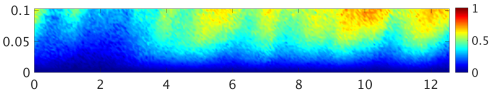
(b) Actual Solution



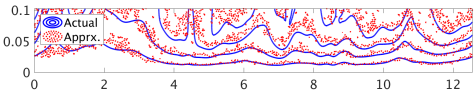
(c) L_2 method(colored)



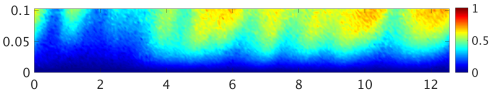
(d) L_2 method(lined)



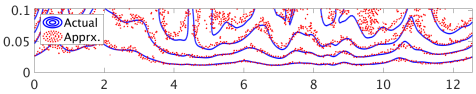
(e) L_1 method(colored)



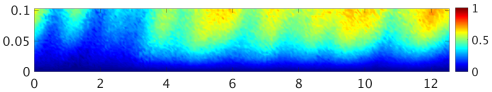
(f) L_1 method(lined)



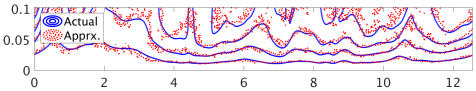
(g) Kriging L_2 method(colored)



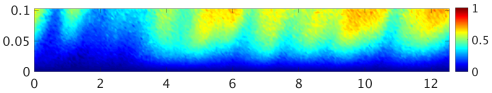
(h) Kriging L_2 method(lined)



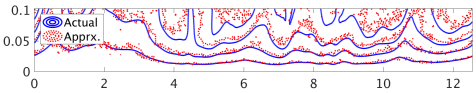
(i) Kriging L_1 method(colored)



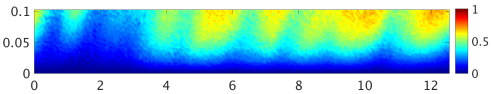
(j) Kriging L_1 method(lined)



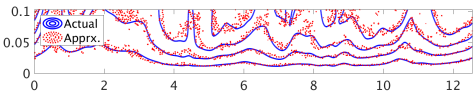
(k) GP Kriging L_2 method(colored)



(l) GP Kriging L_2 method(lined)



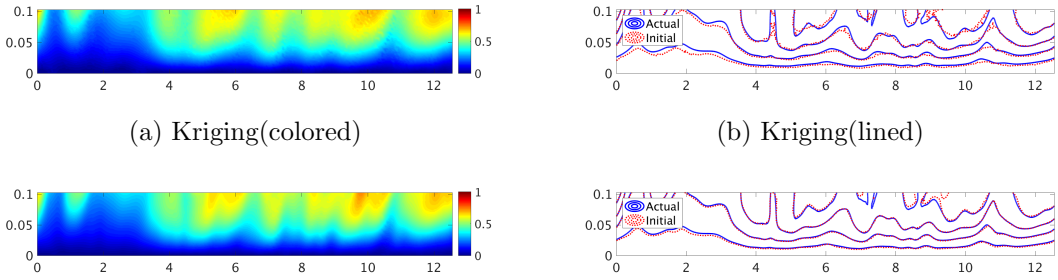
(m) GP Kriging L_1 method(colored)



(n) GP Kriging L_1 method(lined)

Figure 3.46: The reconstructed field for all the cases with unknown basis for reduced channel flow.

When the basis functions are unknown, repairing from sparse data shows promise in obtaining accurate solutions for the low-dimensional system. For example, all the methods perform reasonably well in reconstructing the cylinder flow. However, for a high-dimensional system such as the channel flow, the low-energy modes are diffi-



(a) Kriging(colored) (b) Kriging(lined)

(c) GP kriging(colored) (d) GP kriging(lined)

Figure 3.47: The initial guess from both kriging and GP kriging for reduced channel flow.

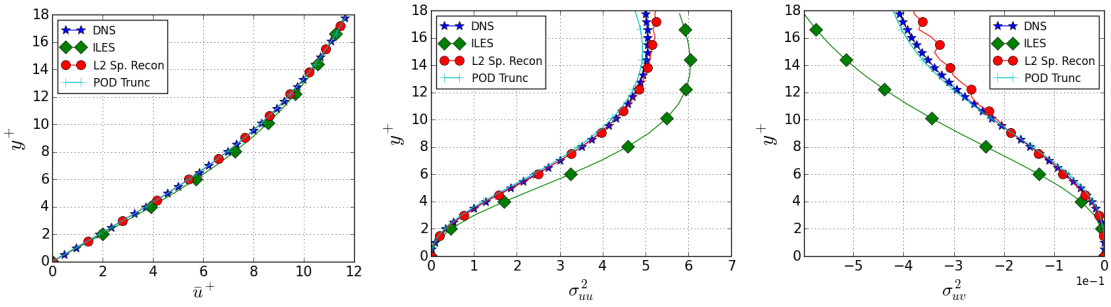


Figure 3.48: The turbulent statistics of reconstruction using L_2 method with unknown basis for reduced channel flow. (Left to right: mean, variance, co-variance)

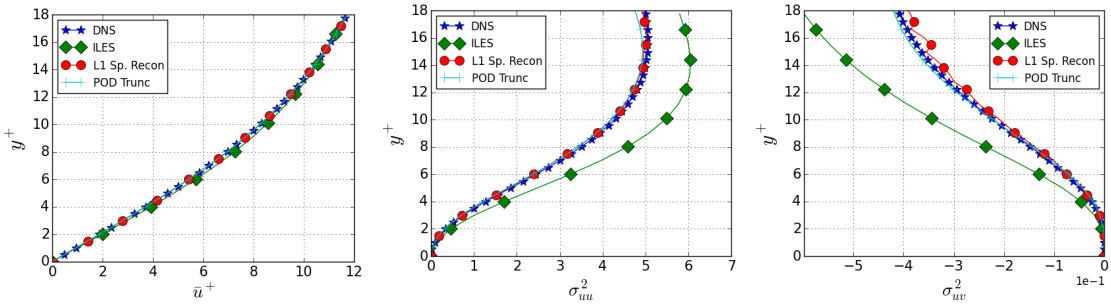


Figure 3.49: The turbulent statistics of reconstruction using L_1 method with unknown basis for reduced channel flow. (Left to right: mean, variance, co-variance)

cult to capture, e.g., only 50% can be captured while potentially retaining sufficient modes to capture 86% of the energy content. Also, noises are introduced during the

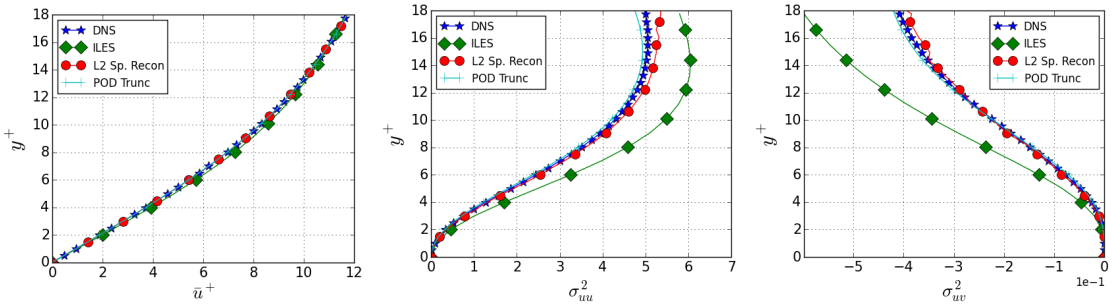


Figure 3.50: The turbulent statistics of reconstruction using Kriging + L_2 method with unknown basis for reduced channel flow. (Left to right: mean, variance, covariance)

