Imperial College of London
Department of Aeronautics

# Data Driven Techniques for Modal Decomposition and Reduced-Order Modelling of Fluids

Mohammad Beit-Sadi

September 17, 2023

Supervised by Dr. Andrew Wynn

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Aeronautics of Imperial College of London
and the Diploma of Imperial College London

I hereby certify that all work presented in this thesis is my own and that all other contributions have been appropriately acknowledged and referenced.

To those who have not been as fortunate,
as many times,
in as many ways.

# Abstract

In this thesis, a number of data-driven techniques are proposed for the analysis and extraction of reduced-order models of fluid flows. Throughout the thesis, there has been an emphasis on the practicality and interpretability of data-driven feature-extraction techniques to aid practitioners in flow-control and estimation.

The first contribution uses a graph theoretic approach to analyse the similarity of modes extracted using data-driven modal decomposition algorithms to give a more intuitive understanding of the degrees of freedom in the underlying system. The method extracts clusters of spatially and spectrally similar modes by post-processing the modes extracted using DMD and its variants.

The second contribution proposes a method for extracting coherent structures, using snapshots of high dimensional measurements, that can be mapped to a low dimensional output of the system. The importance of finding such coherent structures is that in the context of active flow control and estimation, the practitioner often has to rely on a limited number of measurable outputs to estimate the state of the flow. Therefore, ensuring that the extracted flow features can be mapped to the measured outputs of the system can be beneficial for estimating the state of the flow.

The third contribution concentrates on using neural networks for exploiting the nonlinear relationships amongst linearly extracted modal time series to find a reduced order state, which can then be used for modelling the dynamics of the flow. The method utilises recurrent neural networks to find an encoding of a high dimensional set of modal time series, and fully connected neural networks to find a mapping between the encoded state and the physically interpretable modal coefficients. As a result of this architecture, the significantly reduced-order representation maintains an automatically extracted relationship to a higher-dimensional, interpretable state.

# Acknowledgements

I am immensely grateful to my supervisor, Dr Andrew Wynn, for his help, guidance and patience at every turn. The start, the progression and the completion of this work would have been impossible without his indispensable help.

I would like to express my gratitude to Dr Krishna Kankanwadi, Dr Oliver Buxton and Dr Rowan Brackston, for granting us access to velocity field snapshots from the wake of a rectangular cylinder and the wake of an axisymmetric object, respectively.

I would also like to thank my family for their support and understanding over the many years. Truly, their sacrifices have made the completion of this work possible.

In the end, I would like to acknowledge and express my gratitude to EPSRC for the funding of this project.

# Contents

# List of Publications

### Journal Publications

- **Beit-Sadi, M.**, Kroll, J., Wynn. (2021). "Data-driven feature identification and sparse representation of turbulent flows". *International Journal of Heat and Fluid Flow.*

### Conference Proceedings

- **Beit-Sadi, M.**, Kroll, J., Wynn. (2019). "Data-driven feature identification and sparse representation of turbulent flows". 11*th International Symposium on Turbulence and Shear Flow Phenomena.*

# Nomenclature

| | |
|---|---|
| DMD | Dynamic Mode Decomposition |
| GRU | Gated Recurrent Unit |
| LSTM | Long Short-Term Memory |
| NN | Neural Network |
| OMD | Optimal Mode Decomposition |
| POD | Proper Orthogonal Decomposition |
| RNN | Recurrent Neural Network |
| ROM | Reduced-Order-Model |
| SVD | Singular Value Decomposition |

# List of Figures

# List of Tables

Chapter 1

# Introduction

## 1.1 Coherent structures and reduced order modelling

### 1.1.1 A brief historical account

It is often the case that to understand a complex and intricate system, humans need to break its behaviour into simpler constituent parts. Studying the interactions between various simpler subsystems can then lead to a better understanding of the underlying system's behaviour. A fundamental class of highly complex systems for which extraction of human-interpretable information is desirable are fluid flows.

Although fluid flows are well characterised by the Navier-Stokes equations, the inherent non-linearity and mathematical complexity of the governing equations leads to difficulties in extracting a holistic view of a flow's behaviour. As a result, practitioners often characterise a flow with spatially consistent, and temporally persistent, events that can be extracted from the flow. Furthermore, for these events to be of physical relevance, they should make a significant contribution to, some or all, appropriately-averaged statistics of the flow. Such events are often referred to as coherent structures of a fluid flow [2, 105].

This approach is especially important in the case of turbulent flows, where the broadband and chaotic nature of the underlying dynamics creates difficulties in understanding the behaviour of the fluid flow. While in laminar flows, structures such as the Kármán

vortex street were observed in early 1900s [5], the structural view of turbulent flows was only popularised in the 1970s, particularly after Brown and Roshko's study of turbulent mixing layers showed that the behaviour of such layers is dominated by large coherent structures, which entrain fluid from the non-turbulent background flow into the turbulent flow region [28]. A brief historical overview of these structures in the literature can paint a picture of the opportunities and challenges of using coherent structures to understand fluids.

What makes the structural view of fluid flows an attractive proposition is that in most cases, such structures persist, in a statistical sense, for a wide range of flow conditions. This implies that coherent structures with interpretable features (e.g., large spatial length scales) which are observed in lower Reynolds number flows may retain their dynamic importance even at higher Reynolds numbers, providing a compelling argument for studying such structures. For example, early studies of mixing layers at lower Reynolds numbers [55, 120], had already shown similar structures to those observed in the turbulent regime [28]. Importantly, it is also the case that structures extracted from empirical data show a degree of consistency with theoretical results. For example, the structures observed in mixing layers in [28] and [55] were shown to be consistent with the theoretical results from Kelvin-Helmholtz stability theory, at least in a local sense [58, 79].

The studies of many other canonical flows have benefited from this structural view. Among many others, coherent structures have been used to further our understanding of flows over cavities [143, 155, 70], in the modelling and control of the behaviour of jets [141, 36, 131], have enhanced our understating of the scale interactions in turbulent boundary layers [114, 83, 177], and have been used to analyse, model and control flows past bluff bodies [128, 153, 142].

It is worth emphasising that the appeal to coherent structures for flow modelling and control in each of the above examples is not merely convenient, but is often necessary. In fluid dynamics, the computational cost of solving the governing equations implies that, for most estimation and control purposes, practitioners must rely on simplified models of

a flow since real-time decisions require rapid model evaluation. With a view towards flow modelling, it is interesting to note that insights from the process of extracting coherent structures from data can often hint at an appropriate structure for creating reduced-order flow models. We next introduce a general class of coherent structures, which we refer to as *dynamical features*, which are particularly suitable for constructing interpretable models for fluid flows and then discuss, in §2, the statistical methods for computing them.

### 1.1.2 Dynamical features and Reduced Order Modelling

Reduced order models (ROMs) are simplified models that only attempt to describe, loosely-speaking, a subset of a system's dynamical behaviour. As discussed above, ROMs are particularly attractive for flow control and estimation purposes, since they provide a low dimensional state of the flow which can be used to drive estimation and control design. However, ROMs should not only be viewed as a stepping stone towards application of control-theoretic techniques. At their best, ROMs should provide practitioners with an intuitive understanding of the underlying physics of a system.

The natural question that arises from the above definition of reduced order models is: *which dynamical behaviours should a ROM capture?* Heuristically, a ROM would seek to capture the dynamics of a flow's dominant coherent structures. However, the definition of coherent structures in §1.1.1 is very broad and can cover a wide range of dynamical features. As a result, the quality or applicability of a ROM is strongly dependent on the chosen coherent structures that it captures.

More precise definitions of coherent structures can be provided by giving a more specific definition of coherence. For example, a coherent structure may be defined as a local region of spatial coherence in one or more fundamental components of the flow (e.g. velocity components or vorticity). Such a definition corresponds well with, for example, the long spatially-meandering features of alternating velocity fluctuations in the log-layer of a wall bound flow [83], and with the distinct regions of correlated vorticity which comprise the Kármán vortex street [144]. In addition, and in the latter case most distinctly,

such coherent structures may also be associated with a distinct and coherent temporal frequency.

In this work, we will primarily focus on a specific type of coherent structure which possesses both spatial and temporal coherence, and which we refer to as a *dynamical feature*. Dynamical features are spatio-temporal structures that act as a constituent part of the fluid flow. Each dynamical feature has a spatial component, describing its spatial characteristics at different points in the domain, and a temporal component, describing its evolution in time. The flow field can then be approximated by superposition of dynamical features, for example, by using Galerkin projection as will be explained shortly. The advantage of using dynamical features as the building blocks of ROMs is that they provide a natural link between the process of extraction of coherent features and the construction of models for a flow's dynamics. That is, by demanding temporal coherence of the extracted features we expect to gain some regularity when building ROMs from them.

Finding dynamical features of a flow from either data or direct mathematical analysis, relies on a variety of mathematical and statistical tools. Tools from linear algebra such as subspace projection and matrix decomposition techniques (discussed in detail in §2) are vital and can often be expressed as constrained optimisation problems formulated so as to ensure that the extracted coherent structures do in fact replicate aspects of the behaviour of the original system.

There are two broad approaches for obtaining dynamical features in fluid mechanics literature. In the first approach, one appeals directly to the governing equations—for example via eigendecomposition of linearised operators—and finds features and models by applying restrictions on the scale interactions within the flow and numerically characterising the results of the simplified system [15, 157, 10].

A second approach is instead to start from numerical or experimental data and, through the application of statistical and data-driven methods, find dynamical features that make a significant contribution to the system's behaviour. The extracted features can then be used to derive ROMs for the flow.

The general framework for linking spatial features with temporal information, and hence the framework for creating reduced order models is typically inspired by *Galerkin projection*. Since in this study, and in almost all data-driven settings, data is only available in discrete form (both spatially and temporally), we will now discuss Galerkin projection in the context of a discretised model.

Suppose that the velocity field is given by $\mathbf{u}(\mathbf{x}, t)$ at spatial locations $\mathbf{x}$ and at times $t$. Given a finite and *fixed* set of spatial features $(\phi_i(\mathbf{x}))_{i=1}^r$ (computed, for example, using one of the techniques described in §2) one can then form an approximate representation of the flow's velocity field using the expansion

$$\mathbf{u}(\mathbf{x}, t) = \sum_{i=1}^{r} a_i(t)\phi_i(\mathbf{x}) + \nu_i, \tag{1.1}$$

where the coefficients $a_i(t)$ describe the temporal relevance of each spatial feature $\phi_i$ at time $t > 0$, and $\nu_i$ denotes the residual of the approximation. In theory, the expansion (1.1) can be substituted into the Navier-Stokes equations then, after taking successive inner products taken with each spatial feature $\phi_i$, ordinary differential equations governing the temporal terms $a_i(t)$ can be obtained.

The importance of this method is that for any chosen (or constructed) set of spatial mode shapes, there exists a systematic method of extracting a ROM for their dynamics which is derivable directly from the underlying equations of fluid motion. Unfortunately, when the underlying features $\phi_i$ are computed from data (i.e. the second approach mentioned above) then the application of Galerkin projection may often present severe difficulties. These arise because computation of the necessary inner products requires access to *all components* of the underlying flow, including full spatial information across the whole domain and knowledge of the pressure field $p(\mathbf{x}, t)$. As will be detailed in §2 below, spatial features $\phi_i(\mathbf{x})$ extracted from an ensemble of flow data can only contain information of the type contained in the data ensemble from which they are computed. For experimental data in particular, such information is *incomplete*, for example only including a restricted

spatial domain and may not include information about all velocity components or pressure. A lack of such information in the features $\phi_i(\mathbf{x})$ implies that the inner product calculations required to obtain dynamic models from the Galerkin procedure may either be impossible to compute, or may only be crude approximations to their true values. It is well-known that such limitations can restrict the validity of the resulting ROMs [139], with [127] showing further that, for shear flows, it is crucial that pressure information is included in the underlying data ensemble if an effective ROM is to be computed via the Galerkin approach.

So far, we have seen that given an ensemble of spatial structures $(\phi_i(\mathbf{x}))_{i=1}^r$, one can, theoretically at least, create a reduced-order model of the underlying flow. However, in cases where the collected data is partial, sparse or corrupted by measurement noise, it may not be possible to derive analytical models for the evolution of the associated time series coefficients $a_i(t)$. Consequently, for such cases, the model extraction process must rely on empirical or statistical methods. This suggests a challenging general question of how to optimally create spatial features $\phi_i(\mathbf{x})$, which represent interpretable coherent structures, whose combination gives a good representation of the underlying flow, while also ensuring that their associated temporal coefficients $a_i(t)$ are easy to model dynamically.

As will be discussed in Chapter 2, these two aims are typically in competition for well-established algorithmic methods for data-driven dynamical feature extraction. As a result, the question of how to optimally obtain a set of coherent dynamical features of a fluid flow remains an active and important open question in fluid mechanics, and the aim of this thesis is to make progress towards answering it.

## 1.2 Challenges for data-driven dynamical feature extraction in fluid mechanics

In this thesis we will consider three open questions related to the optimal creation of dynamical features from data-ensembles of fluid flow data. These three questions all relate to extending feature extraction algorithms in order to prioritise additional beneficial characteristics in the feature extraction process. Specifically, we ask if it is possible to: 1. Identify small sets of dominant dynamical features from a large initial set of features; 2. Create dynamical features which can facilitate estimation of a flow field from limited sensor information; and 3. Utilise interpretable dynamical features to promote interoperability of nonlinear reduced-order dynamical models for fluid flows. These open questions are now explained in more detail.

**Question 1: Given an ensemble of fluid flow data, is there a systematic way of extracting a small number of dynamical features which represent physically important flow features?**

Standard methods of obtaining dynamical features, which will be detailed in Chapter 2, typically create a large number of modes to provide an accurate representation of the underlying flow. Specifically, the dimension $r$ in (1.1) can be large (often of order $10^3$ or more) in order to give a small fitting residual $\nu_i$. From such a large initial set of features, it is clearly desirable to select a much smaller subset of features to enable physical flow analysis or to use as a basis for reduced-order modelling.

For well-established feature extraction algorithms, there exist methods of ranking the obtained dynamical features via energetic content or spectral information. However, as will be discussed in Chapter 3, it is often found that a large number of spatial features $\phi(\mathbf{x})$ are extracted which relate to the same underlying physical dynamical feature in a flow and, in such cases, standard methods of energetic or spectral ranking are not effective

for selecting a small number of features for analysis. This behaviour is particularly problematic for flows at high Reynolds numbers which contain flow features across significant ranges of length and time scales.

To address this problem, Chapter 3 of this thesis introduces an algorithm that can post processes non-orthogonal modes, such as those produced by the widely used Dynamic Mode Decomposition algorithm (see §2.3 for a definition), to find clusters of similar modes. The clusters provide an interpretable view of the degrees of freedom of the underlying flow, and a systematic method of combining possibly large numbers of similar features to enable dimension reduction. The developed algorithm is demonstrated via application to flows with a wide range of Reynolds numbers. The work in this Chapter has been published in [18].

**Question 2: Can the extraction of dynamic features be adapted to take into account a known measurement process?**

Most leading feature extraction algorithms work by solving an optimisation problem whose cost is designed to optimally (in some sense) reconstruct the full, high-dimensional, underlying snapshots contained in a data ensemble of a flow field. The dynamical features $\phi_i(\mathbf{x})$ extracted using such methods, and hence their time-varying mode coefficients $a_i(t)$, are consequently not computed with reference to other dynamical processes that may occur simultaneously. One example of practical importance is if real-time sensor information is to be used, for example, as a signal to estimate the full flow field, or for use in a feedback control strategy.

A problem that may occur is that if a measurement process is not considered during data-driven extraction of dynamical features, then these features (and any reduced order model computed from them) may not perform well when used in conjunction with the specific real-time measurement process available. While some progress has been made in the literature concerning dynamical feature extraction with reference to the structure of

an actuator input (which is discussed in §2.6), there has to date been little research on the problem of feature extraction with reference to a measurement process.

To address this problem, in Chapter 4 of this thesis, we develop the Output-regulated Optimal Mode Decomposition algorithm. This extracts dynamical features from a data ensemble by solving an optimisation problem which balances the cost of capturing the flow dynamics with that of obtaining features which are well correlated with a known measurement process. This provides a new feature extraction methodology which creates "sensor-aware" or "observable" coherent structures.

**Question 3: Can efficient reduced order states, extracted using nonlinear data compression techniques, maintain a connection to interpretable, linearly extracted coherent structures?**

Many established techniques for feature extraction in fluid mechanics, including those which underpin the two questions above, rely on linear theory to motivate their methods of feature selection. A natural question, in light of the recent and rapid development of nonlinear black-box modelling approaches such as neural networks, is whether such nonlinear approaches can be exploited to improve the performance of feature extraction algorithms in fluid mechanics. While existing linear modal decomposition techniques are not as efficient as neural networks in reconstructing snapshots of a flow-field, they do extract human-interpretable features at a low computational cost. An important question is therefore to ask whether the reduced interpretability of black-box methods poses any barrier to their robust use in fluid flow modelling, estimation, and control.

Chapter 5 of this thesis investigates this question by seeking to combine neural network modelling techniques with traditional linear coherent structure extraction methods to generate reduced order models for fluid flows. The objective to is to find the correct balance, in the context of flow modelling, between model interpretability and the increased modelling capability of recent nonlinear machine learning-based modelling.

## 1.3 Outline of Thesis

The remaining chapters of the thesis are organised as follows.

Chapter 2 gives an in-depth literature review of the various dynamical feature extraction algorithms widely used in the fluid mechanics community. The purpose of this chapter is to give sufficient mathematical and algorithmic detail of these methods to both motivate and explain the extensions of them that will be developed subsequently in the thesis with a view to answering the three questions highlighted in §1.2. The three fundamental linear algorithms discussed are Proper Orthogonal Decomposition (POD), Dynamic Mode Decomposition (DMD) and Optimal Mode Decomposition (OMD), each of which can be viewed as adding complexity to the former methodology. The methodology chapter concludes with a brief review of neural network methods for data-driven modelling, which will be used in the final chapter to enable nonlinear data-driven modelling of fluid flows.