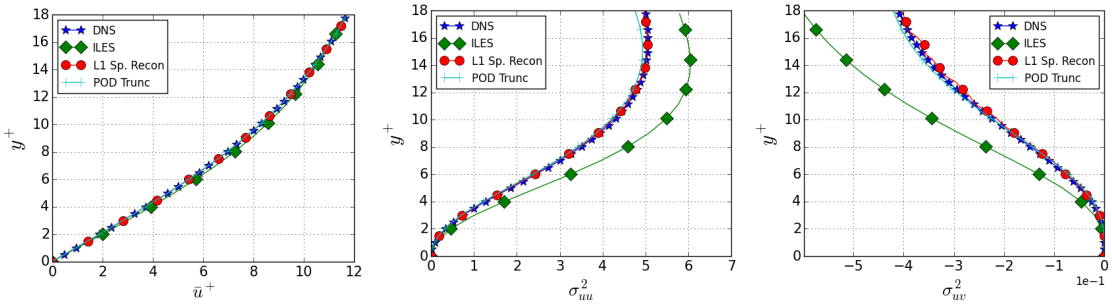


Figure 3.51: The turbulent statistics of reconstruction using Kriging + L_1 method with unknown basis for reduced channel flow. (Left to right: mean, variance, covariance)

iteration procedure. The ability to identify correct eigenvalues and eigenvectors have improved enormously for the moderate-dimensional reduced channel flow, i.e., 90% of the energy can be captured while retaining sufficient modes to capture 95% of the energy content. As before, noisy solutions still tend to appear during the iterative procedure. This indicates that the low-energy modes carry significant energy, which when predicted erroneously can corrupt the reconstruction. A valid hypothesis is that the measurement locations can be optimally chosen to excite low-energy modes. In the real experiment, it could be beneficial for experimentalist if one knows the minimum number of sensors and where to place them. However, the information of

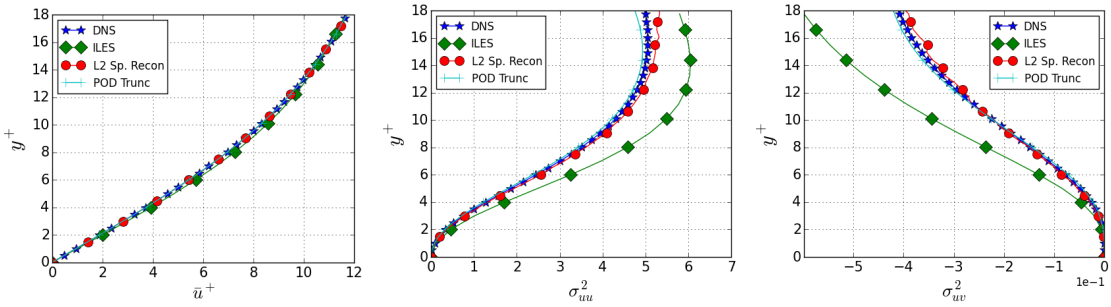


Figure 3.52: The turbulent statistics of reconstruction using GP Kriging + L_2 method with unknown basis for reduced channel flow. (Left to right: mean, variance, covariance)

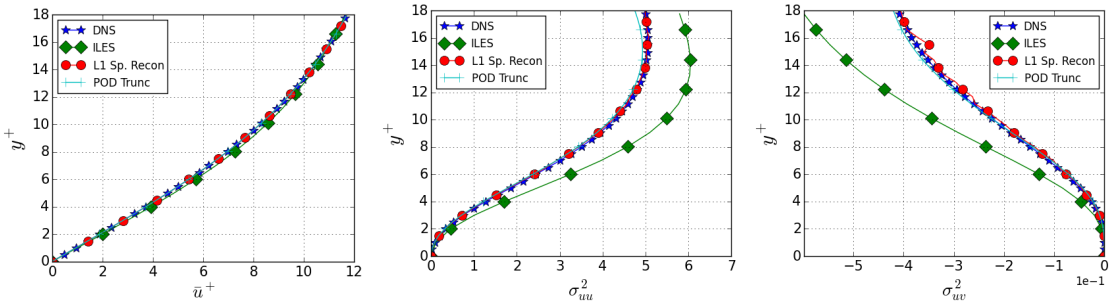


Figure 3.53: The turbulent statistics of reconstruction using GP Kriging + L_1 method with unknown basis for reduced channel flow. (Left to right: mean, variance, covariance)

sensor placement is not always available and requires further investigation.

CHAPTER 4

ONLINE UPDATES FOR DYNAMICALLY EVOLVING CONVOLUTION OPERATORS

4.1 Motivation and Review

In many data-driven problems, the training data is extremely huge or not all available at any given time. The former typically generates a computational limitation while the latter is typical of phenomena that are dynamic with data acquired in real-time. In such cases, building a model in one go is not possible and requires continuous updates as data streams in. For instance, the data-driven models developed in chapter 2 require continuous evolution to incorporate information from the most recent data. In this chapter, methods for dynamically updating the singular value decomposition of the data is explored with the ultimate aim of building dynamic models that incorporate newer data for faster decision making in real-time simulations.

Rebuilding predictive model from scratch can be computationally prohibitive (Peherstorfer & Willcox 2015, 2016), and it is particularly true for higher dimensional fluid flow systems. As an example, the Boussinesq problem presented in chapter 2 requires more than 100,000 degrees of freedom for direct numerical simulation. To analyze three scalar components, i.e., with three times as many grid points (300,000), the required memory for storing all the data snapshots can quickly overload a regular desktop computer. Online modeling with dynamic updates allows for resources with low computational power, memory and storage capacity to be leveraged for building dynamic models as is often the case in onboard flight systems. In Hemati et al. (2014), they utilized the idea of online updates to overcome the memory problem

for large data-set. Practical usage of incremental updates for reducing memory consumption has also been shown in Oxberry et al. (2017), Matsumoto & Indinger (2017). Additionally, online updates methods are advantageous when computing the eigen-decomposition of a system with a large number of snapshots, i.e., $> 10,000$, because the computational effort required for solving such problem is very high. These algorithms, consequently, are critical towards developing effective and efficient predictive models from large data-set.

A second motivation is related to the issue of needing to improve predictive models of dynamically evolving flows. For example, the Boussinesq flow modeled using the sparse convolution method proved to be inaccurate as the convolution operator based on POD modes became outdated as the prediction evolves beyond the training region. In other words, the POD modes used to build the convolution operator is not completely relevant to the dynamical system that it is helping to predict over time. This is consistent with the observations by Peherstorfer & Willcox (2015, 2016) that offline dynamical models may fail to predict the behavior of the system that changes online unless they settle into repeatable limit-cycle behavior. To obtain accurate models, one requires online updates that adapt the model to the evolving system. Specifically, for the Boussinesq flow, it is expected that dynamically updating the convolution operator based on the most recent data can capture the evolving physics better.

4.2 Objective and Contribution

The ultimate objective of this research study is to integrate online update methodology with the sparse convolution framework for developing dynamic data-driven models that can handle sparse data streaming in over time. Dynamic updates enable both model adaption and improved prediction for real-time decision making in a rapidly changing environment. The contribution from this dissertation work is to

develop an online singular value decomposition(SVD) implementation that allows one to update the POD basis incrementally when new data snapshot(s) or prediction(s) are available.

Updating the SVD can be traced back to 1970, Businger (1970) proposed a method that was adapted from QR-decomposition as mentioned by Brand (2006). Bunch & Nielsen (1978) also developed SVD updates when appending a row or a column or deleting a row or a column to the initial matrix through rank-one modification of the symmetric eigenvalue problem. Later, Brand (2006) identified a general identity for additive modification of an SVD and summarized special cases for rank-1 updates, downdates, and revision. Furthermore, Brand (2006) also developed an extended decomposition method that aims to reduce the computational effort. This allows for straightforward procedures for implementing online SVD algorithm as evidenced in Peherstorfer & Willcox (2015), Matsumoto & Indinger (2017), and Oxberry et al. (2017). In this study, the general identity developed from Brand (2006) and a modified rank-1 update algorithm proposed from Oxberry et al. (2017) are adopted and combined to implement a modified rank-K updates procedure, which relaxes the restriction of performing rank-1 updates.