Chapter 3 develops a new algorithmic method for reducing the dimension of the models, or equivalently the projection basis provided by common modal decomposition methodologies. The motivation for this work is that common methods, such as Dynamic Mode Decomposition, typically output a large number of dynamical features which are non-orthogonal. Due to this non-orthogonality, the data subspace spanned by these features can potentially be described by significantly fewer modes than those outputted by such algorithms. By appealing to the graph-theoretic technique of *maximal cliques*, this chapter presents a new method of clustering dynamic modes which are likely to describe the same underlying feature present in a data ensemble. A method of ranking the importance of the obtained clusters is then proposed to enable clear criteria for dimension reduction. Detailed analysis of the algorithm's performance on three bluff body flows, of increasing complexity, is presented and it is observed that the clustering algorithm is able to identify a small number of dynamical features which correspond to known modes of all considered flows. This chapter seeks to answer research Question 1, described in the previous section.

Chapter 4 develops a new extension to the Optimal Mode Decomposition (OMD) algorithm, with the intention to prioritise dynamical features which are in the observable subspace of a known measurement process. The proposed method takes advantage of the fact, described in the methods Chapter 2, that OMD is a two-stage optimisation algorithm which seeks to identify both a projection subspace and linear dynamics as part of the modelling process. Due to the freedom implied by the subspace modelling component of OMD, it is possible to propose a natural extension of the algorithm which prioritises the creation of dynamical features which are better correlated (i.e., more observable) than those nominally produced by the algorithm. The developed algorithm's performance is analysed on a number of synthetic test-cases in which observability is well defined, before being applied to a challenging high Reynolds number bluff body test case. The method developed in this chapter seeks to answer Research Question 2, as described in the previous section.

Chapter 5 investigates the use of neural networks for nonlinear data-driven modelling of fluid flows. Modelling is performed using autoencoder neural networks applied to the low-dimensional subspace of amplitude coefficients obtained by a nominal feature extraction methodology, such as DMD. The use of neural network methods is shown to enable more efficient dimension reduction than modal truncation, due to the nonlinear decoder and autoencoder mappings employed. It is shown for the canonical cylinder wake flow that shedding mode harmonics may be modelled effectively using only minimal degrees of freedom associated with a dominant global mode, with harmonic dynamics captured via nonlinear mappings. The performance of nonlinear autoencoders for both dimension reduction and dynamic modelling is compared with that produced by existing modal decomposition approaches. This chapter addresses the final research Question 3, introduced in the previous section.

Chapter 6 summarises the work presented in the thesis, as well as detailing some possible extensions of the approaches presented in an outlook for future research.

Chapter 2

# Modal Decomposition Methods for Fluids Analysis

We have seen that to understand, analyse, and model fluid flows, one can concentrate on the coherent structures of the flow. We have also defined dynamical features of the flow as spatiotemporal patterns that, at least in the aggregate, approximate the behaviour of the original flow. Here we will review ways in which such features can be extracted from data and then subsequently be used for Reduced Order Model (ROM) construction and flow-control purposes. In this context, coherent structures are often referred to as *modes*, and modal decomposition methodologies are the class algorithms typically used to identify them. In the following, we will cover some of the most relevant modal decomposition techniques to the current study. For a thorough review of modal decomposition techniques in fluid mechanics, the reader is also referred to [159].

## 2.1 Proper Orthogonal Decomposition

For completeness, we first consider the Proper Orthogonal Decomposition (POD) algorithm since its use underpins many more modern modal decomposition methodologies. POD was introduced in the context of fluid mechanics by Lumley [109], and is equivalent to the Karhunen–Loeve decomposition and Principle Component Analysis, PCA, in

statistics. The method decomposes fluid flow data into energetically optimal, spatially orthogonal features. Due to its algorithmic simplicity (as explained below) and clear justification in terms of energetic optimality, POD has been used in a wide range of studies in fluid mechanics. For example, POD has been used to study bluff body wakes including two dimensional cylinders [128], finite-length cylinders [169] and axisymmetric bluff bodies [142]. Other examples include the study of flat plate boundary layers [9, 140], pipe flows [74], jets in cross-flow [36, 117] and swirling jets [131].

The POD approach addresses the scenario in which we wish to linearly decompose a flow field $\mathbf{u}(\mathbf{x}, t)$, in terms of a set of pairwise orthogonal modes $(\phi_i(\mathbf{x}))_{i=1}^r$, as in (1.1), with the property that such a reduced-order representation with $r \in \mathbb{N}$ degrees of freedom captures the maximum energy (in the sense of the Euclidean norm) possible out of all such linear decompositions with $r$ spatial mode shapes.

To describe the POD algorithm, we assume that at $N$ sample points in time $t_1, t_2, \ldots, t_N$, velocity field data is available at $\tilde{p} \in \mathbb{N}$ spatial locations in the flow, and that at each location $\tilde{d} \in \mathbb{N}$ scalar-valued components of flow data are known (e.g. velocity components, pressure). Letting $p = \tilde{d}\tilde{p} \in \mathbb{N}$, it follows that at sample time $t_j$ the available data—typically referred to as a *snapshot*—can be arranged as a vector $x_j \in \mathbb{R}^p$. For the purposes of analysis, it is convenient to arrange the $N$ snapshots into a data matrix

$$X = \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ x_1 & x_2 & \cdots & x_N \\ \downarrow & \downarrow & & \downarrow \end{pmatrix} \in \mathbb{R}^{p \times N}.$$

The aim is to extract a set of orthogonal coherent structures, the POD modes $\phi_i \in \mathbb{R}^p$, which satisfy

$$\phi_i^\top \phi_j = \delta_{ij},$$

where $\delta_{ij}$ is the Kronecker delta function, with the additional property that each subset $(\phi_i)_{i=1}^r$ of the first $r \leq N$ modes captures the maximum energy of the underlying

data $X$ possible with $r$ mode shapes. This aim can be rigorously posed via the following optimisation problem

$$\begin{aligned} \underset{\boldsymbol{\Phi}}{\text{minimise}} \quad & \left\| X - \Phi\Phi^\top X \right\|_F^2, \\ \text{subject to} \quad & \Phi^\top\Phi = \mathbf{I}, \\ & \Phi \in \mathbb{R}^{p\times r}, \;\; 1 \leq r \leq N, \end{aligned} \tag{2.1}$$

where $\|\cdot\|_F$ denotes the Frobenius matrix norm*. The resulting POD modes $\phi_i$ are extracted as columns of the optimal matrix $\Phi$, i.e.,

$$\Phi = \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ \phi_1 & \phi_2 & \cdots & \phi_r \\ \downarrow & \downarrow & & \downarrow \end{pmatrix}.$$

The optimisation problem 2.1 formulates the POD algorithm by minimising the error of the projection operator $(I - \Phi\Phi^\top)$ when applied to the data ensemble $X$.

## Solving the POD optimisation problem

To solve (2.1), one can use the identity $\|A\|_F^2 = \mathrm{tr}(AA^\top)$, where $\mathrm{tr}(\cdot)$ is the trace operator, and the orthonormality of the matrix $\Phi$ to rewrite the cost function as

$$\begin{aligned} \left\| X - \Phi\Phi^\top X \right\|_F &= \mathrm{tr}\Big( \big(X - \Phi\Phi^\top X\big)\big(X - \Phi\Phi^\top X\big)^\top \Big) \\ &= \mathrm{tr}(XX^\top) - \mathrm{tr}(XX^\top\Phi\Phi^\top) \\ &= \left\|X\right\|_F^2 - \left\|\Phi^\top X\right\|_F^2. \end{aligned}$$

It is therefore clear that minimisation in (2.1) is equivalent to maximisation of the cost $\left\|\Phi^\top X\right\|_F^2 = \mathrm{tr}(\Phi^\top XX^\top\Phi)$ subject to the same orthogonality constraints as in (2.1). It is well known that this cost function is maximised if the columns of $\Phi$ are taken as a subset of the necessarily orthogonal eigenvectors of the positive semi-definite matrix $XX^\top$,

* In particular for a matrix $A = (a_{ij}) \in \mathbb{R}^{n\times m}$ we have $\|A\|_F^2 = \sum_{i=1}^n \sum_{j=1}^m |a_{ij}|^2$.

specifically the $r$ eigenvectors corresponding to the $r$ largest (real) eigenvalues of $XX^\top$. The importance of the above observation is that the eigenvectors of the matrix $XX^\top$ are inherently linked to the notion of the Singular Value Decomposition (SVD) of the matrix $X$.

The SVD of the matrix $X \in \mathbb{R}^{p\times N}$ is its decomposition in the form

$$X = \Phi\Sigma V^\top, \tag{2.2}$$

where the columns of $\Phi \in \mathbb{R}^{p\times N}$ are eigenvectors of $XX^\top$, the matrix $\Sigma \in \mathbb{R}^{N\times N}$ is a diagonal and contains the singular values $\sigma_i \geq 0$ (the square roots of the eigenvalues of $XX^\top$), which quantifies the energetic contribution of each mode $\phi_i$ to the underlying data ensemble $X$. Finally, the rows of the matrix $V^\top$ can be used to compute temporal coefficients $a_i(t_j)$ describing the relative importance of each mode at each sample time $t_j$.

In the above discussion, it is important to note that we choose to calculate $r \leq N$ modes from the POD analysis. That is, we can extract less modes than the number of available snapshots for data analysis. This is desirable, since our underlying aim is to form reduced-order models. The key assumption for POD-based order reduction is that a given flow field can be expressed as a superposition of spatially coherent, more highly energetic, dynamically relevant modes and spatially incoherent, low-energy modes. By discarding the low-energy modes, POD-based order reduction aims to keep the most energetically dominant subspace of the observed flow. The equivalence of POD and SVD has the fortunate implication that, due to the Eckart-Young theorem [47], the first $r$ POD modes are known to be the optimal $r$-dimensional bases for minimising the reconstruction error $\|X - \Phi_r\Phi_r^\top X\|_F^2$. However, this does not address the fact that finding a sensible dimension for the reduced order model—i.e., a sensible choice for the cut-off dimension $r$—typically requires prior knowledge of the underlying data ensemble and how it was obtained.

## Methods of Truncation

Typically, a user will aim to retain as many modes as necessary to reconstruct a certain percentage of the flow's energy [26], for example, selecting the smallest possible $1 \leq r \leq N$ for which

$$\frac{\|X - \Phi_r \Phi_r^\top X\|_F}{\|X\|_F} \leq 0.1$$

if one wants the smallest dimensional model which can capture 90% of the ensemble energy. However, more recent developments use statistical insights for choosing an appropriate subset of POD modes [52, 59]. Here we will briefly explain two such approaches.

For a dataset $X \in \mathbb{R}^{p \times N}$ assume that $X = \hat{X} + \alpha Z$, where $\hat{X}$ contains the noise-free measurements, $\alpha$ is the noise amplitude and the columns of matrix Z are identically and independently distributed Gaussian noise with zero mean and unit variance. If the noise amplitude $\alpha$ is known, the method presented in [59] finds the optimal threshold for the singular values as a function of the dimensions $p$ and $N$. For example, if the snapshot matrix is a square matrix $X \in \mathbb{R}^{N \times N}$, the optimal cut-off for the singular values is determined by

$$\lambda = \frac{4}{\sqrt{3}} \sqrt{N} \alpha.$$

This implies that any mode with a singular value smaller than $\lambda$ should be truncated out of the POD set. Importantly, this method allows the practitioner to reduce the problem of finding a sensible dimension for a ROM, to one of quantifying the amplitude of noise in the underlying data ensemble.

The Entropy Line Fit (ELF) method [26] provides an alternative method of selecting a subset of POD modes, which is applicable when the nature and amplitude of noise is unknown. EFL uses an information theoretic approach to determine the level of randomness of each POD mode. The ELF method computes the Shannon entropy of the discrete cosine transform of each POD mode $\phi_i$ and its corresponding time series $(a_i(t_j))_{j=1}^N$. Modes are then retained or discarded according to a predetermined entropy threshold.

## 2.2 Extensions of POD

Recent extensions of the POD algorithm have concentrated on promoting sparsity in the spatial modes $\Phi$ or in the temporal information contained in the matrix $V^\top$ of the associated SVD decomposition (2.2). This is achieved through adding either $\ell_1$ and $\ell_2$ penalties to the cost function of the POD optimisation problem (2.1) in order to regularise the mode selection process [179, 43, 167].

In the context of reduced-order modelling, relevant extensions of the POD algorithm focus on extracting coherent structures that are more suitable for model discovery and control purposes. We will now briefly discuss some of the more widely used extensions of this type.

### Balanced POD

Balanced Proper Orthogonal Decomposition, BPOD, [145] is an extension of the POD algorithm that allows for tractable data-driven extraction of so-called balanced ROMs, explained below, of the underlying data. BPOD is based on an classical control theoretic concept of *balanced truncation* [123], which we now briefly introduce.

Consider the case of a real-valued, discrete-time linear dynamical system

$$x_{i+1} = Ax_i + Bu_i$$
$$y_i = Cx_i,$$

where $x_i \in \mathbb{R}^p$ is the state of the system, $A \in \mathbb{R}^{p\times p}$ is the state matrix, $u_i \in \mathbb{R}^{n_u}$ contains the control input to the system, $B \in \mathbb{R}^{p\times n_u}$ is the input matrix, $y_i \in \mathbb{R}^{n_y}$ is the measurable output of system and matrix $C \in \mathbb{R}^{n_y\times p}$ maps the state $x_i$ and the system output $y_i$.

The controllability of a state $x_i \in \mathbb{R}^p$, loosely defined as how easily the system is able to be driven to the state $x_i$ using a control input sequence $(u_i)_{i\geq 0}$, is quantified by the quadratic form $x_i^\top W_C x_i$, where

$$W_C = \sum_{m=0}^{\infty} A^m B B^\top (A^\top)^m \tag{2.3}$$

is the *controllability Gramian*, a positive definite matrix. Analogously, the observability of a state $x_i \in \mathbb{R}^p$ is defined as how well the state $x_i$ can be inferred using a measurable output sequence $(y_i)_{i\geq 0}$. This is quantified by the product $x_i^\top W_O x_i$ where

$$W_O = \sum_{m=0}^{\infty} (A^T)^m C^T C A^m \tag{2.4}$$

is known as the *observability Gramian*, also a positive definite matrix. The idea behind balancing is to find a linear coordinate transformation $z = Tx$ of the system for which, after transformation via

$$W_O \mapsto T^\top W_O T, \qquad W_C \mapsto T^{-1} W_C (T^{-1})^\top,$$

observability and controllability Grammians are diagonal and equal. That is, one hopes to find a transformation $T$ and a diagonal matrix $\Sigma$ such that

$$T^\top W_O T = T^{-1} W_C (T^{-1})^\top = \Sigma.$$

The interpretation of this is that states $z$ of the transformed system are as observable as they are controllable, which then provides a compelling criterion for model reduction. Indeed, balanced ROMs are constructed by eliminating states that are both hard to reach (i.e. require large control effort) and hard to observe (i.e. they are difficult to infer based on the measurable outputs). If the Grammians $W_C$ and $W_O$ are available, then it is easy to compute the matrix $T$ by finding and truncating the set of eigenvectors of the product $W_C W_O$. The difficulty in a data-driven setting or for high-dimensional systems with $p \gg 1$ is that the matrices $W_C$ and $W_O$ may be difficult or impossible to calculate, since they require the solution of large-scale Lyapunov equations.

Balanced POD is a data-driven method that addresses this problem, by approximating the Grammians directly from snapshots drawn from simulations of the underlying system's impulse response. In particular, if it is possible to create snapshots of both the impulse response of the system (arranged in a matrix $X$) and the impulse response of the adjoint system (arranged in a matrix $Y$), then the transformation matrix $T$ can be computed in terms of the singular value decomposition of the (smaller) matrix $Y^\top X = U\Sigma V^\top$. Indeed,

$$T = XV_1\Sigma_1^{-1/2},$$

where $V_1$ and $\Sigma_1$ are appropriately truncated forms of arising from the SVD. From this, the columns of the transformation matrix $T$ are viewed as the extracted modes of the flow and represent features which take into account not just their energetic contribution to the underlying data, as in POD, but also the relation of these features to the input and measurement processes of the system.

It is important to note that BPOD requires impulse response data from the *adjoint system*, which renders this method unsuitable for use with experimental data. As a result the application of this method [84, 3] and its modifications [46] have been limited to studies in which numerical simulation of the underlying flows is possible.

Finally, we note that in §4 we introduce a method that is inspired by, though not directly connected to, the BPOD algorithm in its attempt to balance energetic and dynamical considerations when extracting coherent structures and reduced-order representations of a fluid flow from data.

## Spectral POD

As POD modes are derived irrespective of temporal or spectral considerations, the temporal coefficients of spatial POD modes—that is the time series $a_i(t_j)$ defined corresponding to a decomposition (1.1)—generally contain a mix of frequencies. The spectral POD

(SPOD) algorithm [109, 165] addresses this issue using ideas from the problem of estimating the power spectral density of a signal. SPOD first divides the time-resolved snapshot ensemble $X \in \mathbb{R}^{p\times N}$ into $M$ overlapping blocks $x_i^\ell$

$$X^\ell = \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ x_1^\ell & x_2^\ell & \cdots & x_M^\ell \\ \downarrow & \downarrow & & \downarrow \end{pmatrix},$$

each containing $n$ of the original snapshots. The Discrete Fourier Transform (DFT) is then applied to each snapshot block. The Fourier coefficients $\hat{x}_f^\ell$ corresponding to each block $\ell$ and frequency $f$, are then collected rearranged according into matrices

$$\hat{X}_f = \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ \hat{x}_f^1 & \hat{x}_f^2 & \cdots & \hat{x}_f^M \\ \downarrow & \downarrow & & \downarrow \end{pmatrix}.$$

Once the DFT is known at each frequency $f$, one can then find the modes at each frequency $f$ by performing POD on the matrix

$$\hat{X}_f = \Phi_f \Sigma_f V_f^\top$$

after which the columns of $\Phi_f$ are defined to be the SPOD modes at frequency $f$. For simplicity, here we have assumed that the snapshots are appropriately scaled after the DFT procedure. For a more detailed explanation of scaling and windowing steps, see [165].