In the next section, the mathematical theory and detailed derivation for a general online SVD identity will be discussed. In section 4.4, the algorithm for online rank-K update is shown. The last section of this chapter will present validation results and assess the computational efficiency.

4.3 Mathematical Theory and Derivation

From Brand (2006), the problem of interest is to compute the updated SVD for a matrix $X \in \mathbb{R}^{N\times M}$ and new information AB^T. The general identity can be expressed as follows.

$$X + AB^T, \tag{4.1}$$

where matrix $A \in \mathbb{R}^{N \times C}$ and matrix $B \in \mathbb{R}^{M \times C}$. Matrix X and AB^T have rank R and C, respectively. If the SVD of X is USV^T, Equation 4.1 can be rearranged:

$$X + AB^T = USV^T + AB^T, \tag{4.2a}$$

$$X + AB^T = \begin{bmatrix} U & A \end{bmatrix} \begin{bmatrix} S & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} V & B \end{bmatrix}^T, \tag{4.2b}$$

where $U \in \mathbb{R}^{N \times R}$, $S \in \mathbb{R}^{R \times R}$, and $V \in \mathbb{R}^{M \times R}$. The right hand side of Equation 4.2b has similar identity from SVD results, but matrices $\begin{bmatrix} U & A \end{bmatrix}$ and $\begin{bmatrix} V & B \end{bmatrix}$ are not orthogonal. To obtain an updated SVD, two orthogonal basis P and Q and two matrices G_A and G_B are defined such that:

$$PG_A = (I - UU^T)A, \tag{4.3a}$$

$$QG_B = (I - VV^T)B. \tag{4.3b}$$

As mentioned in Brand (2006), Equation 4.3 is similar to QR-decomposition, but G_A and G_B need not be upper-triangular or square. By pre-multiplying the transpose of P and Q on Equation 4.3, G_A and G_B can be expressed as:

$$G_A = P^T(I - UU^T)A, \tag{4.4a}$$

$$G_B = Q^T(I - VV^T)B. \tag{4.4b}$$

Then the left bracket from the right side of Equation 4.2b can be rewritten as:

$$\begin{bmatrix} U & A \end{bmatrix} = \begin{bmatrix} U & P \end{bmatrix} \begin{bmatrix} I & U^TA \\ 0 & G_A \end{bmatrix}, \tag{4.5a}$$

$$\begin{bmatrix} U & A \end{bmatrix} = \begin{bmatrix} U(I) + P(0) & UU^TA + PG_A \end{bmatrix}, \tag{4.5b}$$

$$\begin{bmatrix} U & A \end{bmatrix} = \begin{bmatrix} U & UU^TA + (I - UU^T)A \end{bmatrix}, \tag{4.5c}$$

$$\begin{bmatrix} U & A \end{bmatrix} = \begin{bmatrix} U & UU^TA + A - UU^TA \end{bmatrix}, \tag{4.5d}$$

$$\begin{bmatrix} U & A \end{bmatrix} = \begin{bmatrix} U & A \end{bmatrix}. \tag{4.5e}$$

Similarly, the right bracket from the right side of Equation 4.2b can be rewritten as:

$$\begin{bmatrix} V & B \end{bmatrix}^T = \begin{bmatrix} I & V^T B \\ 0 & G_B \end{bmatrix}^T \begin{bmatrix} V & Q \end{bmatrix}^T, \tag{4.6a}$$

$$\begin{bmatrix} V & B \end{bmatrix} = \begin{bmatrix} V & Q \end{bmatrix} \begin{bmatrix} I & V^T B \\ 0 & G_B \end{bmatrix}, \tag{4.6b}$$

$$\begin{bmatrix} V & B \end{bmatrix} = \begin{bmatrix} V(I)Q(0) & VV^T B + QG_B \end{bmatrix}, \tag{4.6c}$$

$$\begin{bmatrix} V & B \end{bmatrix} = \begin{bmatrix} V & VV^T B + (I - VV^T)B \end{bmatrix}, \tag{4.6d}$$

$$\begin{bmatrix} V & B \end{bmatrix} = \begin{bmatrix} V & VV^T B + B - VV^T B \end{bmatrix}, \tag{4.6e}$$

$$\begin{bmatrix} V & B \end{bmatrix} = \begin{bmatrix} V & B \end{bmatrix}. \tag{4.6f}$$

Equation 4.5a and 4.6a can be substituted into Equation 4.2b as:

$$X + AB^T = \begin{bmatrix} U & P \end{bmatrix} \begin{bmatrix} I & U^T A \\ 0 & G_A \end{bmatrix} \begin{bmatrix} S & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & V^T B \\ 0 & G_B \end{bmatrix}^T \begin{bmatrix} V & Q \end{bmatrix}^T. \tag{4.7}$$

From the right hand side of Equation 4.7, the outer matrices are both orthogonal, so the remaining step is to compute SVD of the inner matrices. Let Z be defined as shown in Equation 4.8, and the associated SVD for Z is shown in Equation 4.9.

$$Z = \begin{bmatrix} I & U^T A \\ 0 & G_A \end{bmatrix} \begin{bmatrix} S & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I & V^T B \\ 0 & G_B \end{bmatrix}^T, \tag{4.8a}$$

$$Z = \begin{bmatrix} S & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} U^T A \\ G_A \end{bmatrix} \begin{bmatrix} V^T B \\ G_B \end{bmatrix}^T, \tag{4.8b}$$

$$Z = U' S' (V')^T. \tag{4.9}$$

The updated SVD for Equation 4.1 now can be written as:

$$X + AB^T = \begin{bmatrix} U & P \end{bmatrix} (U')S'(V')^T \begin{bmatrix} V & Q \end{bmatrix}^T. \tag{4.10}$$

Equation 4.10 can be simplified as:

$$X + AB^T = U_{new} S_{new} V_{new}^T, \tag{4.11}$$

where $U_{new} = \begin{bmatrix} U & P \end{bmatrix} (U')$, $S_{new} = S'$, and $V_{new}^T = (V')^T \begin{bmatrix} V & Q \end{bmatrix}^T$. Important to note, the computational expense for SVD in Equation 4.9 depends on the size of $R + C$ or less. If the SVD of Equation 4.1 is computed from the regular batch SVD, the cost would be snapshot dependent. This implies that online SVD is advantageous for a low-rank system, i.e., the snapshot number is much greater than the rank of the system.

In their implementations, Peherstorfer & Willcox (2015), Matsumoto & Indinger (2017), and Oxberry et al. (2017) have utilized rank-1 updates when a new snapshot of data is available. However, the rank-1 updates can be modified into rank-K updates. This modification allows relaxing the updating procedure when K snapshots of data are received. To this extent, the rank-K updates algorithm is presented as a modified version of rank-1 updates in the following section.

4.4 Modified Rank-K Updates Algorithm

For the modified rank-K updates algorithm, the derivation for matrix Z in Equation 4.8 is adopted from previous section. The key is to compute the matrices $U^T A$ and G_A, i.e., Equation 4.8 can be simplified as:

$$Z = \begin{bmatrix} S & U^T A \\ 0 & G_A \end{bmatrix}, \tag{4.12}$$

where Z is a sparse matrix, and its SVD is relatively easy to solve. The updating procedures can be summarized in Algorithm 14 and 15. Important to note, a modified

Gram-Schmidt orthogonalization is applied to compute P in rank-K updates which indirectly increases the computational expense. In a typical rank-1 updates (Oxberry et al. 2017), this effort can be avoided because only one snapshot is updated at a time. In particular, Matrix P becomes a column vector, and matrix G_A is simplified as a scalar. For brevity, the detailed algorithm of the rank-1 updates is not shown here, and research articles such as Peherstorfer & Willcox (2015), Matsumoto & Indinger (2017), and Oxberry et al. (2017) are strongly recommended for readers who are interested in its implementation. Regardless of rank-1 or rank-K updates, both approaches require applying modified Gram-Schmidt orthogonalization at the end to ensure that the orthogonality of left singular vectors U_{new}. This step is also suggested in Brand (2002) and Oxberry et al. (2017) for numerical robustness. Nonetheless, the online rank-K updates proposed in this study is rank-dependent and more generalized as compared to rank-1 updates.