The advantage of SPOD over the original POD method is that, by definition, its modes are associated with a single frequency, providing a natural way of interpreting physical events. However, the disadvantage of this method is that it requires, in some cases significantly, more data (in terms of the number of snapshots) than the original POD method for numerical convergence. Moreover, due to the use of the underlying DFT method,

the temporal frequencies captured are on a uniform grid. As a result, much like spectral density estimation problems, the results obtained by SPOD require the chosen grid to be sufficiently fine for resolving the scales present in the data.

Alternative methods, such as Dynamic Mode Decomposition (DMD), which we will discuss subsequently in §2.3, address these issues. Instead of multiple discrete Fourier decompositions, DMD relies on a single eigendecomposition to extract spectral information from a flow. This eliminates the need to split a data ensemble into multiple segments, each of which must be capable of capturing various time-scales, and therefore DMD can potentially be suitable for application to smaller data ensembles. Moreover, the eigendecomposition in DMD does not require predefined frequency bins. The reader is referred further to [165] for a more detailed comparison of SPOD with Dynamic Mode Decomposition. Finally, we note that while we do not use SPOD in the current thesis, in Chapter 3 we introduce a method that can combine the versatility of DMD with the natural ordering of features in SPOD.

## 2.3 Dynamic Mode Decomposition and Related Algorithms

Dynamic Mode Decomposition (DMD) [152] and its many variants form a now widely employed class of modal decomposition algorithms, which concentrate on extracting coherent structures which are, in a sense, more dynamically interpretable than POD modes. The original motivation for DMD [146], as a possible improvement to POD, was that in many flows, large-scale coherent structures exist which oscillate at known dominant frequencies. In some situations, such as a jet in crossflow originally studied in [146], no single POD mode can be found whose time-dependent projection coefficients have spectral content which cleanly matches the known dominant frequencies of the underlying flow. Indeed, for this flow, the time series $a_i(t)$ of a given POD mode is observed to contain

frequencies associated with multiple flow features. In this sense, POD modes are often found to be spectrally polluted. With such behaviour in mind, the development of Dynamic Mode Decomposition arose as an attempt to extract individual coherent structures, each of which is associated with a single frequency. In cases where dominant frequencies exist in a flow, there is then hope that the DMD modes represent spatial structures which efficiently describe the underlying physics of the oscillatory structures. In this section, we will detail how the DMD algorithm achieves this extension of POD.

As expected, a strong theoretical relationship exists between DMD methods and the POD-based approaches of §2.1. At the same time DMD is also motivated by links with dynamical systems theory, and in particular the theory of Koopman operators [93, 146]. Unlike POD, where the goal is to find a projection that ensures energetic optimality, DMD concentrates on finding both modes and an optimal linear model for the dynamics of the flow observed in a given data ensemble. To explain DMD, which also underpins much of the novel algorithms proposed in this thesis, we begin by introducing the Arnoldi interpretation of the DMD algorithm as presented in [152], before discussing more numerically stable and noise-robust variants.

For clarity, we note that the Arnoldi DMD has not been explicitly used in this thesis, for reasons related to numerical stability discussed in §2.3.1. Instead, the Arnoldi interpretation of DMD is included in this thesis since it will allow a theoretical connection to be drawn between DMD and Koopman operator theory [146].

### 2.3.1 Arnoldi DMD

As in §2.1, we begin with an ensemble of $N+1$ snapshots of the velocity field of a fluid flow, where at each sample time $t_j$, the snapshot data is arranged as a column vector $x_j \in \mathbb{R}^p$. With a view to mode calculation which takes into account dynamical considerations, it is now assumed that the sample times are separated by a *common timestep* $\Delta t$. That is, $t_{j+1} = t_j + \Delta t$ for $j = 1, \ldots, N$. Data is then arranged into two snapshot matrices

$$X = \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ x_1 & x_2 & \cdots & x_N \\ \downarrow & \downarrow & & \downarrow \end{pmatrix} \in \mathbb{R}^{p\times N}, \quad X' = \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ x_2 & x_3 & \cdots & x_{N+1} \\ \downarrow & \downarrow & & \downarrow \end{pmatrix} \in \mathbb{R}^{p\times N}.$$

It is important to note that the snapshots in matrix $X'$ are shifted forward in time by one time-step $\Delta t$ compared to the snapshots in $X$.

The modelling assumption underpinning DMD is that of a linear mapping between the current snapshot of the flow field, $x_i \in \mathbb{R}^p$, and its future value, $x_{i+1} \in \mathbb{R}^p$ after one time-step $\Delta t$. In other words, DMD hopes to approximate the evolution of the flow over a single (small) time-step by a model of the form

$$x_{i+1} = Ax_i + \nu_{i+1}, \tag{2.5}$$

where $A \in \mathbb{R}^{p\times p}$ is a linear dynamics matrix and $\nu_{i+1}$ is the residual between the model and the true data. For cases where the number of collected snapshots of data $N$ is such that $p > N + 1$, which is typically the case in fluid dynamics, the fact that the unknown model contains $p^2$ degrees of freedom is a challenge for modelling: there are only $p(N+1)$ known data values, which may be significantly less than the number of free parameters in the model, and this will necessary lead to over-fitting if no action is taken to mitigate this problem. The approach that DMD takes to resolve this issue is to instead search for a lower-rank matrix in place of $A$. We now explain how this can be achieved.

Since one does not have a priori knowledge of matrix $A$, DMD's approach is based on a variant of the Arnoldi algorithm where explicit knowledge of $A$ is not required [147]. Suppose first that sufficient snapshots of a flow field have been collected such that the subspace described by the image space of the snapshot matrix $X$ is not expanded by adding any further snapshots. If this assumption is approximately correct then the $(N+1)^{\text{th}}$ snapshot, $x_{N+1}$, will be expressible, after a residual, as a linear combination of the previous snapshots

$$x_{N+1} = a_1 x_1 + a_2 x_2 + \cdots + a_N x_N + \nu_{i+1}$$

where $a_i$ are real-valued constants and $\nu_{i+1}$ is a (hopefully small) residual. Based on this assumption, defining a matrix

$$S = \begin{pmatrix} 0 & 0 & \cdots & & a_1 \\ 1 & 0 & \cdots & & a_2 \\ & \ddots & \ddots & & \vdots \\ & & 1 & 0 & a_{N-1} \\ & & & 1 & a_N \end{pmatrix} \in \mathbb{R}^{N \times N}, \tag{2.6}$$

with ones below the leading diagonal acting to exactly shift the first $N-1$ snapshots, it follows that

$$X' = XS + \nu, \tag{2.7}$$

where $\nu$ is a matrix containing the residuals

$$\nu = \begin{pmatrix} \uparrow & \uparrow & & \uparrow & \uparrow \\ 0 & 0 & \cdots & 0 & \nu_{N+1} \\ \downarrow & \downarrow & & \downarrow & \downarrow \end{pmatrix}. \tag{2.8}$$

Due to the fact that $S$ relates snapshots in $X$ to those measured one time-step later in $X'$, it is hoped that analysis of the $N \times N$ matrix $S$, which is significantly lower dimensional that the $p \times p$ matrix $A$ proposed above, can extract dynamical information about the underlying data.

It is appropriate to choose the coefficients $\{a_i\}_{i=1}^N = \{a_1, a_2, \cdots, a_N\}$, and hence also the matrix $S$, by optimising these constants such that the residual $\nu$ is minimised. This can be achieved by performing a $QR$-decomposition of $X$ so that $X = QR$, where $Q$ is an orthonormal matrix and $R$ is an upper triangular matrix, and solving the least squares fit $\min_a \|x_{N+1} - Xa\|_F^2$ to obtain

$$a = R^{-1}Q^{\top}x_{N+1}. \tag{2.9}$$

Now, appealing to the belief that $A \in \mathbb{R}^{p\times p}$ exists for which $X' \approx AX$ and using (2.7),

$$AQR = AX \approx X' = XS + \nu = QRS + \nu,$$

and one would expect that if the residual $\nu$ can be made to be small through the choice of $S$, then

$$S \approx (R^{-1}Q^{\top})A(QR).$$

Consequently, $S$ and $A$ are approximately similar and it then follows that the eigenvalues of $S$ would approximate a subset of the eigenvalues of $A$.

Performing an eigendecomposition $S = P\Lambda P^{-1}$, the DMD modes are defined to be the columns of $\Phi = XP$. What makes this method attractive is the fact that each mode is associated with a single eigenvalue of $S$, and therefore each mode has a corresponding and single temporal frequency and growth rate. The frequency of each mode is extracted using the phase of the eigenvalue and the growth rate is determined by its magnitude. Throughout the current document, we will refer to modes with these characteristics as *single-frequency* modes. In contrast, POD modes are not single-frequency modes, since their temporal coefficients typically contain more than one temporal frequency.

**Linking DMD and Dynamical Systems Theory via Koopman Analysis.** The studies in [146] and [119], provide a theoretical connection between the DMD modes and eigenvalues and the Koopman theory of nonlinear dynamical systems [93]. Koopman theory concerns representing nonlinear dynamical systems, which evolve on a finite-dimensional manifold $\mathcal{Y}$, using a linear but infinite-dimensional operator. Consider the autonomous nonlinear dynamical system

$$y_{j+1} = f(y_j)$$
$$x_j = g(y_j),$$

where $f(\cdot)$ and $g(\cdot)$ are possibly nonlinear functions, $(y_j)_{j\geq 0} \subset \mathcal{Y}$ is the system state and $(x_j)_{j\geq 0}$ is an output of the system measured though the *observable* function $g$. In the context of fluid mechanics, the states of the above system should be viewed in the following way: $y_j$ represents the underlying true state of the flow, for example, the solution of an appropriate PDE (or a high-dimensional approximation to it); while $x_j$ is a measurement that may be taken from the flow, given knowledge of its true underlying state $y_j$; the observable $g$ then represents the process of taking such a measurement, for example, the process of recording a number of velocity components in a given domain of the fluid.

Koopman analysis seeks to represent dynamical systems of this form in terms of a particular representation of the observable process $g$. To do this, we define a linear infinite-dimensional operator $\mathcal{K}$, known as the Koopman operator. This acts on any scalar-valued one-dimensional observable $\gamma : \mathcal{Y} \to \mathbb{R}$ to produce a new observable process $\mathcal{K}\gamma$ defined by

$$(\mathcal{K}\gamma)(y) = \gamma(f(y)) = (\gamma \circ f)(y). \tag{2.10}$$

That is, $\mathcal{K}\gamma$ provides the same measurement as $\gamma$, but only after one time-step of the underlying system dynamics (described by $f$).

Koopman theory seeks to represent dynamical systems in terms of eigenfunctions $\alpha_i(\cdot)$ of the Koopman operator, which are well-defined since $\mathcal{K}$ is linear. Furthermore, since $\mathcal{K}$ is defined entirely in terms of $f$, these eigenfunctions are expected to convey dynamical information relating to the true underlying system. Now, given the Koopman eigenfunctions $\alpha_i(\cdot)$ and their associated eigenvalues $\lambda_i^{\mathcal{K}}$, a vector-valued observable $g : \mathcal{Y} \to \mathbb{R}^p$ which provides a $p$-dimensional measurement (as in our fluids examples) can be expressed as an infinite sum

$$g(y) = \sum_{i=1}^{\infty} \phi_i^{\mathcal{K}} \alpha_i(y)$$

where $\phi_i^{\mathcal{K}} \in \mathbb{R}^p$ are referred to as the *Koopman modes* of the system.

In terms of a sequence of data $(x_j)_{j\geq 0} \subset \mathbb{R}^p$ arising from such an observable, we can then write

$$\begin{aligned} x_j = g(y_j) &= \sum_{i=1}^{\infty} \phi_i^{\mathcal{K}} \alpha_i(y_j) \\ &= \sum_{i=1}^{\infty} \phi_i^{\mathcal{K}} \alpha_i(f(y_{j-1})) \\ \text{(by (2.10))} &= \sum_{i=1}^{\infty} \phi_i^{\mathcal{K}} \left(\mathcal{K}\alpha_i\right)(y_{j-1}) \\ \text{(since } \alpha_i \text{ are eigenfunctions)} &= \sum_{i=1}^{\infty} \phi_i^{\mathcal{K}} \cdot \lambda_i^{\mathcal{K}} \alpha_i(y_{j-1}) \\ &\vdots \\ \text{(repeating the above)} &= \sum_{i=1}^{\infty} \phi_i^{\mathcal{K}} \left(\lambda_i^{\mathcal{K}}\right)^{j-1} \alpha_i(y_1). \end{aligned} \tag{2.11}$$

The underlying hope is that if the underlying system is sparse enough dynamically, i.e. if a small number of Koopman modes/eigenfunctions are dominant dynamically, then the infinite dimensional sum may be well-approximated by a finite dimensional expansion. If this is the case, then one may hope to extract the Koopman modes and eigenvalues, which are intrinsic to the system's dynamics, using only the available measured data ensemble $(x_j)_{j\geq 0}$. This opens the door to a link with DMD. For a discussion about the conditions under which the Koopman operator is finite dimensional the reader is referred to [1].

To explain the link with Arnoldi DMD and the above analysis, note first that the eigen-decomposition of the matrix $S$ yields $S = P\Lambda P^{-1}$, and the particular *companion* form (2.6) of the matrix $S$ implies that if the eigenvalues in $\Lambda$ are distinct, then the matrix $P^{-1}$ has the Vandermonde form

$$P^{-1} = \hat{V} = \begin{pmatrix} 1 & \lambda_1 & \lambda_1^2 & & \lambda_1^{N-1} \\ 1 & \lambda_2 & \lambda_2^2 & \cdots & \lambda_2^{N-1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \lambda_N & \lambda_N^2 & & \lambda_N^{N-1} \end{pmatrix}.$$

Since the DMD modes were defined as $\Phi = XP$, it follows trivially that

$$X = \Phi \hat{V},$$

which can be rewritten in the form

$$x_j = \sum_{i=1}^{N} \phi_i \lambda_i^{j-1}, \qquad \text{for } j = 1, 2, \ldots, N.$$

Note that the first $N$ snapshots are captured exactly. For the last snapshot, $x_{N+1}$, we use the approximation in (2.7) to obtain

$$X' = X(P\Lambda P^{-1}) + \nu = \Phi \Lambda \hat{V} + \nu \tag{2.12}$$

where $\nu$ has the form as in (2.8). Therefore the final snapshot can be expressed as

$$x_{N+1} = \sum_{i=1}^{N} \phi_i \lambda_i^N + \nu_{i+1}, \tag{2.13}$$

where $\nu_{i+1}$ is the residual between the DMD's approximation and the true snapshot. From the similarity of the expansions in (2.12) and (2.13), and (2.11), it is often argued [146] that DMD modes can be considered as a finite set of finite-dimensional modes that approximate the theoretically infinite set of infinite dimensional Koopman modes.

An insightful comparison between the Koopman operator and DMD is provided in [10] via the application of Koopman decomposition to a weakly nonlinear model of the super-critical wake of two-dimensional flow past a circular cylinder, specifically the model developed in [157]. The results of the theoretical Koopman analysis are then compared with those arising from the application of Arnoldi DMD to the same dataset. For this noise-free test-case, it was found that the Koopman eigenvalues belong to harmonically arranged equidistant frequency bins. At each frequency, there are a number of Koopman modes with various damping ratios, in this case, all with negative growth rates. Very

similar results are also obtained when Arnoldi DMD is applied to a set of snapshots of data evolving towards, or fully on, the flow's limit cycle corresponding to the Kármán vortex street. However, if DMD is applied to a larger data set containing snapshots from both limit cycle and transition regimes, the DMD eigenvalues arrange into clusters with slightly varying frequencies around the theoretical Koopman frequencies, and also possess a range of decay rates. This phenomenon was explained by the fact that the full dataset includes a segment where flow is in non-modal algebraic growth, as explained by [151] and observed in other flows [180], that cannot be explained by individual linear oscillators. Since DMD uses such terms as an expansion basis for the underlying data ensemble, more than one mode will be necessary to explain this phenomenon.

Despite strong theoretical backing provided by the link with Koopman theory, the Arnoldi DMD algorithm is known to be numerically unstable as a result of concentrating the residual minimisation process on only the last snapshot $x_{N+1}$ in the data [152]. We now discuss a more stable version of the algorithm, which also motivates further extensions of DMD which will be used later in the thesis.

### 2.3.2 SVD-based DMD

The assumption that $S$ has the companion form (2.6) allows for the exact reconstruction of snapshots $x_2$ to $x_N$ [146]. However, as mentioned, this technique leads to numerical instability. As explained in [33], this instability arises since, in a fully developed flow, the modes with the highest growth rates will eventually dominate. As a result, the snapshots may become nearly linearly dependent. In such a case the inverse matrix $R^{-1}$ in (2.9) will not be well defined.

A solution to this problem is given in [152] which employs an alternative formulation in which the reconstruction residual is spread over all the snapshots. Instead of a $QR$ decomposition, we start with applying POD to $X$ such that $X = U\Sigma V^\top$. In this formulation, the problem of obtaining a linear approximation $A \in \mathbb{R}^{p\times p}$ for the flow's evolution over one time-step is instead expressed as

$$x_{i+1} = Ax_i + \nu_{i+1} = U\hat{A}U^\top x_i + \nu_{i+1},$$

where $\hat{A} \in \mathbb{R}^{N\times N}$ is a lower-dimensional matrix which is similar to $A$, and $\nu_{i+1}$ is the residual of the fit. Solving the unconstrained least squares optimisation problem

$$\underset{\hat{A}}{\text{minimize}} \quad J := \|X' - U\hat{A}U^\top X\|_F^2, \tag{2.14}$$

the optimal matrix $\hat{A}$ can be computed in terms of the SVD matrices and the matrix of latter snapshots $X'$ via

$$\hat{A} = U^\top X' V\Sigma^{-1}. \tag{2.15}$$

As before, since the matrix $\hat{A}$ satisfies $A = U\hat{A}U^\top$, the hope is that its eigenvalues approximate some of the eigenvalues of the "true" matrix $A$.