Algorithm 14: Online Rank-k SVD updates part I

input : Left singular vector: $U \in \mathbb{R}^{N \times R}$

Singular values: $S \in \mathbb{R}^{R \times R}$

Left singular vector: $V \in \mathbb{R}^{R \times M}$

Current rank of the inputs: R

Maximum cut off rank: R_{max}

New data snapshots: a

Desired rank for updates: K

output: Updated snapshot number: M_{new}

Updated rank: R_{new}

Updated left singular vector: $U_{new} \in \mathbb{R}^{N \times R_{new}}$

Singular values: $S_{new} \in \mathbb{R}^{R_{new} \times R_{new}}$

Left singular vector: $V_{new} \in \mathbb{R}^{R_{new} \times M_{new}}$

1 **Begin**

2 Compute matrix $g = (I - UU^T)a$ and apply modified Gram-Schmidt algorithm to compute orthogonal basis P from matrix g.

3 Compute matrix $G_a = P^T g$.

4 Construct Matrix $Z = \begin{bmatrix} S & P^T g \\ 0 & R_a \end{bmatrix}$.

5 Compute the SVD: $Z = U' S' (V')^T$.

Algorithm 15: Online Rank-k SVD updates part II

if $R >= R_{max}$ **then**

  compute $U_{new} = UU'_{:,1:R}$, $S_{new} = S'_{1:R,1:R}$, and $V_{new} = \begin{bmatrix} V & 0 \\ 0 & 1 \end{bmatrix} V'_{:,1:R}$

else

  compute $U_{new} = \begin{bmatrix} U & P \end{bmatrix} U'$, $S_{new} = S'$, and $V_{new} = \begin{bmatrix} V & 0 \\ 0 & 1 \end{bmatrix} V'$

  $R_{new} = R + K$

Perform modified Gram-Schmidt algorithm to ensure the orthogonality of left singular vectors U_{new}.

4.5 Results and Discussion

In this section, the online rank-K SVD updates are performed as compared to the online-batch-1 and online-rank-1 updates to illustrate its accuracy and efficiency. Online-batch-1 updates recompute SVD from scratch whenever a new data snapshot is received. Online-rank-1 updates are essentially the simplified version of the algorithm as against to the rank-K updates. The details of online-rank-1 updates can be referred to section 4.4. For comparison purposes, K is chosen as 5, 20, 50 and 100, and 500 snapshots of cylinder flow data at $Re = 100$ are utilized to perform online updates. All the cases are summarized in Table 4.1.

First, the L^2 error norms of the reconstructed solution for each updating procedure are defined in Equation 4.13 to quantify the performance and accuracy of the rank-K SVD updates.

$$\phi = \sqrt{\frac{1}{M}\frac{1}{N}\sum_{i=1}^{M}\sum_{j=1}^{N}(\bar{x}_{i,j}^{Recon} - \bar{x}_{i,j}^{True})^2} \quad . \tag{4.13}$$

The corresponding error plot for each case is shown in Figure 4.1. In comparison, recomputing the SVD from scratch(online batch-1) is around order of magnitude of

10^{-9}. As the test data is single precision, obtaining an order of magnitude of 10^{-8} for the L^2 error norms is considered to be fairly accurate. More importantly, the errors tend to stabilize as the number of new snapshot reaches 200 for all the cases. This observation proves that the modified rank-K algorithm is performing properly. The numerical values are also summarized in Table 4.1.

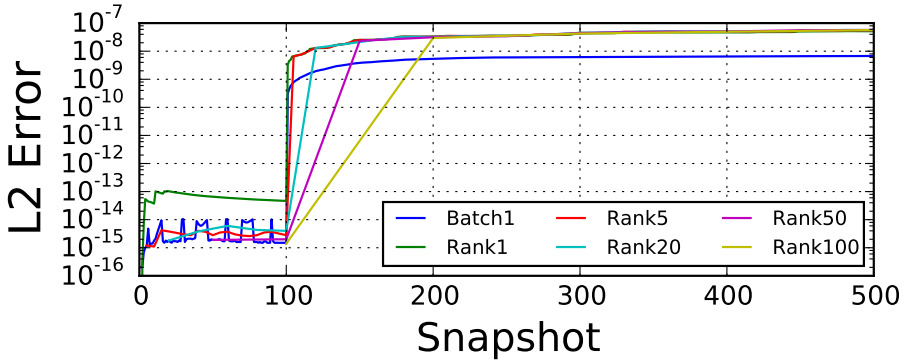


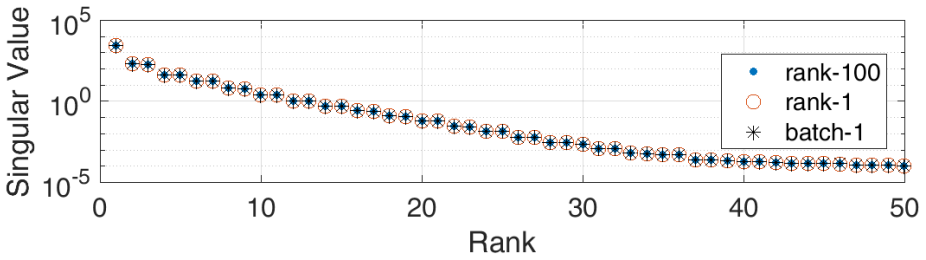
Figure 4.1: L^2 error norms for online SVD updates

Table 4.1: The L^2 error norms of the reconstruction for online SVD updates

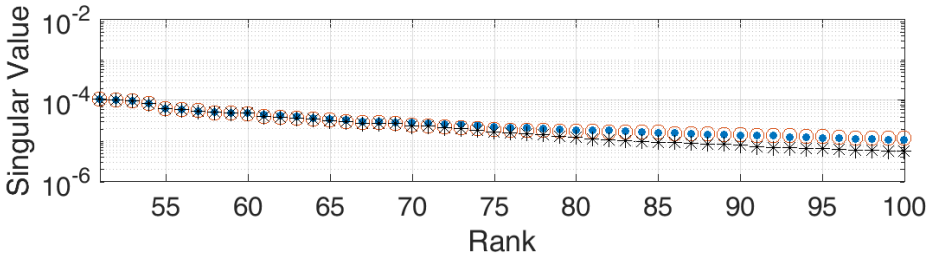
Cases	Cputime(s)	Ratio β	L^2 error norms
Rank-1	924.98	1.00	5.1383E-8
Rank-5	211.69	0.87	5.5182E-8
Rank-20	61.59	0.75	5.6402E-8
Rank-50	30.93	0.60	5.7050E-8
Rank-100	22.04	0.42	5.6711E-8
Batch-1	742.74	-	6.7659E-9

The updated singular values for rank-100 updates are also shown in Figure 4.2. The updated singular values for both rank-1 and rank-100 updates appear to match reasonably well. In comparison to the results from online batch-1 updates, only the low singular values appear to deviate. However, this small difference will not affect the reconstruction errors because the maximum singular value is large as compared to

the small ones. Due to the fact that the updating procedure may erode the orthog-



(a) L_2 Minimization



(b) L_1 Minimization

Figure 4.2: The updated singular values for online rank-100 updates.

onality of the left singular vectors U_{new} (Brand 2006), an orthogonality number ψ is computed and examined using Equation 4.14.

$$\psi = \sum_{i=1}^{R_{new}} (U_{new}(:, i) \cdot U_{new}(:, i)). \tag{4.14}$$

The corresponding ψ for online rank-1, rank-5, and rank-100 updates is also shown in Figure 4.3. All of three cases are preserving orthogonality for the left singular vector, and maximum of ψ is approaching to the number of rank retained which is expected.

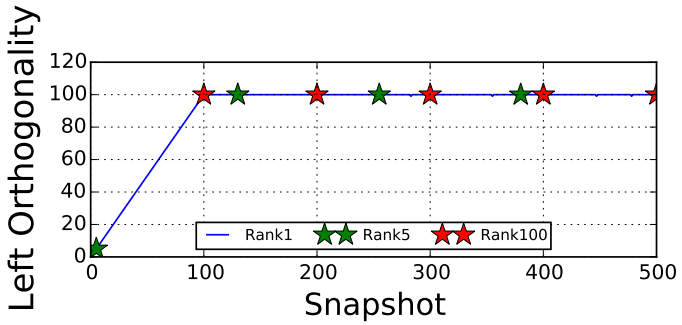


Figure 4.3: Orthogonality number ψ for online SVD updates

After examining the accuracy of the implementations, the efficiency of the online rank-K updates is investigated. The CPU time in seconds for each update process is computed and shown for all the cases in Figure 4.4.

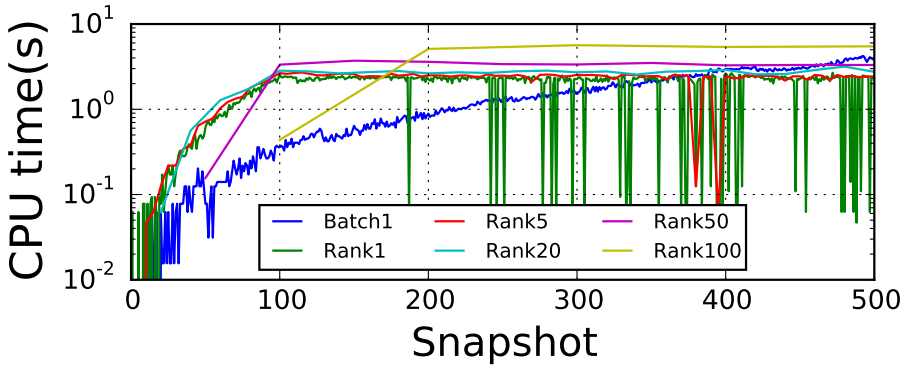


Figure 4.4: CPU time for each updating process.

The required computational time is reasonably stable for rank-K updates. Over a certain number of snapshots, e.g., 500 snapshots, re-computing SVD from scratch becomes quite expensive. In time-critical applications, one cannot afford to rebuild the models from scratch. This implies that the online rank-K updates are more efficient when encountering a large data-set. Figure 4.5 shows the total CPU time with the associated rank. As the number K is increased, the required total CPU time reduces accordingly. An advantage for rank-K updates is that one can afford to perform the dynamical updates with the desired number of snapshots, i.e., number of ranks. Performing rank-1 updates may not be the most efficient way if a large number of data snapshots are received. Instead, the rank-K updates can be performed to save computational effort as shown in Figure 4.5. However, the rank number K must be chosen wisely so that the dynamical models can capture the changing dynamics while maintaining computational efficiency. For Boussinesq flow, the convolutions or POD basis change completely when the training data are moved from 1-400 snapshots to 1-800 snapshots. Obviously, rank-400 updates would not work for this flow. Hence, the strategy for identifying an optimal choice of rank number for online modeling

is not clear, and this choice needs to balance the evolution of the physics with the computational efficiency of low-rank updates.

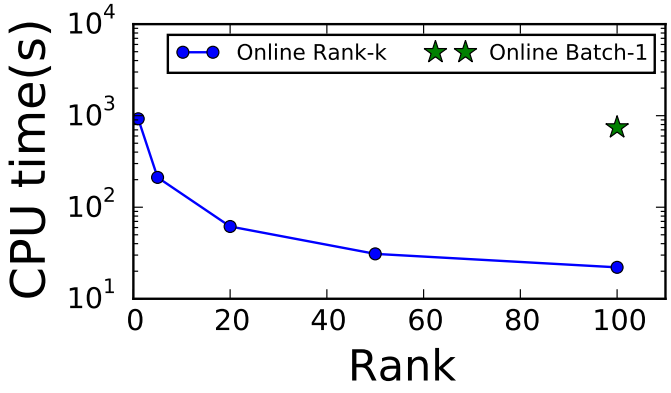


Figure 4.5: Total CPU time for online SVD updates

The relative ratio β of CPU time for the online rank-K updates is computed in Equation 4.15. to establish the efficiency as against to the rank-1 updates.

$$\beta = \frac{\text{CPU Time}_{\text{rank}-K}}{\text{CPU Time}_{\text{rank}-1}} \cdot \frac{1}{K} \tag{4.15}$$

The corresponding results are shown in Table 4.1. The relative efficiency reduces as K increases. Ideally, a ratio of 1 is desired regarding memory consumption and time-saving. As an additional Gram-Schmidt orthogonalization process is needed for rank-K updates, one cannot achieve the ideal efficiency. However, the total time saved from this approach is still attractive if the memory storage permits. Thus, the online rank-K updates presented in this study are more favorable than the regular batch computation.

In conclusion, a capability for incremental SVD updates is developed in this work. The online rank-K update procedure is implemented as a generalization to the rank-1 updates that are reported in Peherstorfer & Willcox (2015), Matsumoto & Indinger (2017), Oxberry et al. (2017). While providing dynamic system modeling, another advantage of online updates is to bypass the need for storing all the data. This is an essential component of fast computation and decision making. The errors from the

reconstruction (using the POD modes) for all the cases are computed and found to be accurate for all the cases considered. The singular values from online rank-1 and rank-K updates are also correctly computed from the dynamic update. Further, the orthogonality number for the left singular vectors is computed, and shown to match the maximum rank retained for all the cases. Althought the relative efficiency reduces as K increases, the total time saved for online rank-K updates is still remarkable as compared to online batch-1 and rank-1 updates. To conclude, the resulting rank-K updates are correctly implemented as compared to the online batch-1 updates. This method can be integrated into sparse convolution framework for modeling systems that evolves in time.

CHAPTER 5

CONCLUSIONS AND FUTURE RECOMMENDATION

5.1 Key Conclusion

In chapter 2, a sparse generalized convolution framework was presented for modeling and predicting fluid flow systems based on the Koopman operator theory. This framework employs the ideas of single/multi-layer convolution and sparse representation for accurate and efficient model prediction. The effectiveness of the framework comes from the reliance on the type of convolution and sparsification techniques rather than learning nonlinear operators. To demonstrate the connection with the Koopman operator, the leading eigenvalues and eigenmodes are accurately captured for flow past a cylinder at $Re = 100$ and $Re = 1000$. For model prediction, the convolution framework is capable of capturing the underlying dynamics for both limit-cycle and systems that approach a limit-cycle, i.e., the periodic and transient cylinder flows. However, the physics of highly convective problems such as Boussinesq flow are not predicted accurately due to the fast-evolving nature of the system.

In chapter 3, sparse reconstruction techniques including L_2-based, L_1-based, and Kriging methods are investigated to recover sparse fluid flow measurements with the known basis, known but inexact basis, and the unknown basis for flow past a cylinder, full channel flow, and reduced channel flow. The number of measurements that are sufficient to recover sparse data with known basis is related to the number of ranks retained, i.e., $P > \max(K, K_{95\%})$. For unknown basis, the performance of reconstruction techniques is evaluated to capture the eigenvalues and eigenvectors of the system. The cylinder flow can be reconstructed successfully with just 25%

of the data. However, the reconstructed solutions for both channel and reduced channel flows are less than ideal because the low-energy modes cannot be accurately approximated. As a result, the reconstructions are corrupted by noise. Thus, further improvements to the algorithms will be required.