To summarise, this SVD-based implementation of DMD avoids overfitting by first using the projection $U^\top X$ of the snapshots $X$ onto the $N$ dimensional subspace of $\mathbb{R}^p$ spanned by the POD modes. It then applies the linear dynamics matrix $\hat{A}$ to the projected snapshots. The operation $U : \hat{A}U^\top X \mapsto U\hat{A}U\top X$ then lifts the mapping back to the original $p$ dimensional space of the underlying data ensemble.

Crucially, the same mapping can be used for the mapping of the eigenvectors of $\hat{A}$ to those of $A$. Using the eigendecomposition $\hat{A} = P\Lambda P^{-1}$, where $P$ contains the eigenvectors of $\hat{A}$ and $\Lambda$ contains its discrete-time eigenvalues, the DMD modes in this context, i.e. approximate eigenvectors of $A$, are defined to be

$$\Phi = UP. \tag{2.16}$$

The eigenvalues in the diagonal matrix $\Lambda$ are by design discrete-time and can be converted to the continuous-time DMD eigenvalues using

$$\mu_i = \frac{\log(\lambda_i)}{\Delta t}. \tag{2.17}$$

In practice, it is well-known that SVD-based DMD does not suffer from numerical stability issues. This has led to SVD-based DMD being widely adopted in place of the Arnoldi implementation. However, the Arnoldi method was shown to have a strong connection to Koopman theory, with implications for its applicability in control and estimation problems (see [1] for example). For completeness, therefore, it is desirable to briefly highlight the connection between the SVD-based DMD and Arnoldi-based DMD, and as a result with Koopman operator theory. Considering the decomposition $X = U\Sigma V^\top$, Wynn et al. [174] showed that if both approaches yield the same approximation, i.e. if

$$U\hat{A}U^\top X = XS$$

then one can see that

$$\hat{A} = (\Sigma V^\top)S(V\Sigma^{-1}),$$

and that the two matrices $A$ and $S$ which underpin model fitting in each DMD implementation are similar. An analogous expression to the expansion in (2.12) can therefore be derived for SVD-based DMD. However, in this case, the residual matrix $\nu$ is full and has the general form

$$\nu = \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ \nu_1 & \nu_2 & \cdots & \nu_{N+1} \\ \downarrow & \downarrow & & \downarrow \end{pmatrix}. \tag{2.18}$$

Another difference between the SVD based and the Arnoldi-based DMD is that the SVD-based DMD modes are normalised and therefore the amplitude of the modes must be extracted through a post-processing step such as the one presented in [86]. In §3 we will discuss alternative ways to find appropriate mode amplitudes.

Another useful observation which can be inferred from the presentation of SVD-based DMD is that the DMD modes, given by (2.16), are by definition (complex) linear combinations of the POD modes. This somewhat arbitrary choice, made primarily for simplicity of computation, has been relaxed in a number of extensions of the DMD algorithm, some

of which we discuss in §2.4. Further, DMD arbitrarily chooses the dimension of $\hat{A}$ and the number of DMD modes to be the same as the number of the snapshots in each of the data matrices $X$ and $X'$, i.e. it extracts $N$ coherent structures. We have already mentioned the difficult and at times arbitrary nature of selecting truncation dimensions in POD. More detailed discussions of the effects of choosing an arbitrary number of modes are presented in §2.4 and §3.1.

DMD and SPOD both extract modes associated with a single frequency. While DMD does not specify the frequency at which the modes should be extracted, SPOD finds a hierarchy of modes at a single predefined frequency. The advantage of DMD over SPOD is the automatic manner in which it detects the coherent structures and that it requires fewer snapshots of data to converge [165]. Both of these On the other hand, as discussed, DMD can produce clusters of similar modes, with similar spectral and spatial information. The relation between DMD and SPOD is analysed in more detail in [165], where SPOD modes are described as optimally-averaged DMD modes. In §3 we consider this idea from a different perspective and introduce a method that identifies clusters of DMD modes and finds a hierarchy of coherent structures within each cluster.

Due to its simplicity and effectiveness, DMD has been adopted in a variety of fields in which extracting linear dynamics can be beneficial. Applications of DMD in fluid mechanics have been wide and varied. These include finding dynamic modes of a helium jet [154], analysing the effects of forcing on jets [154, 168], identifying single-frequency modes of a jet in a cross-flow [146], identifying structures in cavity flows [155], the study of stall in airfoils [113], identifying dynamics of streak line instability in wall turbulence [35], structures of pulsatile blood flow [71] and the flow structures around a high-speed train [125]. We note that a recent study in [78] has used DMD for devising a data-driven method for resolvent analysis, which is beyond the scope of the current thesis.

DMD has also enjoyed widespread adoption in other disciplines, such as computer vision [69, 53, 98], finance [112], epidemiology [138], civil engineering [48], neuroscience [29], electricity demand forecasting [122] and robotics [19]. Finally, for an engineering

perspective on the effects of parameters such as sampling rate $\Delta t$ on the results of DMD, the reader is referred to the recent contribution in [101].

## Variants of the DMD Algorithm

DMD, especially the robust SVD-based DMD implementation, is capable of extracting useful dynamical information. Specifically, the single-frequency nature of the modes provides a natural relationship between a coherent structure, captured by the spatial shape of the DMD mode, and physical phenomena with similar temporal frequencies to that of the associated DMD eigenvalue. However, there are many scenarios in which the performance of the DMD algorithm can be improved by applying modifications. The wide range of DMD's applicability in fluid mechanics has led to an increasing interest in developing such variants. Many contributions have concentrated on analysing and improving various aspects of DMD's performance, by first raising and subsequently adapting the algorithm to address its shortcomings.

Some studies have sought to establish best practices for making DMD robust to data collection challenges, e.g. non-sequential data [166], noise-corrupted data [75, 45], and large or streaming datasets [76, 4]. Alternatively, research has focused on generalising the DMD framework for a wide variety of underlying dynamic systems, for example, flows with quasi-periodic behaviour [99], transient characteristics [99, 129], or systems with time-varying dynamics [178]. Methods of extracting a small set of coherent structures from DMD (recalling that in its original form, DMD extract as many structures $N$ as there were snapshots in the original ensemble) have also been advanced in two important ways. Firstly, by allowing the extraction of low-rank models of the flow [39, 174], and secondly by re-scaling the dynamic modes [86]. There has also been a renewed effort in improving DMD's ability to approximate the Koopman operator [171, 6, 94].

In the following, we will discuss some notable examples of each category of DMD variants, before we have a detailed look at the variants most relevant to the methodologies developed in this thesis.

**Robust implementations** Here we will have a look at modifications that make DMD more robust to practical issues such as noise, memory constraints or irregular sampling frequency.

In the discussion about DMD in §2.3, it was assumed that the data was sampled at regular time-steps $\Delta t$. The *Exact DMD algorithm*, proposed in [166], relaxes this assumption. Here, it is shown that for a non-sequential dataset, so long as the snapshots are collected in pairs such that each column of $X'$ is sampled one time-step ahead of the corresponding column in $X$, an optimal set of DMD modes can be found using the same $\hat{A}$ as in (2.15). However, it is shown in [166] that for such cases a different relation between the eigenvectors $P$ of $\hat{A}$ and the DMD modes must be used, resulting in the definition of exact-DMD modes via

$$\phi_i = \lambda_i^{-1} X' V \Sigma^{-1} p_i, \tag{2.19}$$

where $\lambda_i$ and $p_i$ are the $i$th eigenvalue and eigenvector of $\hat{A}$. It is also proven that the exact-DMD modes and eigenvalues are a proper subset of the eigenvalues and eigenvectors of the matrix $A$ in (2.5), specifically one which minimises the residuals $\nu_i$. This is true regardless of the validity of the assumption that a high dimensional operator $A$ can accurately approximate the underlying dynamics.

Importantly, [166] also shows a connection between DMD and older model discovery algorithms such as Eigen-Realisation Algorithm (ERA), and Linear Inverse Modelling. Another important connection was made in [39] where a strong connection between DMD and the Discrete Fourier Transform is shown. The analysis shows that subtracting the mean of the data reduces Arnoldi DMD to the temporal DFT which is restrictive and, as argued previously, generally undesirable. This is in contrast to POD analysis where, to satisfy boundary conditions, the mean flow field must be subtracted as the background flow [128, 127]. On the other hand, it is known that subtracting an equilibrium point generally preserves the DMD spectrum and modes. An optimised version of DMD was also

proposed in [39] that seeks to extract a reduced number of DMD modes and eigenvalues. This optimised DMD algorithm will be further discussed in §2.4.

A significant limitation of the standard DMD algorithm is in its ability to handle noisy datasets. The effects of process and measurement noise on the extracted DMD eigenvalues have been studied in [11, 45]. In particular, [11] use weak noise theory to show that for flow evolving on a limit cycle, such as the 2D cylinder wake at a low Reynolds number, the presence of process noise results in artificial damping of the DMD eigenvalues. This noise-induced damping increases quadratically with the eigenvalue frequency, and linearly with the noise amplitude. Another observation of the effect of noise on DMD results is presented in [45], where an expression for the deterministic effect of noise on the DMD dynamics matrix $\hat{A}$ in (2.15) is derived. This shows that measurement noise has a similar effect on the DMD eigenvalues as that of process noise, studied in [11].

Based on this analysis, [45] presents extensions of DMD that can correct for measurement noise, based on the assumption that the noise-to-signal ratio is small. For a dataset where the noise is normally distributed with mean 0 and variance $\sigma^2$, they proposed the *noise-corrected DMD* algorithm, in which the matrix $\hat{A}$ is first calculated based on the snapshot data contained in $X$ as in (2.15), and the deterministic effect of noise is corrected using the equation

$$\hat{A}_{NC} = \hat{A}(I - N\sigma^2(XX^\top)^{-1}). \tag{2.20}$$

where $\hat{A}_{NC}$ is the noise-corrected dynamics, $N$ is the number of snapshots and $I$ is the identity matrix of appropriate dimensions. Note that the high dimensional nature of $XX^\top$ makes this method computationally prohibitive. They produce a similar, less computationally prohibitive, formula for the cases where the POD modes are truncated before applying DMD, and also propose two methods for cases where the nature of the noise is not known: *Forwards-Backwards DMD* and *Total-Least-Squares DMD*.

We discuss here a version of Total-Least-Squares DMD presented in [75] which is agnostic to the version of DMD which underpins it. The idea behind the algorithm is the

observation that by separating a noisy dataset into matrices $X$ and $X'$ and finding $X$ as a function of $X'$, DMD essentially treats the snapshots in $X$ as noise-free, and assumes all noise is concentrated in $X'$ which is clearly not the case in general. The proposed solution is to find a projection matrix that can de-noise both matrices $X$ and $X'$ simultaneously. To achieve this, it is proposed in [75] that one should stack the matrices $X$ and $X'$ to form the matrix

$$Z = \begin{pmatrix} X \\ X' \end{pmatrix} \tag{2.21}$$

and then perform an SVD on $Z$ instead to obtain $Z = \tilde{U}\tilde{\Sigma}\tilde{V}^\top$. By truncating the temporal POD modes $\tilde{V}$, a projection matrix for unbiased elimination of noise is found. The next step is to use this projection to find the de-noised matrices $\tilde{X} = X\tilde{V}$ and $\tilde{X}' = X'\tilde{V}$. The practitioner can then perform the DMD variant of their choosing on the filtered matrices $\tilde{X}, \tilde{X}'$ to calculate the DMD modes and eigenvalues.

The above algorithms allow for the extraction of previously collected data. A modification presented in [76], allows DMD to be applied to streaming data, and continuously updates the DMD modes and eigenvalues as new snapshots of data become available. Another such method utilises the Full Orthogonalisation Arnoldi method to develop a similar streaming DMD method [4].

**Modifications of DMD Motivated by Dynamical Considerations.** As mentioned previously, much literature has concentrated on modifying the original assumptions of DMD to make them more suitable for different ranges of dynamical systems. It is known that the single-frequency nature of DMD lends itself well to systems which exhibit periodic behaviour, but that it does not perform as well in capturing transient non-oscillatory modes [129]. In such scenarios, the non-oscillatory mode is captured using a cluster of oscillatory modes, as we exemplify later in §3.

For flows with transient and quasi-periodic behaviour, or when the underlying system is temporally broadband, higher-order-DMD [99] provides a vector-auto-regressive extension

of DMD, where each snapshot $x_{i+d}$, is approximated using a linear model of the form

$$x_{i+d} = A_1 x_i + A_2 x_{i+1} + \cdots + A_d x_{i+d-1} + \nu_{i+d}, \qquad \text{for } i = 1, 2, \ldots, (N-d),$$

where $A_j$ are matrices and $d$ is the number of past snapshots used in the prediction of the current time-step. Using the POD modes of the snapshot matrix $X$, the snapshots are first projected onto a lower dimensional space (as in DMD), where a similar equation to the one above is then expressed in matrix form. The resulting matrix is then treated in an analogous manner to the companion DMD matrix (2.6) and decomposed into modes and eigenvalues. The contribution in [129], which allows for more complex models, is also known to improve the performance of DMD for flows during transition. The method has been successfully applied on a laminar supercritical cylinder wake.

The variants above assume that the dynamics of the underlying system are time-invariant. Online DMD [178] modifies the cost function in (2.14) to get

$$J_i \coloneqq \sum_{j=1}^{i} \rho^{i-j} \|x'_j - A_i x_j\|_2^2,$$

where $A_i$ can now capture a slowly-time-varying dynamics matrix at time step $i$ and $0 < \rho \leq 1$ is a constant that determines the level of emphasis on reconstructing the recent states.

Importantly, the above variants concentrate on modelling the natural evolution of an unforced dynamical system. DMD with control (DMDc) [137] is an algorithm designed for data collected in the presence of control inputs. The algorithm allows for simultaneous approximations of a dynamics matrix $A$ and an input matrix $B$ describing the influence of an actuator on the extracted low-order model. We will discuss this algorithm in more detail in §4.1.

**Improved Approximation of the Koopman Operator.** As discussed in §2.3.1, the accuracy of the Koopman approximation, where the finite-dimensional nonlinear dynam-

ics of the system are approximated using a linear operator, is subject to the observable measurements $x$ being rich enough to form an invariant subspace. In theory, the required vector of observables for satisfying this condition can be infinite-dimensional. However, DMD is based on the assumption that a rich, finite-dimensional state can lead to an approximation of the Koopman operator. There have been a number of approaches to finding improved approximations of the Koopman operator (over and above standard DMD) in cases where the measured data is low-dimensional or insufficiently rich to produce such a good approximation.

An example of this approach is the *Extended DMD* algorithm [171] where, instead of applying DMD to measured snapshots $X$ and $X'$, first a predefined dictionary of functions $\Psi(\cdot)$ is applied to the measured states in $X$ and DMD is applied to model the time progression of the transformed matrix of snapshots from $\Psi(X)$ to $\Psi(X')$. For example, for two-dimensional snapshots $x_i \in \mathbb{R}^2$ a quadratic polynomial dictionary could be used to yield the extended snapshot $\psi(x_i)$ as follows:

$$x_i = \begin{pmatrix} a \\ b \end{pmatrix} \longmapsto \psi(x_i) = \begin{pmatrix} a \\ b \\ ab \\ a^2 \\ b^2 \end{pmatrix}.$$

DMD can then solve the optimisation problem

$$\underset{A}{\text{minimize}} \quad \|\Psi(X') - A\Psi(X)\|_F^2, \tag{2.22}$$

where $A$ is a matrix of appropriate dimensions. In this way, higher-order spatio-temporal relations may be searched for within the given data ensemble. It has been shown that for a large enough functional dictionary $\Psi(\cdot)$, the resulting matrix $A$ does indeed converges to the Koopman operator [94], and there have also been efforts to simultaneously extract the

optimal dictionary functions $\Psi(\cdot)$ [103]. A practical limitation of this method, however, is that increasing the order of the underlying functional dictionary rapidly leads to the dimension of $\Psi(X)$ becoming much larger than the dimension of the underlying snapshot data ensemble. Consequently, robust model fitting cannot be achieved in this context, limiting the polynomial order of the dictionary used. This is not so problematic if the dimension of the underlying snapshots is small, a situation which arises if the snapshots are the states of a low-order nonlinear model, such as the Lorenz attractor, which is often used to motivate development of such methods. However, when the underlying data is drawn from a realistic fluid mechanics problem the underlying size of each snapshot $x_i \in \mathbb{R}^p$ is typically large with $p \geq 10^5$. Here, even moving from linear to quadratic dictionaries immediately runs in overfitting problems.

While alternative methods have been proposed recently to address this issue, such as *Kernel DMD* [172] or *Hankel DMD* [6], it is still unclear to what extent this so-called *curse of data-dimensionality* can be adequately resolved in the application of extended-DMD methods. In this thesis, we will be using high dimensional and often noisy experimental data, where the applicability of existing methods is limited. We will explore in §5 the use of higher-order methods for structure extraction and dynamic modelling which both exploit DMD and can be applied to noisy high-dimensional data.

## 2.4 Optimal Mode Decomposition and Low-rank DMD Variants

To-date there have been several attempts to modify the DMD algorithm in order to extract low-rank approximations *of any chosen order* directly from high dimensional data. In §3.1 we will give a brief, yet critical, overview of these techniques and their weaknesses. Here we present a more detailed review of the existing variants.

The first attempt at a low-rank DMD model was presented in [152], where it is shown that a further advantage of SVD-based DMD over the Arnoldi formulation is that it implies a natural truncation that can be used to discard low-energy modes. In particular, the fact that SVD-based DMD explicitly uses the POD matrix $U$ in its optimisation problem formulation (2.14) implies that further, and natural, dimensional reduction can be achieved simply by truncating the POD matrix to retain only higher-energy modes, before solving a reduced version of (2.14). Such an approach assumes that lower-energy modes are associated with measurement noise and incoherent events. This remains the most widely used method for low-rank DMD modelling.