In chapter 4, the ideas of online SVD updates that can potentially enable dynamical convolution updates are explored. The online rank-K updates are implemented in this work, and the validated results such as reconstructed error, orthogonality number for left singular vectors, and singular values are shown for flow past a cylinder. Furthermore, the computational efficiency of online rank-K updates appears to be justified as compared to online batch-1 and rank-1 updates.

5.2 Future Work and Recommendation

The primary scope of this work is to investigate three major components: developing predictive models, sparse reconstruction, and online updates. Hence, the ultimate recommendation is to combine all of these three components for developing dynamic data-driven models that are capable of handling sparse non-linear fluid flow data. However, a list of potential problems in each research direction can be pursued for future advancement as outlined.

5.2.1 Sparse Convolution Model

1. A comprehensive parametric study should be performed for sensitivity evaluation when sparsifying the convolution. For instance, the truncation method based on the energy content of the POD mode is utilized for POD-convolution. The ideas of using user-defined constraints (Chen et al. 2012, Wynn et al. 2013, Jovanović et al. 2014) for selecting optimal basis are strongly recommended to explore how well they could improve the performance. Similarly, for GP-convolution, other center placement techniques such as (Cohen et al. 2003,

Willcox 2006, Yildirim et al. 2009) should also be investigated and compared whether optimal center locations can be identified for more accurate and efficient prediction. The impact of hyperparameter σ in GP functions can be explored in conjunction with determining optimal centers for GP-convolutions.

2. When using GP-convolution for modeling transient cylinder flow, there are unanswered questions such as: How does the linear GP model capture the transient behavior of the cylinder flow? What is the source of the nonlinearity in the model? The GP-convolution also appears to be more stable than POD-convolution for modeling transient cylinder flow, and the reason has not been understood yet. The requires further investigation. Both of above questions are critical to ascertain the capability of GP-convolution and need to be addressed.

3. Identifying the regularization(Tikhonov) parameter for obtaining accurate and stable model predictions is rather ad-hoc, which makes the modeling framework less self-consistent. It would be useful to investigate deterministic approaches to regularize the model from data itself.

4. Another recommendation is tied to the fact that the developed convolution framework has failed to predict Boussinesq flow beyond the training region. In particular, the convolution is outdated, and this is the main reason why such systems are challenging to model. Hence, a potential solution is to perform dynamical convolution updates whenever a new prediction is available. This involves using online singular value decomposition for computing POD basis. To this end, the use of online updates for modeling non-linear system such as Bousineseq flow is highly recommended. The implementation of online rank-K updates is validated and can be directly integrated into the convolution framework.

5.2.2 Sparse Reconstruction

1. With unknown basis, the number of measurements for both channel and reduced channel flows was chosen arbitrarily(25% of the data). This makes such methods less attractive for practical use. Therefore, the minimum number of measurements for these two flow systems needs to be identified.

2. The iterative procedure for reconstructing both channel and reduced channel flows appears to corrupt the solutions by introducing noise. This is because the low-energy modes cannot be accurately approximated. The next logical step is to de-noise the solution or improve the methods to prevent the generation of noise. One possible solution is to identify optimal sensor placement that may improve the ability to capture low energy modes.

3. Following the previous recommendation, random measurements are used for all the reconstruction analysis. It would be valuable to investigate the influence of sensor placement on the performance of data recovery. Determining the locations using methods such as Cohen et al. (2003), Willcox (2006) and coarse-graining is worth exploring in the future.

5.2.3 Online SVD Updates

1. The implementation of online rank-K updates from this work is validated for accuracy and stability. The test problem, i.e., flow past a cylinder, results in a matrix with a row dimension of $24,000$. However, the study of computational cost for matrices with different levels of dimensions has not been performed yet. The orthonormalization processes could be too expensive for a taller matrix, e.g., Boussinesq flow. As such, the first suggestion is to examine the performance of online rank-K updates for problems that contain a higher degree of freedom. If the computation effort is too expensive, an extended decomposition method

that was proposed by Brand (2006) is highly recommended to improve the speed of the updating processes.

2. For Boussinesq flow, the strategy for identifying an optimal choice of rank number for online modeling is not clear, and this choice needs to balance the evolution of the physics with the computational efficiency of low-rank updates. Hence, another recommendation is to determine an optimal K for updating Boussinesq flow accurately and efficiently.

REFERENCES

Absil, P.-A., Mahony, R. & Sepulchre, R. (2009), *Optimization algorithms on matrix manifolds*, Princeton University Press.

Adrian, R. J. (2005), 'Twenty years of particle image velocimetry', *Experiments in fluids* **39**(2), 159–169.

Akhtar, I., Wang, Z., Borggaard, J. & Iliescu, T. (2012), 'A new closure strategy for proper orthogonal decomposition reduced-order models', *Journal of Computational and Nonlinear Dynamics* **7**(3), 034503.

Bai, Z., Brunton, S. L., Brunton, B. W., Kutz, J. N., Kaiser, E., Spohn, A. & Noack, B. R. (2017), Data-driven methods in fluid dynamics: Sparse classification from experimental data, *in* 'Whither Turbulence and Big Data in the 21st Century?', Springer, pp. 323–342.

Bai, Z., Wimalajeewa, T., Berger, Z., Wang, G., Glauser, M. & Varshney, P. K. (2014), 'Low-dimensional approach for reconstruction of airfoil data via compressive sensing', *AIAA Journal* .

Bishop, C. (2007), 'Pattern recognition and machine learning (information science and statistics), 1st edn. 2006. corr. 2nd printing edn', *Springer, New York* .

Blocken, B., Stathopoulos, T. & Carmeliet, J. (2007), 'Cfd simulation of the atmospheric boundary layer: wall function problems', *Atmospheric environment* **41**(2), 238–252.

Boyd, S., Parikh, N., Chu, E., Peleato, B. & Eckstein, J. (2011), 'Distributed optimization and statistical learning via the alternating direction method of multipliers', *Foundations and Trends® in Machine Learning* **3**(1), 1–122.

Boyd, S. & Vandenberghe, L. (2004), *Convex optimization*, Cambridge university press.

Brand, M. (2002), 'Incremental singular value decomposition of uncertain data with missing values', *Computer VisionECCV 2002* pp. 707–720.

Brand, M. (2006), 'Fast low-rank modifications of the thin singular value decomposition', *Linear algebra and its applications* **415**(1), 20–30.

Bright, I., Lin, G. & Kutz, J. N. (2013), 'Compressive sensing based machine learning strategy for characterizing the flow around a cylinder with limited pressure measurements', *Physics of Fluids* **25**(12), 127102.

Brunton, S. L., Proctor, J. L. & Kutz, J. N. (2013), 'Compressive sampling and dynamic mode decomposition', *arXiv preprint arXiv:1312.5186* .

Brunton, S. L., Proctor, J. L. & Kutz, J. N. (2016), 'Discovering governing equations from data by sparse identification of nonlinear dynamical systems', *Proceedings of the National Academy of Sciences* **113**(15), 3932–3937.

Brunton, S. L., Tu, J. H., Bright, I. & Kutz, J. N. (2014), 'Compressive sensing and low-rank libraries for classification of bifurcation regimes in nonlinear dynamical systems', *SIAM Journal on Applied Dynamical Systems* **13**(4), 1716–1732.

Bui-Thanh, T., Damodaran, M. & Willcox, K. (2004), 'Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition', *AIAA journal* **42**(8), 1505–1516.

Bunch, J. R. & Nielsen, C. P. (1978), 'Updating the singular value decomposition', *Numerische Mathematik* **31**(2), 111–129.

Businger, P. (1970), 'Updating a singular value decomposition', *BIT* **10**(3), 376–385.

Candès, E. J. & Wakin, M. B. (2008), 'An introduction to compressive sampling', *IEEE signal processing magazine* **25**(2), 21–30.

Candès, E. J. et al. (2006), Compressive sampling, *in* 'Proceedings of the international congress of mathematicians', Vol. 3, Madrid, Spain, pp. 1433–1452.

Cantwell, C. D., Moxey, D., Comerford, A., Bolis, A., Rocco, G., Mengaldo, G., De Grazia, D., Yakovlev, S., Lombard, J.-E., Ekelschot, D. et al. (2015), 'Nektar++: An open-source spectral/hp element framework', *Computer Physics Communications* **192**, 205–219.

Chen, K. K., Tu, J. H. & Rowley, C. W. (2012), 'Variants of dynamic mode decomposition: boundary condition, koopman, and fourier analyses', *Journal of nonlinear science* **22**(6), 887–915.

Cohen, K., Siegel, S. & McLaughlin, T. (2003), Sensor placement based on proper orthogonal decomposition modeling of a cylinder wake, *in* '33rd AIAA Fluid Dynamics Conference, Orlando, AIAA', Vol. 4259, p. 2003.

Csató, L. & Opper, M. (2002), 'Sparse on-line gaussian processes', *Neural computation* **14**(3), 641–668.

Dandou, A., Tombrou, M., Kalogiros, J., Bossioli, E., Biskos, G., Mihalopoulos, N. & Coe, H. (2017), 'Investigation of turbulence parametrization schemes with reference to the atmospheric boundary layer over the aegean sea during etesian winds', *Boundary-Layer Meteorology* pp. 1–27.

Daubechies, I., Defrise, M. & De Mol, C. (2004), 'An iterative thresholding algorithm for linear inverse problems with a sparsity constraint', *Communications on pure and applied mathematics* **57**(11), 1413–1457.

de Baar, J. H., Percin, M., Dwight, R. P., van Oudheusden, B. W. & Bijl, H. (2014), 'Kriging regression of piv data using a local error estimate', *Experiments in fluids* **55**(1), 1650.

Deutsch, C. V. & Journel, A. G. (1998), 'Geostatistical software library and users guide', *Oxford University Press, New York* .

Du, Y., Symeonidis, V. & Karniadakis, G. (2002), 'Drag reduction in wall-bounded turbulence via a transverse travelling wave', *Journal of fluid mechanics* **457**, 1–34.

Edelman, A., Arias, T. A. & Smith, S. T. (1998), 'The geometry of algorithms with orthogonality constraints', *SIAM journal on Matrix Analysis and Applications* **20**(2), 303–353.

Everson, R. & Sirovich, L. (1995), 'Karhunen–loeve procedure for gappy data', *JOSA A* **12**(8), 1657–1664.

Evolving Gaussian Processes and Kernel Observers (n.d.).
URL: *http://daslab.illinois.edu/software.html*

Gilbert, A. C., Guha, S., Indyk, P., Muthukrishnan, S. & Strauss, M. (2002), Near-optimal sparse fourier representations via sampling, *in* 'Proceedings of the thiry-fourth annual ACM symposium on Theory of computing', ACM, pp. 152–161.

Gilbert, A. C., Strauss, M. J., Tropp, J. A. & Vershynin, R. (2007), One sketch for all: fast algorithms for compressed sensing, *in* 'Proceedings of the thirty-ninth annual ACM symposium on Theory of computing', ACM, pp. 237–246.

Goulart, P. J., Wynn, A. & Pearson, D. (2012), Optimal mode decomposition for high dimensional systems, *in* 'Decision and Control (CDC), 2012 IEEE 51st Annual Conference on', IEEE, pp. 4965–4970.

Grosek, J. & Kutz, J. N. (2014), 'Dynamic mode decomposition for real-time background/foreground separation in video', *arXiv preprint arXiv:1404.7592* .

Gunes, H. & Rist, U. (2008), 'On the use of kriging for enhanced data reconstruction in a separated transitional flat-plate boundary layer', *Physics of Fluids* **20**(10), 104109.

Gunes, H., Sirisup, S. & Karniadakis, G. E. (2006), 'Gappy data: To krig or not to krig?', *Journal of Computational Physics* **212**(1), 358–382.

Hemati, M. S., Williams, M. O. & Rowley, C. W. (2014), 'Dynamic mode decomposition for large and streaming datasets', *Physics of Fluids* **26**(11), 111701.

Holmes, P. (2012), *Turbulence, coherent structures, dynamical systems and symmetry*, Cambridge university press.

Jordan, S. K. & Fromm, J. E. (1972), 'Oscillatory drag, lift, and torque on a circular cylinder in a uniform flow', *The Physics of Fluids* **15**(3), 371–376.

Jovanović, M. R., Schmid, P. J. & Nichols, J. W. (2014), 'Sparsity-promoting dynamic mode decomposition', *Physics of Fluids* **26**(2), 024103.

Kim, J., Moin, P. & Moser, R. (1987), 'Turbulence statistics in fully developed channel flow at low reynolds number', *Journal of fluid mechanics* **177**, 133–166.

Kim, M., Lee, J. H. & Park, H. (2016), 'Study of bubble-induced turbulence in upward laminar bubbly pipe flows measured with a two-phase particle image velocimetry', *Experiments in Fluids* **57**(4), 55.

Kingravi, H. A., Maske, H. & Chowdhary, G. (2015), Kernel controllers: A systems-theoretic approach for data-driven modeling and control of spatiotemporally evolving processes, *in* 'Decision and Control (CDC), 2015 IEEE 54th Annual Conference on', IEEE, pp. 7365–7370.

Kingravi, H. A., Maske, H. R. & Chowdhary, G. (2016), Kernel observers: Systems-theoretic modeling and inference of spatiotemporally evolving processes, *in* 'Advances in Neural Information Processing Systems', pp. 3990–3998.

Koopman, B. O. (1931), 'Hamiltonian systems and transformation in hilbert space', *Proceedings of the National Academy of Sciences* **17**(5), 315–318.

Kramer, B., Grover, P., Boufounos, P., Nabi, S. & Benosman, M. (2017), 'Sparse sensing and dmd-based identification of flow regimes and bifurcations in complex flows', *SIAM Journal on Applied Dynamical Systems* **16**(2), 1164–1196.

Kunisch, K. & Volkwein, S. (2002), 'Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics', *SIAM Journal on Numerical analysis* **40**(2), 492–515.

Le, Q., Sarlós, T. & Smola, A. (n.d.), Fastfood-approximating kernel expansions in loglinear time.

Lee, S., Kevrekidis, I. G. & Karniadakis, G. E. (2015), 'Resilient algorithms for reconstructing and simulating gappy flow fields in cfd', *Fluid Dynamics Research* **47**(5), 051402.

Liu, J.-G., Wang, C. & Johnston, H. (2003), 'A fourth order scheme for incompressible boussinesq equations', *Journal of Scientific Computing* **18**(2), 253–285.

Lophaven, S. N., Nielsen, H. B. & Søndergaard, J. (2002), Dace-a matlab kriging toolbox, version 2.0, Technical report.

Lumley, J. L. (2007), *Stochastic tools in turbulence*, Courier Corporation.

Marinoni, O. (2003), 'Improving geological models using a combined ordinary–indicator kriging approach', *Engineering Geology* **69**(1), 37–45.

Matsumoto, D. & Indinger, T. (2017), 'On-the-fly algorithm for dynamic mode decomposition using incremental singular value decomposition and total least squares', *arXiv preprint arXiv:1703.11004* .

Mezić, I. (2005), 'Spectral properties of dynamical systems, model reduction and decompositions', *Nonlinear Dynamics* **41**(1), 309–325.

Mezić, I. (2013), 'Analysis of fluid flows via spectral properties of the koopman operator', *Annual Review of Fluid Mechanics* **45**, 357–378.

Moreno, A. I., Jarzabek, A. A., Perales, J. M. & Vega, J. M. (2016), 'Aerodynamic database reconstruction via gappy high order singular value decomposition', *Aerospace Science and Technology* **52**, 115–128.