An alternative approach, *Optimised DMD* introduced in [39], is based on the Arnoldi form of DMD where the eigendecomposition of the companion DMD matrix $S = P\Lambda P^{-1}$ form yields the Vandermonde matrix

$$P^{-1} = \hat{V} = \begin{pmatrix} 1 & \lambda_1 & \lambda_1^2 & & \lambda_1^{N-1} \\ 1 & \lambda_2 & \lambda_2^2 & \cdots & \lambda_2^{N-1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \lambda_N & \lambda_N^2 & & \lambda_N^{N-1} \end{pmatrix}.$$

The Koopman approximation of the snapshots $X$ can then be expressed using the DMD modes $\Phi$ and matrix $\hat{V}$ expressed as $X = \Phi\hat{V} + \nu$, where $\nu$ is a matrix of residuals. An optimal $r$-dimensional DMD, would lead to the new Vandermonde matrix

$$\hat{V}_r = \begin{pmatrix} 1 & \lambda_1 & \lambda_1^2 & & \lambda_1^{N-1} \\ 1 & \lambda_2 & \lambda_2^2 & \cdots & \lambda_2^{N-1} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 1 & \lambda_r & \lambda_r^2 & & \lambda_r^{N-1} \end{pmatrix},$$

which can in turn be used in the approximation

$$X = \Phi_r\hat{V}_r + \nu,$$

where $\Phi_r$ is the reduced-order matrix of DMD modes. Since the matrix $\hat{V}_r$ is not invertible, the Moore-Penrose pseudoinverse $\hat{V}_r^+$ can be used to find the optimal modes $\Phi_r = X\hat{V}_r^+$ as a function of the optimal matrix $\hat{V}_r$ and the snapshot matrix $X$. Using this expression, the question of finding the optimal low-order DMD, is reduced to that of finding the optimal Vandermonde matrix that minimises the cost function $J \coloneqq \|X - X\hat{V}_r^+\hat{V}_r\|_F^2$. Unfortunately, at present, there are no computationally efficient algorithms to solve this optimisation problem due to the nonlinearities introduced by the structure of the Vandermonde matrix in the optimal cost.

In particular, the difficulty is that one must work with the Vandermonde form of a matrix, and as a result the problem should be formulated as one of finding $r$ eigenvalues $\{\lambda_i\}_{i=1}^r = \{\lambda_1, \lambda_2, \cdots, \lambda_r\}$ that minimise the cost function $J$. Nonetheless, in cases where a solution is available, the results of Optimised DMD have been shown to outperform the original DMD method [163]. An improved version of the algorithm has been proposed recently that calculates a range of optimised-DMD models and averages them to find more robust models [148].

Throughout this thesis, we will make extensive use of the Optimal Mode Decomposition (OMD) algorithm [174]. We will use this algorithm both for the decomposition of challenging datasets, and also extend it to present a new decomposition technique in §4 which seeks to prioritise measurement processes in the computation of coherent structures and reduced-order models. Here, we will first present OMD as an algorithm for extracting a low-order DMD-like dynamic model. Once we have presented the algorithm in detail, we will comment on its potential as a general method for finding dynamically relevant projections of high-dimensional data.

OMD does not assume, as in DMD, that the POD modes of the snapshots provide the optimal projection matrix for finding a low-order linear model. Instead, OMD finds the optimal projection matrix and the reduced order dynamics matrix simultaneously, in the sense that it solves the optimisation problem

$$\begin{aligned} \underset{L,M}{\text{minimize}} \quad & \|X' - LML^\top X\|_F^2 \\ \text{s.t.} \quad & L^\top L = I, \\ & M \in \mathbb{R}^{r\times r},\ L \in \mathbb{R}^{p\times r} \end{aligned} \tag{2.23}$$

where $r \in \mathbb{N}$ is the rank of the matrix $LML^\top$ fitting the evolution of the flow over one time-step and is a parameter prescribed by the user. The matrix $L \in \mathbb{R}^{p\times r}$ is an optimisation variable representing the projection between the high dimensional snapshots and their $r$ dimensional representation $L^\top X$.

To solve the optimisation problem 2.23, the cost function is first differentiated, as is shown in [174], with respect to the matrix $M$ to give

$$\frac{\partial \|X' - LML^\top X\|_F^2}{\partial L} = 2(LXX^\top LM - L^\top XX'^\top L).$$

By equating the value of the derivative to zero, an expression for the optimal low-order dynamics matrix $M$ can be obtained given any fixed projection matrix $L$. In particular, that $M(L)$ is given by

$$M(L) = L^\top X'X^\top L(L^\top XX^\top L)^{-1}. \tag{2.24}$$

Note that for $L = U$, where $U$ is the matrix of POD modes of the snapshot matrix $X$, the optimal matrix $M$ is equal to the DMD dynamics matrix in (2.15). In other words, DMD is a special case of OMD where $L = U$.

With this expression in hand, the OMD optimisation problem (2.23) is transformed into a single-variable optimisation over $L$, by substituting (2.24) in (2.23). Applying the identity $\|A\|_F^2 = \mathrm{tr}(AA^\top)$ to the resulting cost function one can further simplify the optimisation problem to

$$\begin{aligned} \underset{L}{\text{maximize}} \quad & g(L) := \|L^\top X' Q(L)\|_F^2. \\ \text{s.t.} \quad & L^\top L = I, \quad L \in \mathbb{R}^{p\times r}, \\ & Q(L) = X^\top L (L^\top X X^\top L)^{-1} L^\top X. \end{aligned} \tag{2.25}$$

Here we must introduce the Grasman manifold $G_{r,p}$, which is defined as the manifold of all $r$-dimensional subspaces of the $p$-dimensional space. An involved, yet trivial, calculation implies that for any orthogonal transformation matrix $R$, the pairs $(L, M)$ and $(LR, R^\top MR)$ satisfy

$$g(L) = g(LR).$$

In view of the fact that $\mathrm{Im}(L) = \mathrm{Im}(LR)$ for any orthogonal matrix $R$, this implies that the OMD cost function is equivalent for any matrix $L$ whose columns span the same $r$-dimensional subspace of $\mathbb{R}^p$. Thus, OMD effectively searches over the Grassman manifold $G_{r,p}$ to find an optimal $r$-dimensional subspace of $\mathbb{R}^p$ upon which to model the dynamics observed in the underlying, high dimensional, data ensemble.

The optimisation problem (2.25) can be solved using an iterative conjugate-gradient method [65] which takes advantage of the fact that each element of the manifold can be represented by infinitely many orthonormal matrices $L$. This algorithm is presented in more detail in §4, where a new variant of the algorithm is developed.

It should be noted that the function $-g(L)$ is in general nonconvex and therefore the conjugate gradient-based algorithm is not guaranteed to converge to a globally optimal solution. However, in a sense, the OMD algorithm necessarily improves on DMD. The reason is that OMD initialises the matrix $L$ with an appropriately truncated set of POD modes and, as explained above, DMD is a special case of OMD where the projection matrix $L$ is fixed to be the matrix of POD modes. Hence, if the conjugate gradient implementation of OMD makes a step to decrease $-g(L)$, then this cost will be lower than that of DMD, meaning that even if a local minimum is found, this can be viewed as an improvement on DMD.

Once the optimisation problem (2.25) has been solved, the OMD modes and eigenvalues are defined in a similar fashion to DMD. The eigendecomposition of matrix $M$ gives

$$M = P\Lambda P^{-1}$$

and the modes (columns of $\Phi$) and eigenvalues are defined by

$$\Phi = LP, \qquad \mu_i = \frac{\log(\lambda_i)}{\Delta t},\ i = 1, \ldots, r. \tag{2.26}$$

As this discussion shows, OMD does not rely on POD modes for projecting the dataset onto a low-dimensional space. By finding the optimal dynamics matrix $M$ as a function of the projection matrix $L$, the algorithm defines the problem of finding a low-order dynamics model to one of finding a suitable linear projection. The true potential of this algorithm is its versatility in taking dynamical considerations into account while simultaneously finding low-order projections.

A situation in which the extra flexibility is beneficial is considered in §4. While extracting DMD modes from high-dimensional snapshots of a flow field can help the practitioner model the flow's dynamics, it is often the case that one would like to *estimate* the current state of the flow using only a very limited number of measurements (e.g. those which could realistically be sampled in real-time such as hotwire probes). In §4, a new algorithm is developed which builds upon the fact that the projection matrix $L$ can be viewed as an optimisation parameter in order to balance (i) the ability of the extracted model to reconstruct the dynamics of the underlying flow with (ii) its ability to capture structures which are observable from the given measurable output.

Since the development of the original OMD method, there have been many extensions, often inspired by Koopman theory, which also seek to replace the POD projection operator $U$ with more advanced projection techniques [160]. The majority of these methods use nonlinear mapping techniques, and often neural networks in particular, to find a reduced-

order state that follows a linear temporal progression. In the next section, we will review some of these methods as well as the necessary mathematical tools used in the design of neural network models before proposing a new application of this approach in §5.

## 2.5 Neural Networks for Data-Driven Model Discovery

In this section, we review the use of neural networks (NNs) in dimensionality reduction and reduced-order modelling of fluid flows. To understand why NNs are finding increasing application in data-driven fluid mechanics, in §2.5.1 we will review some of the mathematical tools and techniques that are fundamental to NN models. We will also introduce the necessary terminology to develop the modelling approach proposed later in §5, but refer the reader to [100, 64, 23], and [31] for an in-depth discussion of Neural Networks.

### 2.5.1 An Overview of Neural Networks

Neural networks are simply a class of, generally nonlinear, mapping functions. Despite the wide variety of techniques and architectures that fall under the broad category of a neural network, one must always keep in mind that, in essence, a neural network is simply a mathematical function, like any other, with an input and an output.

Depending on the specifics of a problem, the input $x$ of the neural network may be a scalar, a vector or a tensor of any dimensions. In the following discussion, we assume that the inputs and outputs are vectors.

The building blocks of a neural network are called *layers*. Each layer is a function whose input is the output of the previous layer. The output of the final layer is then the output of the network. In mathematical terms, a neural network maps its input $x$ to an output $y$ through the composition of several layers of functions $f_i(\cdot)$ such that

$$y = f(x) = (f_m \circ f_{m-1} \circ f_{m-2} \circ \cdots f_1)(x). \tag{2.27}$$

The parameter $m$ quantifies the number of function compositions and is referred to as the *depth* of the network. For $1 \leq i < m$, the outputs of the functions $f_i(\cdot)$ are referred to as the *hidden layers*. The functions $f_i(\cdot)$ are, in general, vector-valued and the dimension of the output vector at each layer is referred to as the dimension of the $i^{\text{th}}$ hidden layer.

What motivates the recent surge of interest in the use of neural networks, particularly in the context of data-driven model discovery, is the so so called *universal approximation theorem* [82], which states that any function may be approximated by a sufficiently large and deep neural network. To explain this theorem in more precise terms we must first introduce the notion of *fully connected layers*, which are the simplest type of NN.

**Fully connected layers.** A fully-connected layer has two simple building blocks: matrix multiplication and element-wise, typically nonlinear, functions referred to as *activation functions*. An illustrative example is of a one-layer fully connected network given, for an input $x \in \mathbb{R}^p$, by

$$f_1(x) = \boldsymbol{\sigma}(W_1 x + b_1) = \begin{pmatrix} \sigma(z_1) \\ \sigma(z_2) \\ \vdots \\ \sigma(z_n) \end{pmatrix}, \qquad z = W_1 x + b_1 = (z_i)_{i=1}^n \tag{2.28}$$

where $\boldsymbol{\sigma} : \mathbb{R}^p \to \mathbb{R}^n$ is a nonlinear *activation function*, which acts equivalently on each of its $n$ elements via a scalar-valued function $\sigma : \mathbb{R} \to \mathbb{R}$. Here, $n$ is the dimension of the layer, $b_1 \in \mathbb{R}^n$ is a vector of constants referred to as the *bias* vector, and $z$ is the output of the linear function $z = W_1 x + b_1$ where $W \in \mathbb{R}^{n \times p}$ is a matrix of constant weights. Activation functions can take many forms, with a common choice being the rectified linear function

$$\sigma(\xi) = \begin{cases} 0 & \xi < 0, \\ \xi & \xi \geq 0, \end{cases}$$

which introduces structured nonlinearity into the network.

A fully connected NN with only one hidden layer is referred to as a *shallow network*. A deeper, fully connected, network is made of many fully connected layers of the form (2.28) arranged in composition as in (2.27), where the output of the previous layer is used as the input of the current layer.

The universal approximation theorem states that for an arbitrary function $g : \mathbb{R}^p \to \mathbb{R}^r$, there exists a one-layer neural network of the form (2.28), possibly with a large internal dimension $n$, such that the neural network can approximate $g$ to an arbitrary level of accuracy. Specifically, there exists $n \in \mathbb{N}$, weights $W_1 \in \mathbb{R}^{n \times p}$ and biases $b_1 \in \mathbb{R}^n$ and a further linear operation $W_2 \in \mathbb{R}^{r \times n}$ such that

$$g(x) = W_2 f_1(x) + \nu(x).$$

and for which the fitting residual $\nu(x)$ can be made arbitrarily small on any compact subsets of the input parameter $x \in \mathbb{R}^p$ [82]. This, however, does not imply that finding the matrices $W_1$ and $W_2$ and the activation function $\sigma(\cdot)$ for an arbitrarily small fitting error is trivial.

In practice, in most cases increasing the depth of the model, by repeating the composing the same building blocks often leads to better approximations than increasing the complexity (i.e., increasing $n$) of a single layer. A more recent contribution has proven the universal approximation theorem for NNs with arbitrary depth [107]. For completeness, here, we expand a typical deep fully connected network in terms of its weights and biases such that

$$f(x) = h(W_m(\cdots \sigma(W_2(\sigma(W_1 x + b_1)) + b_2) \cdots) + b_m)$$

where $W_i$ and $b_i$ denote the weights and biases of each layer and $\sigma$ is a nonlinear activation function. Note that the function $h(\cdot)$ at the last layer can be specifically designed for a certain task. For regression models, which are of interest to us, function $h(\cdot)$ is often a linear matrix multiplication.

It is important to note that the number of optimisation parameters in the model increases with the depth of the NN and also with the dimension of the hidden layers. Fully connected networks are therefore not typically suitable for processing high-dimensional data such as images or fluid flow fields, due to the high number of optimisation parameters that such an aim would require.

**Convolutional Neural Layers.** Convolutional neural layers are often used in conjunction with *pooling layers*, explained below, as means of reducing the dimensions of a high dimensional input before a specific task, such as regression or classification, is carried out using the output of the convolutional layers. Convolutional layers can achieve dimensionality reduction with significantly fewer parameters than fully connected networks.

The main building block of a convolutional layer is a cross-correlation operation, essentially a discrete convolution operation. For a vector $x \in \mathbb{R}^p$, the $i^{\text{th}}$ element of the cross-correlation vector with a filter $W = (w_i)_{i=1}^n$ is defined by

$$(x \star W)_i = \sum_{m=0}^{n} x_{i+m} w_m,$$

where $\star$ is the notation used for cross-correlation. In a similar way to a fully connected network, an element-wise nonlinear activation function is then applied to the outputs of the cross-correlation operation such that

$$\boldsymbol{\sigma}(x \star W + b) = \begin{pmatrix} \sigma((x \star W)_1 + b_1) \\ \sigma((x \star W)_2 + b_2) \\ \vdots \\ \sigma((x \star W)_{p-n+1} + b_{p-n+1}) \end{pmatrix},$$

where $b$ is a vector of constant biases.

The cross-correlation operation is often accompanied by a so-called *pooling* operation. An example of a pooling operation is the max-pooling operation where the input vector is divided into blocks of length $\ell$ and the element with the highest values in each block $i$ is defined to be the $i^{\text{th}}$ element of the output vector. In each element of the output of a max-pooling operation, can be expressed as

$$(h(\sigma(x \star W + b)))_i = \max(x_j, x_{j+1}, \cdots, x_{j+\ell-1}) \quad \text{for } j = (i-1)\ell + 1.$$

Note that other pooling operations can be applied in a similar way, e.g. averaging of the elements within each block. An illustrative example of the operations above is presented in §A.2

The combination of the cross-correlation operation and the pooling operation is especially useful in cases where the neighbouring elements of the input vector are highly correlated and it is reasonable to assume that a smaller number of elements can represent the information within each local block of the input vector. An example of an input that would satisfy this criterion is an image with spatially coherent structures, where pixels in a local region can be highly correlated [23]. We note that to maintain an intuitive understanding of the local regions in an image, it is often represented as a tensor. In a similar way to the case of fully connected networks, the depth of the network can be increased by using the output of one layer as the input to another convolutional layer. Typically, convolutional layers achieve dimensionality reduction with fewer optimisable parameters than fully connected networks, making them an appealing technique to work with high dimensional data, such as images [96, 73].

The main application of neural networks in this thesis, is in analysing modal time series. We now discuss a class of neural networks specifically designed to process temporal data.

### 2.5.2 Recurrent neural networks

If one needs to model the relationship between an output and a sequence of time dependent inputs, fully connected or convolutional networks require the input sequence to be processed simultaneously. That is, there will be a unique weight and bias corresponding to the same quantity at each time step. As a result, finding a mapping between a long input sequence and an output using fully connected or convolutional NNs will require a large number of optimisation parameters. An alternative approach is to exploit the sequential nature of the input to reduce the number of parameters needed [64].

Recurrent neural networks, RNNs, are a class of neural networks designed for processing time series. In a similar fashion to a discrete-time dynamical system, RNNs apply the same optimisable weights to each snapshot of the sequence. We will give a detailed review of RNNs in §5.2.1, where we employ them for order-reduction of modal time series. Here, we give an overview of RNNs' and argue their benefit over more traditional time-series analysis tools, and over other NN frameworks.

RNN's have close connections to discrete time dynamical systems. We first describe the concept of an *Elman Neural Network* [50], defined analytically by the expression

$$\begin{aligned} h(t_j) &= \tanh(W_{hi}a(t_j) + W_{hh}h(t_{j-1}) + d_h), \\ b(t_j) &= W_o h(t_j) + d_o, \end{aligned} \tag{2.29}$$

that maps the input $a(t)$ to the output $b(t)$. The optimisable weights $W_{hi}$, $W_{hh}$, and $W_o$ are matrices of appropriate dimensions and are identified through training the network on a training data ensemble. Note that if the activation function $\tanh(\cdot)$ is replaced by the linear activation function $\sigma(x) = x$, the equations become equivalent to those of a linear discrete-time system with the state $h$, input $a$ and the measurable output $b$. Similar to the use of a rectified linear function described above, the use of the hyperbolic tangent as an acquisition function, subtly, introduces nonlinearity into the model fit.

Note that the introduction of the nonlinear activation functions allows the model to appeal to the universal approximation theorem and thus be able to replicate any given optimal mapping. At the same time, the sequential nature of the model, applying the same mapping functions to repeat at each time-step, potentially lowers the number of optimisable parameters. However, as will be further explained in §5.2.1, the recursive multiplication $W_{hh}h(t_{j-1})$ in the Elman model presented here causes difficulties in training the neural network since the underlying optimisation parameters enter the problem in a highly nonlinear manner rendering the underlying problem non-convex. Alternative RNN models have been developed to address this issue [80].

Having explained three of the most notable variants of neural networks, we can now look at a modelling architecture that is typically used in nonlinear order reduction, which will be of fundamental interest to our study of low-order modelling of fluid flows.

### 2.5.3 Autoencoders and nonlinear dimensionality reduction

Autoencoders are models designed to find a low-order representation of a higher-dimensional input. This can then be used as a simpler surrogate approximation to the input vector, in an analogous manner to how the POD mode amplitudes enable a lower-dimensional encoding of a high-dimensional fluid flow field.

In mathematical terms, for an input ensemble $(x_i)_{i=1}^N \subset \mathbb{R}^p$, an autoencoder seeks to minimise the cost

$$\sum_{i=1}^{N} \left\| x_i - (g \circ f)(x_i) \right\|_F^2,$$

by search over an appropriate space of, generally nonlinear, functions $f : \mathbb{R}^p \mapsto \mathbb{R}^r$ and $g : \mathbb{R}^r \mapsto \mathbb{R}^p$. The function $f(\cdot)$ is known as the encoder function and maps the (large) size of the input variables $x$ to a user defined reduced-order dimension $1 \leq r < p$. The function $g(\cdot)$ is referred to as the decoder function, which is used to recreate a high-dimensional vector, given input values in the low-order space $\mathbb{R}^r$. In typical applications, both $f$ and

$g$ are chosen to themselves be of one of the Neural Network classes introduced in previous sections.

An interesting link to the linear dimensionality reduction techniques introduced in §2 is the equivalence of shallow linear neural networks with POD [14]. If functions $f$ and $g$ are assumed to be shallow fully connected neural networks, with a *linear* activation function $\sigma(x) = x$, the optimal linear autoencoder is equivalent to projecting the input onto its POD modes. The introduction of nonlinear activation functions can then allow the model to capture more complex relations by appealing to the universal approximation theorem.

### 2.5.4 Neural networks for order reduction in fluids

There are currently two popular approaches in ROM of fluids that utilise NNs. The first approach involves the use of NNs to find a coordinate transformation that maps each snapshot of collected data to a vector that evolves linearly in time [160, 176, 110]. This approach is inspired by the connections between the Extended-DMD algorithm [171] and the Koopman operator. However, in extended-DMD a large set of functional dictionaries are predefined or learnt [172, 103]. Using an autoencoder NN, the resulting coordinate transformations in these methods amount to automatically constructing a finite set of dictionary functions used in Extended-DMD [134]. There have also been implementations that make use of the DMD with control [137] algorithm to devise linear models with control inputs [124]. However, the constraint that the temporal evolution is linear makes these models difficult to train and in most cases unsuitable for noisy and high dimensional data. The approach has been more successfully applied in linearising low dimensional PDEs, such as a nonlinear pendulum with a continuous spectrum, a low dimensional model [110], Burgers' equation, the heat equation and the Kuramoto–Sivashinsky equation [61].

A second approach is that of reducing the order of the snapshots by treating each snapshot as an image and using well-established techniques to find patterns in pictures. We have mentioned that convolutional networks are suitable for processing inputs that are arranged as images. An example of such data can be snapshots of a flow field. In

recent years, there have been many studies that have used convolutional autoencoders to find low-order representations of flows. The difficulty in using convolutional networks for dimensionality reduction in fluids is, as explained in §2, that the snapshots of a fluid flow often consist of spatially overlapping physically significant events.

It has been shown that, given a large enough training-set and a large enough set of convolutional filters, a convolutional NN can be used to reconstruct and forecast snapshots of a flow over an airfoil [133] or a circular cylinder [57, 126], with higher precision than that found using linear compression techniques such as POD. The extracted patterns, however, do not correspond well with the well-known and physically intuitive flow structures that linear techniques extract. In many cases, further analysis of the extracted features shows that they include elements of multiple linear modes. Nevertheless, when used in conjunction with RNNs, the extracted low-order models perform very well in reconstruction and forecasting tasks. Datasets analysed by a combination of a convolutional autoencoder and an RNN-based evolution method were obtained using the supercritical laminar flow over a cylinder, flow over an oscillating airfoil [49], shallow water equations [116], Burgers' equations [116, 62], and the supercritical cylinder wake [49]. A study has also shown that a generalisable model can be developed that encodes and predicts the wake of a range of arbitrarily shaped cylinders [72].

### 2.5.5 Optimisation Algorithms for Training Neural Networks

So far, we have considered the theoretical ability of NNs to model complex mappings and some of the building blocks used to create NN models. However, a significant practical limitation to the use of NNs is the fact that their cost functions are highly non-convex and thus have multiple local minima [64]. While this is not the case for shallow autoencoders with linear activation functions [14], once the move to nonlinear modelling is made (the entire point of NNs is their potential as universal approximators) this necessarily introduces local minima.

The existence of local minimum is only a problem when the local minima have a significantly higher cost than the global minimum. However, more recent studies indicate that for a large enough neural network, all local minima of a typical cost function have values close to the global minimum [149, 44, 63].

Other challenges in optimising a neural network are saddle points and plateaus. The study in [44] points out that saddle points are increasingly likely as the dimensions of parameter space increase since in high dimensions, the chance that all the directions around a critical point lead upward (positive curvature) is exponentially small. The contribution in [63] shows that, for canonical benchmark examples, there exists a large region of parameter space where the cost function is flat with a small (or zero) gradient. Consequently, this creates difficulties for the use of classical gradient-based optimisation algorithms. On the other hand algorithms such as Stochastic Gradient Descent, are known to be capable of escaping these regions [63]. This is the reason behind the widespread adoption of gradient-based optimisation algorithms in training neural networks. Since we use a gradient-descent-based algorithm in §5, we now introduce some of the necessary terminology related to gradient descent optimisation in the training of NNs.

In gradient descent optimisation, the optimisable parameters $W$ are iteratively updated by finding the gradient $\frac{\partial J}{\partial W}$, of the cost function $J$ with respect to the NN parameters $W$, and then subsequently updating the parameter values according to

$$W^{(i+1)} = W^{(i)} - \alpha \frac{\partial J}{\partial W}(W_i),$$

where $\alpha > 0$ is a constant known as the *learning rate*.

There are three major difficulties faced when applying gradient-descent algorithms to NN optimisation problems. The first is that of excessive gradients, arising from high sensitivity of the cost function $J$ to some optimisation parameters. If this is the case then, unless the learning rate $\alpha$ is sufficiently small, the optimisation strategy may miss a minimum. The second challenge is presented by regions of parameter space for which

the cost function is flat or has small gradients. Here, the update term $\alpha \frac{\partial J}{\partial W}(W_i)$ will be small and therefore lead to a sluggish process for an iterative optimiser.

There are variants of the gradient descent algorithm that, to an extent, are able to address these issues. The approach employed in this thesis is the ADAM variant [91]. ADAM stands for Adaptive Moments, and addresses the issue of slow convergence by rescaling the learning rate for directions with a low absolute value of gradient, while also correcting for any errors associated with initialisation.

The final major challenge associated with gradient descent for NN optimisation is that of *vanishing gradients*, this is particularly true in models with deep computational graphs, such as RNNs with a long sequence of inputs or outputs. Although the name vanishing gradients may cause a confusion with the problem of flat regions in the cost function, this problem refers to the difficulty of accounting for earlier layers of a computational graph in updating the model parameters. The problem arises in RNNs where the same parameters are used to process a long sequence of inputs to find the output of the model. It can be shown that gradient descent will de-emphasise the information from earlier inputs in updating the optimisable parameters. This issue is discussed in more detail in §5.2.1.

## 2.6 Modal decomposition techniques for flow control

Having discussed many methods of finding reduced order modelling of the flow, we will now review notable cases where reduced order models have been used for the control and estimation of fluid flows.

Before reviewing the contributions in the literature, we need to point out that methods introduced so far have, for the most part, concentrated on extracting coherent features that can be used in finding ROMs for a flow's temporal evolution. In the context of a discrete-time dynamic system of the form (2.30), the extracted modes and time series from POD or DMD can help the practitioner find a dynamics matrix $A$. The challenge of active control of a flow's behaviour is more challenging in that it also requires a model

for how a control input $u_i$ would affect the behaviour of the flow. In the context of the dynamical system shown in 2.30, one would need a model for the input matrix $B$ and, if a measurement signal $y_i$ is to be modelled one also requires a model for the output matrix $C$.

$$\begin{aligned} x_{i+1} &= Ax_i + Bu_i \\ y_i &= Cx_i \end{aligned} \tag{2.30}$$

We will now mention some notable cases where ROM techniques have been used to identify such models and underpin control and estimation methods in fluid mechanics. In doing so, we try to mention how a control strategy was devised based on the ROM. We note that this is by no means an exhaustive list of the applications of modal decomposition in flow control, but rather is indicative of approaches to the use of modal ROM for flow control.

POD modes have been successfully used to guide control strategies for cylinder wakes in the laminar regime in conjunction with neural networks. This approach avoids the need to derive an explicit calculation of the effect of the control on the flow field [60]. A similar approach was used in [156]. Explicit optimal control strategies have also been derived, where one wishes to suppress POD mode amplitudes [67] with minimal actuation. The inclusion of both velocity and pressure POD modes, as well as shift modes, have also enabled robust explicit control strategies [22, 21]. Numerical results show the potential of using POD models for the stabilisation of a cylinder wake in the turbulent regime [181]. Another application of POD-based models is the open-loop determination of the relative importance of each control parameter. An example of such analysis is given in [54] where the dominant control parameters for the wake of a cylinder controlled by a synthetic jet were determined. POD has also been successfully used to estimate a flow-field based on pressure readings [161]. We note that variants of POD have also been used for active flow control. For example, the study in [12] gives a thorough account of the use of BPOD in active control.

DMD, for the most part, remains a tool for analysis and identification of coherent structures. However, there are variants of DMD specifically designed to extract models of the form (2.30), notably the *DMD-with-control* algorithm [137]. This technique has been used to extract models of the effects of vaccination in infectious disease control and, recently, a modification of the DMD-with-control algorithm was used to analyse blood flow with the pulsatile flow-rate as a control input [71].

DMD has also been used to analyse the effect of control inputs on the output of a dynamical system. Examples include the effects of forcing on jets [154, 168] and detonation waves [115].

Chapter 3

# Data-driven feature identification and sparse representation of turbulent flows

Identifying coherent structures in fluid flows is of great importance for reduced order modelling and flow control. However, extracting such structures from experimental or numerical data obtained from a turbulent flow can be challenging. A number of modal decomposition algorithms have been proposed in recent years which decompose time-resolved snapshots of data into spatial modes, each associated with a single frequency and growth-rate. Most prominently among them is dynamic mode decomposition (DMD). However, DMD-like algorithms create an arbitrary number of modes. It is common practice to then choose a smaller subset of these modes, for the purpose of model reduction and analysis, based on some measure of significance. In this work, we present a method of post-processing DMD modes for extracting a small number of dynamically relevant modes. We achieve this through an iterative approach based on the graph-theoretic notion of maximal cliques to identify clusters of modes, before representing each cluster with a single representative mode.

## 3.1 Introduction

We have discussed the role of coherent structures in our understanding of fluid flows. As presented in §2, modal decomposition is often used in data-driven fluid dynamics to obtain coherent structures. At the heart of most modal decomposition methodologies is the assumption that the flow field $\mathbf{u}(\mathbf{x}, t)$, where $t \geq 0$ is time and $\mathbf{x}$ is the spatial variable, can be approximated by an expansion of the form

$$\mathbf{u}(\mathbf{x}, t) = \sum_i a_i(t)\phi_i(\mathbf{x}),$$

where $a_i(t)$ is the time series associated with the $i^{\text{th}}$ spatial mode $\phi_i(\mathbf{x})$. These modes are candidates for the flow's coherent structures.

A number of modal decomposition algorithms have been developed in the past decade where the time series associated with each mode is assumed to have the form $a_i(t_j) = \alpha_i \lambda_i^{j-1}$, where $(t_j)_{j\geq 0}$ is a discrete-time series and $\alpha_i, \lambda_i \in \mathbb{C}$. An important benefit of this assumption is that each spatial mode $\phi_i(\mathbf{x})$ is associated with a single frequency and growth rate and, subsequently, we refer to this class of algorithms as *single-frequency* methods. We have reviewed these methods in detail in §2.3.

For completeness, we will briefly revisit the SVD-based dynamic mode decomposition (DMD), proposed in [152], and which uses an ensemble of time-resolved snapshots to find spatial modes.

In particular, given an ensemble $\mathcal{X} = \{x_i\}_{i=1}^{N+1} \subset \mathbb{R}^p$ of $N+1$ snapshots sampled at a common time-step $\Delta t$, one can construct matrices

$$X = \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ x_1 & x_2 & \cdots & x_N \\ \downarrow & \downarrow & & \downarrow \end{pmatrix}, \quad X' = \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ x_2 & x_3 & \cdots & x_{N+1} \\ \downarrow & \downarrow & & \downarrow \end{pmatrix},$$

where each column of the matrix $X'$ is one time-step ahead of the same column in matrix $X$. DMD approximates the dynamics of the observed data by constructing a model of

the form

$$x_{i+1} \approx U\tilde{A}U^\top x_i \tag{3.1}$$

over the sampling time-step $\Delta t$. Here, $U \in \mathbb{R}^{p\times N}$ is taken from the singular value decomposition $X = U\Sigma V^\top$, while $\tilde{A} \in \mathbb{R}^{N\times N}$ is the solution of the optimization problem

$$\tilde{A} = \text{argmin}_A \left\| X' - UAU^\top X \right\|_F^2 = U^\top X' V\Sigma^{-1}, \tag{3.2}$$

and $\|\cdot\|_F$ denotes the Frobenius norm.

To extract the DMD modes from this model, let $\tilde{A} = P\Lambda P^{-1}$ be an eigendecomposition of $\tilde{A}$ with eigenvectors $p_i$ and eigenvalues contained in $\Lambda = \text{diag}(\lambda_i)$. Since $U\tilde{A}U^\top$ approximates the observed evolution over a time-step $\Delta t$, the DMD modes $\phi_i$ are defined by $\phi_i = Up_i$, each of which is associated with a DMD eigenvalue $\mu_i = \log(\lambda_i)/\Delta t$.

It is worth reiterating that, as was explained in §2.3, in many cases the assumptions upon which the standard DMD algorithm is based must be adjusted. Many variants of DMD seek to improve on specific aspects of the standard implementation addressing, for example, noisy data, irregular sampling, streaming snapshots, and online implementation [166, 75, 76, 178].

In its standard implementation DMD obtains as many modes as the number of snapshots $N$ and, for typical applications this is large: $N \sim \mathcal{O}(10^{2-3})$. There are, however, systematic ways of calculating a smaller number of single-frequency modes. The simplest, and perhaps most natural, is to replace $U$ in (3.2) with the truncated matrix $U_r \in \mathbb{R}^{p\times r}$ whose columns are the first $r$ singular vectors of $X$, and to replace $\Sigma$ and $V$ by analogously truncated matrices to form $\tilde{A}_r = U_r^\top X' V_r \Sigma_r^{-1}$. In this case, only $r$ modes

$$\{\phi_i\}_{i=1}^r =: DMD(\mathcal{X}, r)$$

are obtained, using the eigendecomposition of $\tilde{A}_r$. Many variants of DMD, including exact DMD, DMD$_{tls}$ and streaming DMD, also use rank reduction as a way to reduce the number of resulting DMD modes.

In a similar spirit, optimal mode decomposition, OMD, achieves the goal of finding a small set of modes by solving a rank-constrained optimization problem corresponding to (3.2) [174]. In particular, OMD minimizes $\|X' - LML^\top X\|_F^2$ over both a rank-$r$ projection matrix $L$ (playing an analogous, but generalised, role to the POD modes) and a low-order dynamics matrix $M \in \mathbb{R}^{r\times r}$. Modes are then extracted in an analogous manner to DMD.

In practice, for any method which reduces the number of modes, one must choose an arbitrary value for the mode-reduction parameter $r$. One approach is to extract modes for a range of values of $r$ and analyse the sensitivity of the resulting decomposition. This was followed, using OMD, in [13] for flow past a multi-scale array. However, if the practitioner does not have any prior knowledge of the number of dominant features present in the flow, such analysis may be challenging. Furthermore, dramatically reducing the truncation dimension can have the effect of modifying the spectral information associated with each mode. Figure 3.1(a-b) show the discrete-time eigenvalues $\lambda_i$ for a typical DMD implementation (specifically for flow past an axisymmetric bluff body studied in §3.3). Although, as [45] explains, some of the mode damping ratio is due to measurement noise, Figure 3.1(a-b) indicates that the mode damping ratio increases as $r$ decreases. Such spurious damping is less pronounced when OMD is applied in Figure 3.1(c-d), but is nonetheless clearly present in the output of both algorithms.