Moser, R. D., Kim, J. & Mansour, N. N. (1999), 'Direct numerical simulation of turbulent channel flow up to re τ= 590', *Physics of fluids* **11**(4), 943–945.

Muld, T. W., Efraimsson, G. & Henningson, D. S. (2012), 'Flow structures around a high-speed train extracted using proper orthogonal decomposition and dynamic mode decomposition', *Computers & Fluids* **57**, 87–97.

Needell, D. & Tropp, J. A. (2009), 'Cosamp: Iterative signal recovery from incomplete and inaccurate samples', *Applied and Computational Harmonic Analysis* **26**(3), 301–321.

Noack, B. R., Afanasiev, K., Morzynski, M., Tadmor, G. & Thiele, F. (2003), 'A hierarchy of low-dimensional models for the transient and post-transient cylinder wake', *Journal of Fluid Mechanics* **497**, 335–363.

Noack, B. R., Schlegel, M., Morzynski, M. & Tadmor, G. (2011), Galerkin method for nonlinear dynamics, *in* 'Reduced-Order Modelling for Flow Control', Springer, pp. 111–149.

Oliver, M. A. & Webster, R. (1990), 'Kriging: a method of interpolation for geographical information systems', *International Journal of Geographical Information System* **4**(3), 313–332.

Oxberry, G. M., Kostova-Vassilevska, T., Arrighi, W. & Chand, K. (2017), 'Limited-memory adaptive snapshot selection for proper orthogonal decomposition', *International Journal for Numerical Methods in Engineering* **109**(2), 198–217.

Pan, C., Yu, D. & Wang, J. (2011), 'Dynamical mode decomposition of gurney flap wake flow', *Theoretical and Applied Mechanics Letters* **1**(1).

Peherstorfer, B. & Willcox, K. (2015), 'Dynamic data-driven reduced-order models', *Computer Methods in Applied Mechanics and Engineering* **291**, 21–41.

Peherstorfer, B. & Willcox, K. (2016), 'Dynamic data-driven model reduction: adapting reduced models from incomplete data', *Advanced Modeling and Simulation in Engineering Sciences* **3**(1), 11.

Rahimi, A. & Recht, B. (2009), Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning, *in* 'Advances in neural information processing systems', pp. 1313–1320.

Rajani, B., Kandasamy, A. & Majumdar, S. (2009), 'Numerical simulation of laminar flow past a circular cylinder', *Applied Mathematical Modelling* **33**(3), 1228–1247.

Rapún, M.-L. & Vega, J. M. (2010), 'Reduced order models based on local pod plus galerkin projection', *Journal of Computational Physics* **229**(8), 3046–3063.

Roshko, A. (1954), 'On the development of turbulent wakes from vortex streets'.

Rowley, C. W. & Dawson, S. T. (2017), 'Model reduction for flow analysis and control', *Annual Review of Fluid Mechanics* **49**, 387–417.

Rowley, C. W., Mezić, I., Bagheri, S., Schlatter, P. & Henningson, D. S. (2009), 'Spectral analysis of nonlinear flows', *Journal of fluid mechanics* **641**, 115–127.

Saini, P., Arndt, C. M. & Steinberg, A. M. (2016), 'Development and evaluation of gappy-pod as a data reconstruction technique for noisy piv measurements in gas turbine combustors', *Experiments in Fluids* **57**(7), 1–15.

San, O. & Borggaard, J. (2015), 'Principal interval decomposition framework for pod reduced-order modeling of convective boussinesq flows', *International Journal for Numerical Methods in Fluids* **78**(1), 37–62.

Schmid, P. J. (2010), 'Dynamic mode decomposition of numerical and experimental data', *Journal of fluid mechanics* **656**, 5–28.

Schmid, P. J. (2011), 'Application of the dynamic mode decomposition to experimental data', *Experiments in fluids* **50**(4), 1123–1130.

Schmid, P. J., Li, L., Juniper, M. P. & Pust, O. (2011), 'Applications of the dynamic mode decomposition', *Theoretical and Computational Fluid Dynamics* **25**(1-4), 249–259.

Seena, A. & Sung, H. J. (2011), 'Dynamic mode decomposition of turbulent cavity flows for self-sustained oscillations', *International Journal of Heat and Fluid Flow* **32**(6), 1098–1110.

Smits, A. J. & Marusic, I. (2013), 'Wall-bounded turbulence', *Phys. Today* **66**(9), 25–30.

Taira, K., Brunton, S. L., Dawson, S., Rowley, C. W., Colonius, T., McKeon, B. J.,

Schmidt, O. T., Gordeyev, S., Theofilis, V. & Ukeiley, L. S. (2017), 'Modal analysis of fluid flows: An overview', *arXiv preprint arXiv:1702.01453* .

Traa, J. (n.d.), 'Matrix calculus - notes on the derivative of a trace'.

Tritton, D. (1959), 'Experiments on the flow past a circular cylinder at low reynolds numbers', *Journal of Fluid Mechanics* **6**(04), 547–567.

Tropp, J. A. & Gilbert, A. C. (2007), 'Signal recovery from random measurements via orthogonal matching pursuit', *IEEE Transactions on information theory* **53**(12), 4655–4666.

Tu, J. H., Rowley, C. W., Luchtenburg, D. M., Brunton, S. L. & Kutz, J. N. (2013), 'On dynamic mode decomposition: theory and applications', *arXiv preprint arXiv:1312.0041* .

Venturi, D. & Karniadakis, G. E. (2004), 'Gappy data and reconstruction procedures for flow past a cylinder', *Journal of Fluid Mechanics* **519**, 315–336.

Weinan, E. & Shu, C.-W. (1998), 'Small-scale structures in boussinesq convection', *Physics of Fluids* .

Willcox, K. (2006), 'Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition', *Computers & fluids* **35**(2), 208–226.

Williams, M. O., Kevrekidis, I. G. & Rowley, C. W. (2015), 'A data–driven approximation of the koopman operator: Extending dynamic mode decomposition', *Journal of Nonlinear Science* **25**(6), 1307–1346.

Williams, M. O., Rowley, C. W. & Kevrekidis, I. G. (2014), 'A kernel-based approach to data-driven koopman spectral analysis', *arXiv preprint arXiv:1411.2260* .

Williamson, C. (1989), 'Oblique and parallel modes of vortex shedding in the wake of a circular cylinder at low reynolds numbers', *Journal of Fluid Mechanics* **206**, 579–627.

Wu, Y.-T. & Porté-Agel, F. (2011), 'Large-eddy simulation of wind-turbine wakes: evaluation of turbine parametrisations', *Boundary-layer meteorology* **138**(3), 345–366.

Wynn, A., Pearson, D., Ganapathisubramani, B. & Goulart, P. J. (2013), 'Optimal mode decomposition for unsteady flows', *Journal of Fluid Mechanics* **733**, 473–503.

Yates, F. (1933), 'The analysis of replicated experiments when the field results are incomplete', *Empire Journal of Experimental Agriculture* **1**(2), 129–142.

Yildirim, B., Chryssostomidis, C. & Karniadakis, G. (2009), 'Efficient sensor placement for ocean measurements using low-dimensional concepts', *Ocean Modelling* **27**(3), 160–173.

VITA

CHEN LU

Candidate for the Degree of

Masters of Science

Thesis: SPARSE DATA-DRIVEN MODELING FRAMEWORK FOR NONLINEAR FLUID FLOWS

Major Field: Mechanical and Aerospace Engineering

Biographical:

Personal Data: Born in GuangDong, China in October of 1993.

Education:

Received a Bachelor of Science in Mechanical Engineering from Oklahoma State University, Stillwater, Oklahoma in May 2015.
Completed the requirements for the degree of Master of Science with a major in Mechanical and Aerospace Engineering at Oklahoma State University in December, 2017.

Experience:

After starting M.S. program in Oklahoma State University, worked as graduate research and teaching assistant for two and half years.

Professional Affiliations:

Student member: AIAA.