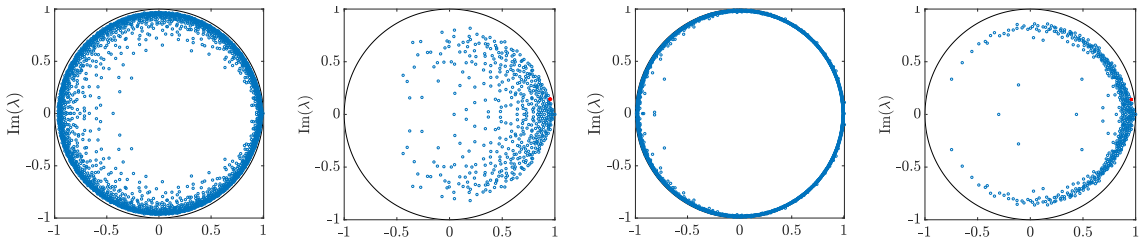



Figure 3.1: Discrete-time eigenvalues of DMD and OMD when applied to data for flow past an axisymmetric bluff body (described in §3.3) at different choices of the mode reduction parameter $r$. a) r = 2500, b) r = 500, and OMD eigenvalues at b) r = 2500 and d) r = 500.

In other words, the benefit of constructing only a small number of modes comes at the cost of potentially corrupted spectral information. An alternative approach that has been widely adopted, for example in [86] and [66], which we also follow in the Thesis, is to first compute a large number of modes and then post-process the results to either pick or generate a smaller number of representative modes.

To achieve this aim, we require a method to determine which DMD modes have the most significant contribution to the underlying dynamics. One way is to appeal to the modal expansion

$$x_j \approx \sum_{i=1}^{N} \alpha_i \phi_i \lambda_i^{j-1}, \tag{3.3}$$

as explained in [146], where $\phi_i \in \mathbb{C}^p$ are the DMD modes, normalised so that $\|\phi_i\|_2 = 1$, and $\alpha_i \in \mathbb{C}$ are complex constants. For sequential data, these are typically obtained by substituting $j = 1$ in (3.3) and solving the linear least-squares problem

$$\min_{(\alpha_i)_{i=1}^n} \|x_1 - \Phi\alpha\|_F^2, \tag{3.4}$$

where $\alpha$ is a column vector containing the constants $\alpha_i$ and $\Phi$ is a matrix whose columns are the DMD modes. This and a similar ranking method, which is more suitable for non-sequential data, were explained in [166] as methods of scaling DMD modes according to the significance of their contribution to the data. The *ranking* of the modes is formed

by assigning a measure of significance, $\sigma_i$, to each DMD mode $\phi_i$. In this case, the value of $\sigma_i = |\alpha_i|$ is referred to as the *amplitude* of the mode $\phi_i$ (see [153] and [68] for example). Note that in this method all DMD modes are used to recreate the snapshots of data. Conversely, a sparse representation of the system would reconstruct much of the original data using as few modes as possible.

Sparsity promoting DMD, DMDsp, proposed in [86], achieves this by solving the optimisation problem

$$\underset{\alpha}{\text{minimize}} \quad \|X - \Phi D_\alpha \hat{V}\|_F^2 + \gamma \|\alpha\|_1, \tag{3.5}$$

where $X$ is the matrix of the snapshots, $\Phi$ is a matrix whose columns are the normalised DMD modes, $D_\alpha := \text{diag}(\alpha)$ and $\hat{V} := (\lambda_i^{j-1})_{i,j=1}^n$ is the Vandermonde matrix formed from the DMD eigenvalues. Finally, $\gamma \geq 0$ is a parameter which encourages a sparse solution to the optimization problem. The first term in (3.5) amounts to fixing $\lambda_i$ and $\Phi_i$ and finding the optimal coefficients $\alpha_i$ in (3.3), while the second seeks to penalise a large number of non-zero $\alpha_i$ coefficients. Importantly, DMDsp does not calculate new modes or eigenvalues. Instead, it retains a sparse subset of the original DMD modes. The $\ell_1$-norm regularisation allows the user to specify the emphasis on the sparsity of the model, and thus models of various orders can be constructed. It is still, therefore, the case that further sensitivity analysis is required in order to determine a sensible number of modes to use for reduced order modelling. Ideally, one would hope that there is an obvious choice of cardinality, where using more modes only marginally reduces the discrepancy between the original data and the reconstructed data. However, as shown in §3.3, this is not always the case for DMDsp and further analysis may be required if the quality of reconstruction does not plateau as the cardinality of the sparse set increases, or if such a plateau occurs at a high cardinality.

In the Thesis, we propose an alternative method which seeks to reduce the model order by analysing the *spatial similarity* and the *spectral similarity* of DMD modes. The presented algorithm identifies clusters of spatially similar modes, represents the modes in

each cluster with a single mode and, by comparing those representative modes, iteratively regroups the modes into larger clusters. In §3.2.1 we first define a measure of similarity between two modes, and then construct a graph which contains the similarity measure between each pair of modes. We then review the necessary concepts from graph theory and formulate the clustering problem in §3.2.2. In §3.2.3 we explain how our method consolidates the (possibly large number of) dynamic modes into a smaller number of clusters of similar modes. In §3.2.4 we explain how to rank each cluster, by appealing to the original ranking measure for the modes in each cluster. §3.2.5 introduces a *spectral similarity* measure which can also be applied to avoid spectrally dissimilar, but spatially alike, modes being clustered. §3.2.6 explains how the practitioner can employ a range of modal decomposition algorithms and mode-ranking methods in conjunction with the presented clustering algorithm. In §3.3 we apply the clustering approach to a hierarchy of test-cases: DNS snapshots of flow past a circular cylinder at $Re = 60$, PIV snapshots of flow past a rectangular cylinder at $Re = 10^4$ and PIV snapshots from the flow past an axisymmetric bluff body at $Re = 1.88 \times 10^5$.

## 3.2 Methodology

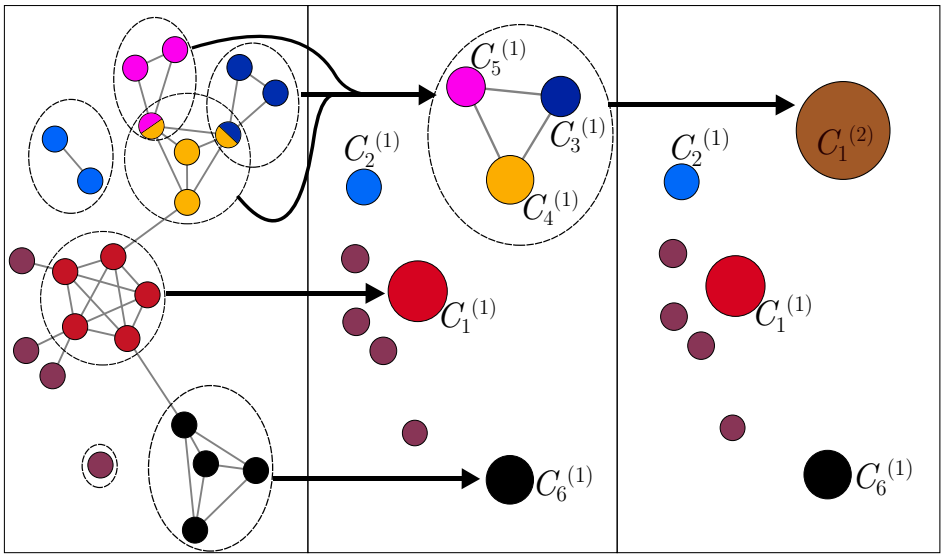



Figure 3.2: A schematic representation of the proposed mode clustering Algorithm 1. The displayed arrows point in the direction of higher algorithm iterations and indicate clusters formed either as an amalgamation of original modes (first iteration, middle pane), or via the amalgamation of clusters (second iteration, right-hand plane).

Before presenting the details of our mode-clustering methodology, we first discuss a schematic overview of the algorithm represented in Figure 3.2. In the left-most panel, each of the possibly large numbers of original dynamic modes is represented by a circle and modes which possess a high metric of similarity (as defined in §3.2.1) are connected by an *edge*. In this way, the original set of DMD modes can be viewed as forming a *graph*. The graph theoretic notion of *maximal cliques*, described in §3.2.2, is then employed to identify initial groups of similar modes, which are shown circled in the left-most panel of Figure 3.2. At the next stage, shown in the middle panel, we refer to each set of modes as a cluster $C_j^{(1)}$, and dimensional reduction is achieved by representing the (possibly large) number of modes in each cluster with a single *representative mode*, denoted by larger circles. The algorithm then proceeds iteratively, by finding maximal cliques of sufficiently similar representative modes and regrouping the underlying modes into larger clusters

(the right-most panel of Figure 3.2). The algorithm terminates when the clusters are sufficiently distinct and the representative modes do not form maximal cliques. We now explain the technical detail underpinning the clustering algorithm.

### 3.2.1 A measure for mode similarity

Assuming that the underlying data ensemble is real-valued, DMD modes and eigenvalues are either real-valued or come in complex-conjugate pairs. As a consequence, the modal expansion (3.3) of the underlying data ensemble implies that the proportion of the data described by the model (3.1), that is $UU^\top \mathcal{X}$, satisfies

$$UU^\top \mathcal{X} \subseteq \bigcup_{i=1}^{n} \operatorname{span}_{\mathbb{R}} \left( \operatorname{Re}(\phi_i), \operatorname{Im}(\phi_i) \right).$$

In other words, regardless of the modelling technique used to find a mode $\phi_i$, the contribution of this mode to the projected dataset $UU^\top \mathcal{X}$ is contained in the subspace spanned by its real and imaginary parts. Importantly, since DMD modes are not spatially orthogonal, the respective subspaces spanned by any given pair of modes $\phi_i$ and $\phi_j$ may have a non-empty intersection (i.e. they may overlap), a property which is captured by the *angle* between the subspaces as defined in (3.6) below. By definition, pairs of modes which are spatially similar will span similar subspaces. Consequently, if a large cluster of modes spanning similar subspaces is found, the degree of multicollinearity amongst those modes implies that the cluster could in theory be represented, more efficiently, by a smaller number of modes. Since such clusters are known to exist as a result of approximating nonlinear phenomena with a linear model [10], this motivates using spatial similarity as a plausible means of dimensionality reduction in modal flow analysis.

A possible limitation for a clustering approach based solely upon spatial similarity is in the case in which the underlying system possesses significantly different dynamical features (i.e., modes with different spectral characteristics) which are, nonetheless, spatially similar. Since, typically, a single mode is related to a single eigenvalue, it would not be

desirable to seek to cluster such spectrally distinct modes. For fluid mechanical systems, however, there is often a strong correspondence between the length-scales and time-scales associated with dynamic phenomena. For example, eddies with larger length scales typically evolve at lower temporal frequencies than eddies with smaller spatial length scales. Consequently, for fluid mechanical systems at least, a clustering methodology based upon spatial similarity alone may be effective. In this chapter, such an approach is presented first. However, in §3.2.5, an extension is presented which can systematically prevent the clustering of modes which are spectrally distinct. This can be applied to systems which are expected to possess spatially similar dynamical features but with distinct spectral content.

Given a DMD mode $\phi_i \in \mathbb{C}^p$, let $A_i := [\mathrm{Re}(\phi_i), \mathrm{Im}(\phi_i)] \in \mathbb{R}^{p\times 2}$. For modes $\phi_i$ and $\phi_j$, the statistic

$$\theta_{ij} = \sin^{-1}\left(\|A_i - A_j(A_i^\top A_j)\|_2\right), \tag{3.6}$$

where $\|\cdot\|_2$ denotes the largest singular value of a matrix, $\theta_{ij}$ is the smallest angle between the subspaces spanned by their respective real and imaginary parts [24]. For simplicity, we consider $(\epsilon = \cos(\theta_{ij})) \in [0, 1]$ as the measure of mode similarity, with 0 denoting the least similar modes, and 1 for modes which span the same subspace of $\mathbb{R}^p$. Note that we have not used the spectral information given by DMD, in our definition of similarity, although this possibility is discussed in §3.2.5.

Our aim is to find *clusters* of spatially similar modes. To achieve this, we will draw upon techniques from graph theory. In the following, it will be useful to express the similarity pattern of a set of modes in a binary matrix. In particular, given a set $\{\phi_i\}_{i=1}^n$ of modes and a tolerance level $0 \le \epsilon \le 1$, define coefficients $d_{ij} \in \{0, 1\}$ by

$$d_{ij} = \begin{cases} 1, & \text{if} \quad \cos(\theta_{ij}) - \delta_{ij} \ge \epsilon, \\ 0, & \text{if} \quad \cos(\theta_{ij}) - \delta_{ij} < \epsilon. \end{cases}$$

We write $d(\{\phi_i\}_{i=1}^n) := (d_{ij})_{i,j=1}^n \in \mathbb{S}^n$ as the symmetric matrix created by these coefficients, with the interpretation that a pair of modes are spatially similar if $d_{ij} = 1$ and dissimilar otherwise if $d_{ij} = 0$. Note that $d_{ii} = 0$, meaning that we discount a mode's inherent self-similarity.

### 3.2.2 A graph-theoretical approach

In order to describe our clustering algorithm precisely, it is necessary to introduce some elementary concepts from graph theory.

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set of vertices $\mathcal{V} \subset \mathbb{N}$ and a set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, where $(i, j) \in \mathcal{E}$ implies that there is an edge connecting vertex $i$ to vertex $j$. More commonly, the connections between the nodes are presented in an *adjacency matrix* $A = (a_{ij}) \in \mathbb{S}^n$ where $a_{ij} \neq 0$ if $(i, j) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise. In this case, we write $(\mathcal{V}, \mathcal{E}) = (\mathcal{V}, A)$. The values $a_{ij}$ are known as the *edge weights* and signify the strength of the connections between nodes $i$ and $j$. An *unweighted* graph, is one where the edge weights are binary, with $a_{ij} = 1$ if $(i, j) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise. A graph is *undirected* when $(i, j) \in \mathcal{E}$ if and only if $(j, i) \in \mathcal{E}$. In this case, $A$ is symmetric. It is also possible for an edge to connect a node to itself. Such an edge would be called a *loop*.

Using the above definitions, a *simple* graph, is an *unweighted, undirected* graph where there are no loops and each pair of nodes are connected with at most one edge.

Now, given a set $\{\phi_i\}_{i=1}^n$ of DMD modes, we may associate with them a simple graph $(\{1, 2, \ldots, n\}, d(\{\phi_i\}_{i=1}^n))$ using the measure of mode similarity from §3.2.1. The problem of identifying clusters of modes is now re-expressed as one of finding sub-graphs with a high connection density. In this context, a useful graph-theoretic notion is that of a *clique* which, by definition, is a complete subgraph (i.e., one where each pair of vertices are connected by a direct edge). A *maximal clique* is a clique which is not strictly contained in any other clique.

All maximal cliques of a simple graph can be obtained using the recursive algorithm developed in [27], variants of which have been used in a variety of applications [8, 92, 81].

A brief explanation of the Bron-Kerbosch algorithm is included in §A.1. Using the notion of maximal cliques to group modes, we ensure that as many similar modes as possible are contained within each cluster. Maximal cliques are not partitions of modes, and therefore a mode can be in two different maximal cliques. This allows for the mode to contribute to more than one dynamical feature and prevents arbitrary classification of a mode. However, this gives rise to another issue that many maximal cliques may contain many of the same modes. In other words, the problem of having many similar modes could just be replaced by that of having many similar clusters. To minimise the chances of this occurring, one can apply this idea iteratively by finding similar clusters and allowing them to coalesce into larger ones.

In the following section, we describe such an iterative algorithm. The ultimate aim is to identify a small number of dynamical features of interest from a larger initial ensemble of modes. Note that in the following, given a graph $\mathcal{G} = (\mathcal{V}, A)$, we let $c(\mathcal{G}) \subseteq 2^{\mathcal{V}}$ * denote the set of maximal cliques (whose members are subsets of $\mathcal{V}$).

### 3.2.3 Creating Dynamic Mode Clusters Using Maximal Cliques

Consider a set of DMD modes $\{\phi_i\}_{i=1}^n$ and define, initially, the coarsest-possible set of clusters $C_j^{(0)} := \{\phi_j\}$, for each $j = 1, \ldots, n$. Next, consider the graph $\mathcal{G}_0 := (\{1, 2, \ldots, n\}, D_0)$ with adjacency matrix $D_0 = d(\{\phi_i\}_{i=1}^n)$ and let $\{H_j^{(1)}\}_{j=1}^{n_1} = c(\mathcal{G}_0)$ be its maximal cliques. For each $j = 1, \ldots, n_1$, we call

$$C_j^{(1)} := \{\phi_i : i \in H_j^{(1)}\} = \bigcup_{i \in H_j^{(1)}} C_i^{(0)}$$

the *cluster* formed of the underlying modes associated with the clique $H_j^{(1)}$.

We now seek to construct a representative mode for each cluster. To do this, we first rescale the modes in each cluster $C_j^{(1)}$ using a given mode-ranking method. Examples of suitable rescaling factors $\sigma$ are the statistics presented in §3.1 or discussed subsequently

* the notation $2^{\mathcal{V}}$ means "the set of subsets of $\mathcal{V}$"

in §3.2.4, but may include any method of assigning to each mode a positive real number representing its importance to the underlying data ensemble. We form the matrices

$$\mathcal{M}_j^{(1)} := \begin{pmatrix} & \uparrow & \\ \cdots & \phi_i\sigma_i & \cdots \\ & \downarrow & \end{pmatrix}, \quad \text{for } i \text{ such that } \phi_i \in C_j^{(1)},$$

perform a singular value decomposition

$$\mathrm{SVD}(\mathcal{M}_j^{(1)}) = \mathcal{U}_j^{(1)} S_j^{(1)} \mathcal{V}_j^{(1)}, \tag{3.7}$$

and let the first left-singular-vector, the first column of $\mathcal{U}_j^{(1)}$, be the representative mode $\phi_j^{(1)}$ of the cluster $C_j^{(1)}$.

The subsequent iteration is to form cliques of the representative modes. To do this, we create the adjacency matrix $D_1 := d(\{\phi_j^{(1)}\}_{j=1}^{n_1})$, from graph $\mathcal{G}_1 = (\{1, \ldots, n_1\}, D_1)$ and find maximal cliques $\{H_j^{(2)}\}_{j=1}^{n_2} = c(\mathcal{G}_1)$. The second-generation clusters of modes are then defined by

$$C_j^{(2)} := \bigcup_{i \in H_j^{(2)}} C_i^{(1)}, \qquad j = 1, \ldots, n_2.$$

This process can be repeated iteratively, as detailed in Algorithm 1, by forming representative modes for each new cluster, forming the associated undirected graph and then obtaining its maximal cliques. Since we are not considering the self-similarity of the modes, the algorithm terminates at an iteration where none of the remaining clusters are sufficiently similar for additional maximal cliques to form.

**Algorithm 1** An iterative algorithm for sparse feature identification

1: $\sigma(\phi) \in \mathbb{R}^+$ ▷ chosen measure of mode significance
2: $C_j^{(0)} \leftarrow \{\phi_j\}$, for $j = 1, \ldots, n$. ▷ $0^{\text{th}}$-generation clusters (DMD modes)
3: $\mathcal{G}_0 = (\{1, \ldots, n\}, d(\{\phi_j\}_{j=1}^n)$. ▷ initial graph
4: $\{H_j^{(1)}\}_{j=1}^{n_1} = c(\mathcal{G}_0)$. ▷ initial maximal cliques
5: $k = 1$.
6: **while** $\{H_j^{(k)}\}_{j=1}^{n_k} \neq \emptyset$ **do**
7: $C_j^{(k)} := \bigcup_{i \in H_j^{(k)}} C_i^{(k-1)}, \qquad j = 1, \ldots, n_k.$ ▷ create clusters
8: $\mathcal{M}_j^{(k)} := (\phi_i \sigma_i, \ldots), \quad i : \phi_i \in C_j^{(k)}$ ▷ matrix $\mathcal{M}_j^{(k)}$ of rescaled DMD modes in $C_j^{(k)}$
9: $\{\phi_j^{(k)}\}_{j=1}^{n_k} \leftarrow \text{SVD}(\mathcal{M}_j^{(k)})$ ▷ representative modes
10: $\mathcal{G}_k = (\{1, \ldots, n_k\}, d(\{\phi_j^{(k)}\}_{j=1}^{n_k}))$ ▷ create graph, based on tolerance $\epsilon$
11: $\{H_j^{(k+1)}\}_{j=1}^{n_{k+1}} \leftarrow c(\mathcal{G}_k)$ ▷ maximal cliques of the graph $\mathcal{G}_k$
12: $k \leftarrow k + 1$
13: **end while**

It is important to note that not all clusters converge (in the sense that their underlying DMD modes do not change) in the final iteration of the algorithm. In particular, at a given iteration $k$, Algorithm 1 regroups the original DMD modes into different clusters, using the similarity of the representative modes $\phi_j^{(k)}$. Now, at a given iteration, one or more representative modes may not be similar to any others. Consequently, the clusters represented by these modes will no longer coalesce with another cluster. This does not imply that $\phi_s^{(k)}$ should now be disregarded, but merely that the flow features represented by it are judged distinct from those described by the remaining representative modes. We, therefore, refer to these clusters as *converged*. Formally, $C_j^{(k)}$ is converged if and only if $C_j^{(k)} \not\subset C_\ell^{(k+1)}$ for all $\ell$. Conversely, if $C_j^{(k)} \subset C_\ell^{(k+1)}$ for some $\ell$, then all the modes contained in $C_j^{(k)}$ are also contained in $C_\ell^{(k+1)}$ and therefore $C_j^{(k)}$ is referred to as a *redundant* cluster.

Since converged clusters may be obtained at any iteration, it is useful to define a measure of *cluster significance* with which the identified clusters can be ranked.

### 3.2.4 Ranking clusters

We first describe existing measures of individual mode significance, then extend these to apply to clusters $C_j^{(k)}$. Appealing to the expansion (3.3), measures of mode significance typically take into account both the amplitude $\alpha_i$ and growth rate $|\lambda_i|$ of each mode $\phi_i$. Given that the spectral information $\lambda_i$ is directly available, e.g. from DMD, the constants $\alpha_i$ can be found by solving the optimisation problem

$$\min_{(\alpha_i)_{i=1}^n} \|X - \Phi D_\alpha \hat{V}\|_F^2$$

where $\Phi$ is the matrix whose columns are the normalised DMD modes, $D_\alpha := \text{diag}(\alpha_i)$ and $\hat{V} := (\lambda_i^{j-1})_{i,j=1}^n$ is the Vandemonde matrix formed using the DMD eigenvalues. This is equivalent to solving (3.5) with $\gamma = 0$ and an analytical solution to it was proposed in [86].

Since we are not utilising the $\ell_1$ norm regularisation, these values may give an inaccurate representation of the significant features in the flow, since a mode may have a large constant $\alpha_i$ yet a small associated eigenvalue $|\lambda_i| \ll 1$. For this reason, in this chapter, we use a measure introduced in [95] in which each mode $\phi_i$'s significance $\sigma(\phi_i)$ is defined by

$$\sigma(\phi_i) := |\alpha_i| \left( \sum_{j=1}^{N+1} |\lambda_i|^{2(j-1)} \right)^{\frac{1}{2}}. \tag{3.8}$$

Note that this metric is the $\ell_2$ norm of the time series multiplying each mode $\phi_i$ in (3.3). Subsequently, we define the significance of a cluster as

$$\sigma(C_j) := \sum_{\{i:\phi_i \in C_j\}} \sigma(\phi_i). \tag{3.9}$$

If two clusters join to form a larger cluster in the next generation, i.e. if $C_i^{(k)}, C_j^{(k)} \subset C_\ell^{(k+1)}$, then the contributions of all the modes which make up the smaller cluster are accounted for in $\sigma(C_\ell^{(k+1)})$. This will make $C_i^{(k)}$ and $C_j^{(k)}$ redundant. However, if $C_i^{(k)}$ converges at iteration $k$, then $\sigma(C_i^{(k)})$ can be used for comparison with other clusters, including the ones created during the later iterations.

### 3.2.5 Spectral similarity

So far, Algorithm 1 has been designed to only employ the spatial similarity of DMD modes. A natural and simple modification is to use both spectral and spatial similarity in the underlying definition of similarity between two modes. In particular, we define the spectral similarity of two modes $\phi_i$ and $\phi_j$ by

$$s_{ij} = \frac{1}{1 + |St_i - St_j|^p}, \tag{3.10}$$

where $St_i$ is the Strouhal number of mode $\phi_i$, and $p \in \mathbb{R}$ is a user-defined parameter. Figure 3.3 shows this family of curves at various values of $p$.

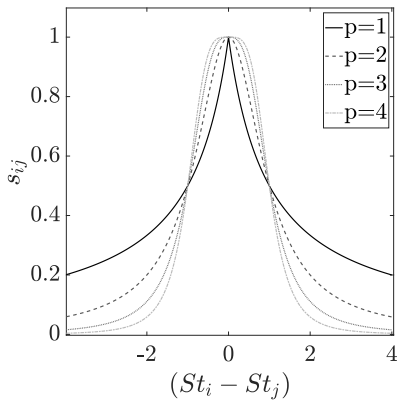



Figure 3.3: Families of spectral similarity curves given by (3.10) for $p = 1, \ldots, 4$.

We can now augment the notion of spatial similarity by applying an analogous cut-off parameter, $s_0$, for spectral similarity and define a new adjacency matrix

$$d_{ij} = \begin{cases} 1 & \text{if } \cos(\theta_{ij}) - \delta_{ij} \geq \epsilon \text{ and } s_{ij} \geq s_0, \\ 0, & \text{if } \cos(\theta_{ij}) - \delta_{ij} < \epsilon \text{ or } s_{ij} \leq s_0. \end{cases}$$

If spectral similarity is employed, in subsequent iterations of Algorithm 1 the mean Strouhal number of the modes in each cluster is used to measure the spectral similarity of clusters.

The spectral similarity statistic $s_{ij}$ considered in this thesis is defined in terms of the absolute difference $|St_i - St_j|$ of the non-dimensional Strouhal numbers of two modes. A possible generalisation of this approach, which could be investigated in future research, is to normalise this statistic to take into account the relative magnitude of each considered pair of Strouhal numbers. For example, a distance defined by replacing $|St_i - St_j|$ with a normalised term of the form $|St_i - St_j|/|St_i + St_j|$ would emphasise differences in Strouhal number in the low-frequency range $St \ll 1$ in comparison to the statistic $s_{ij}$ considered in this thesis. However, as will be observed in the subsequent analysis of bluff body flows across a range of Reynolds numbers, the un-normalised statistic (3.10) proves to be effective due to the fact that the underlying spatial similarity statistic is typically found to be small for modes with even moderately distinct Strouhal numbers (see Figures 3.27 and 3.29, for example).

### 3.2.6 Modularity of the clustering algorithm

It is important to note that the mode ranking defined by (3.8) is only one of many possible methods to use within Algorithm 1. Indeed, the algorithm can employ any other measure of mode significance, $\tilde{\sigma}(\phi_i)$ which assigns a positive real value to each mode. The only required criterion is that the significance measures of all the modes in a cluster can reasonably be aggregated to represent the significance of the cluster. For example, one could use $\tilde{\sigma}(\phi_i) = |\alpha_i|$, where $\alpha_i$ is found using DMDsp or by solving (3.4). It is important to note that for the chosen underlying significance measure, the distribution of modal significance rankings $\sigma(\phi_i)$ is likely to be highly flow-dependent. Consequently,

this flow-dependency will be inherited by the cluster ranking statistic (3.9). Therefore, if a user requires an automatic method of classifying converged clusters into ones which are "significant" and "insignificant", a reasonable method would be to prescribe a cluster rank threshold in terms of proportion of the total absolute sum of significance statistics $\sum_i \sigma(\phi_i)$ of the underlying modes.

Finally, we note that Algorithm 1 does not depend on how the original dynamic modes were obtained, that is, via DMD, exact DMD, OMD or any other standard modal algorithm. If the obtained modes are not spatially orthogonal, Algorithm 1 can be used to attempt to group them into clusters.

## 3.3 Results

In this section, we discuss the performance of Algorithm 1 when applied to modes obtained from flows of increasing dynamic complexity. In particular, we analyse modes obtained from DNS data of flow past a circular cylinder at low Reynolds number $Re = 60$; modes extracted from experimental PIV data of flow past a square cylinder at $Re \approx 10^4$; and, finally, modes obtained from a more dynamically complex, and three-dimensional, flow past an axisymmetric bluff body at $Re \approx 10^5$. In each case, we determine an appropriate value of the spatial similarity threshold $\epsilon$ and study the robustness of Algorithm 1. Algorithmic performance is compared with the existing technique DMDsp, and the benefits of employing the spectral similarity measure described in §3.2.5 are demonstrated for modes obtained from flows of greater dynamic complexity.

### 3.3.1 Flow past a circular cylinder at Re = 60

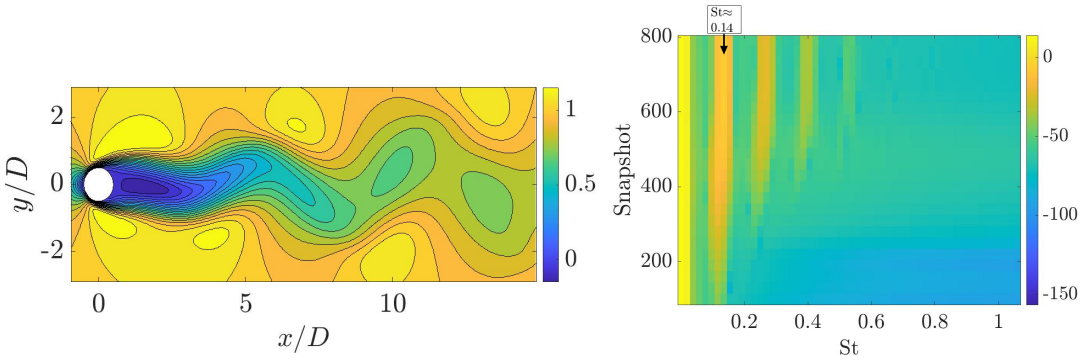
The much-studied two-dimensional flow past a circular cylinder at a low Reynolds number is dominated by the von Kármán street of periodically shed and advected vortical structures. Well-established reduced-order representations of this flow exist and are extracted,

for example, using POD in [128], through weakly-nonlinear analysis as developed in [157], or using OMD as in [97].

The known coherent structures of this flow, which one would expect to extract from a modal analysis, are: the *mean flow field*; the oscillatory *global modes* associated with the dominant vortex-shedding frequency; a non-oscillatory transient mode which adjusts the mean flow from the steady state to the limit cycle, commonly referred to as the *shift mode*; and a series of harmonics of the global mode.

To generate a data ensemble, flow past a circular at $\mathrm{Re} = 60$ is simulated using the in-house solver *Pantheri*, which was previously used and validated for the simulation of cylinder flow in [106]. The code employs a finite volume method on an unstructured grid, with spatial derivatives in the Navier-Stokes equations discretised using a second-order central difference scheme, and time-stepping implemented using a second-order, three-point backward scheme. The spatial domain for the full simulations is $48D$ in length and $32D$ in width, where $D$ is the cylinder diameter. Uniform inflow $U_\infty = 1$ is assumed, with convective outflow, periodic boundary conditions on the lateral domain boundaries, and no-slip conditions imposed on the cylinder boundary. The density and viscosity of the flow were modified to achieve the Reynolds number $\mathrm{Re} = 60$.

The described simulation was used to obtain an underlying data ensemble of 900 snapshots of the velocity-field (the same ensemble was also used for analysis in [97]). This ensemble includes transition of the the flow from close to its unstable steady state, through transition to its stable limit cycle. The 900 snapshots are sequential and separated by a constant time step $\Delta t = 0.25\mathrm{s}$, meaning that the ensemble contains approximately 30 snapshots per shedding period. Figure 3.4 (a) shows one snapshot of the streamwise flow field.



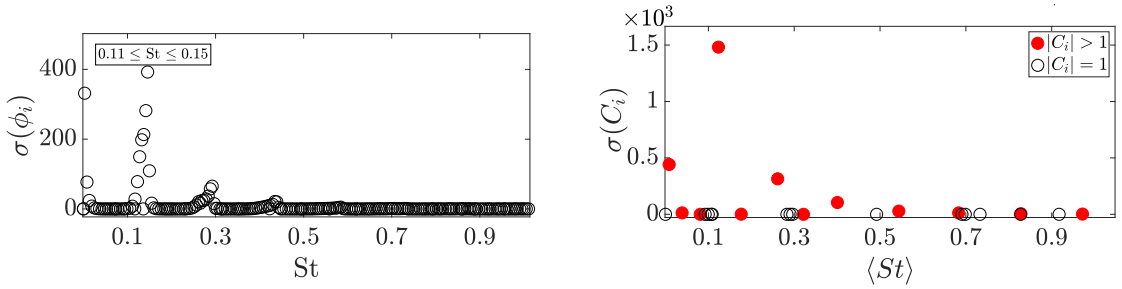

(a) (b)

Figure 3.4: (a) A selected snapshot of the non-dimensionalised streamwise velocity of flow past a circular cylinder at Re=60. (b) Spectrogram of the streamwise velocity signal at $(x/D, y/D) = (6.47, -1.68)$.

To verify the significant frequencies, Figure 3.4 (b) provides the spectrogram of the streamwise velocity at $(x/D, y/D) = (6.47, -1.68)$ for snapshots 100 to 800. At each snapshot $t$, the spectrogram shows the short-time Fourier transform of the signal in the interval between $t - 100\Delta t$ and $t + 100\Delta t$, utilising a Hamming taper function. The Strouhal number associated with the global mode is clearly identified as $St = 0.14$, growing in amplitude through the transition to the limit cycle at approximately snapshot 500. Amplitudes of the global-mode harmonics grow correspondingly throughout the date ensemble.

DMD is applied to the snapshot ensemble and (3.8) is used to rank the significance of each mode, with results shown in Figure 3.5 (a). To improve legibility, the mode associated with the mean flow field is not included.

Peaks in significance associated with the shift mode at $St \approx 0$, the global mode at $St \approx 0.14$ and its less-energetic harmonics are clearly observable.




(a) (b)

Figure 3.5: The significance measure $\sigma$ of DMD modes and clusters for flow past a circular cylinder at Re=60. (a) Individual DMD mode significance, against Strouhal number; (b) mode clusters significance against mean cluster Strouhal number.

We now apply Algorithm 1 to cluster the DMD modes, with $\epsilon = 0.98$ with no consideration of spectral similarity ($s_0 = 0$). Figure 3.5 (b) shows the variation of cluster significance with the average Strouhal number of the modes inside each cluster. Solid red markers indicate clusters which contain more than one mode; hollow markers are singleton clusters. To improve legibility, only clusters with mean Strouhal number in the range $0 \leq \langle St \rangle \leq 1$ are shown, and the singleton cluster associated with the mean flow is not plotted. Comparing Figure 3.5s (a) and (b), it can be seen that clusters of modes close to the dominant flow frequencies (with mean cluster Strouhal numbers $\langle St \rangle \approx 0.12, 0.25$ and 0.38) are obtained. In addition, the second-highest ranked cluster with $\langle St \rangle \approx 0$, corresponding to the shift mode is also identified. The representative modes for the three highest-ranked clusters are shown in Figure 3.6.

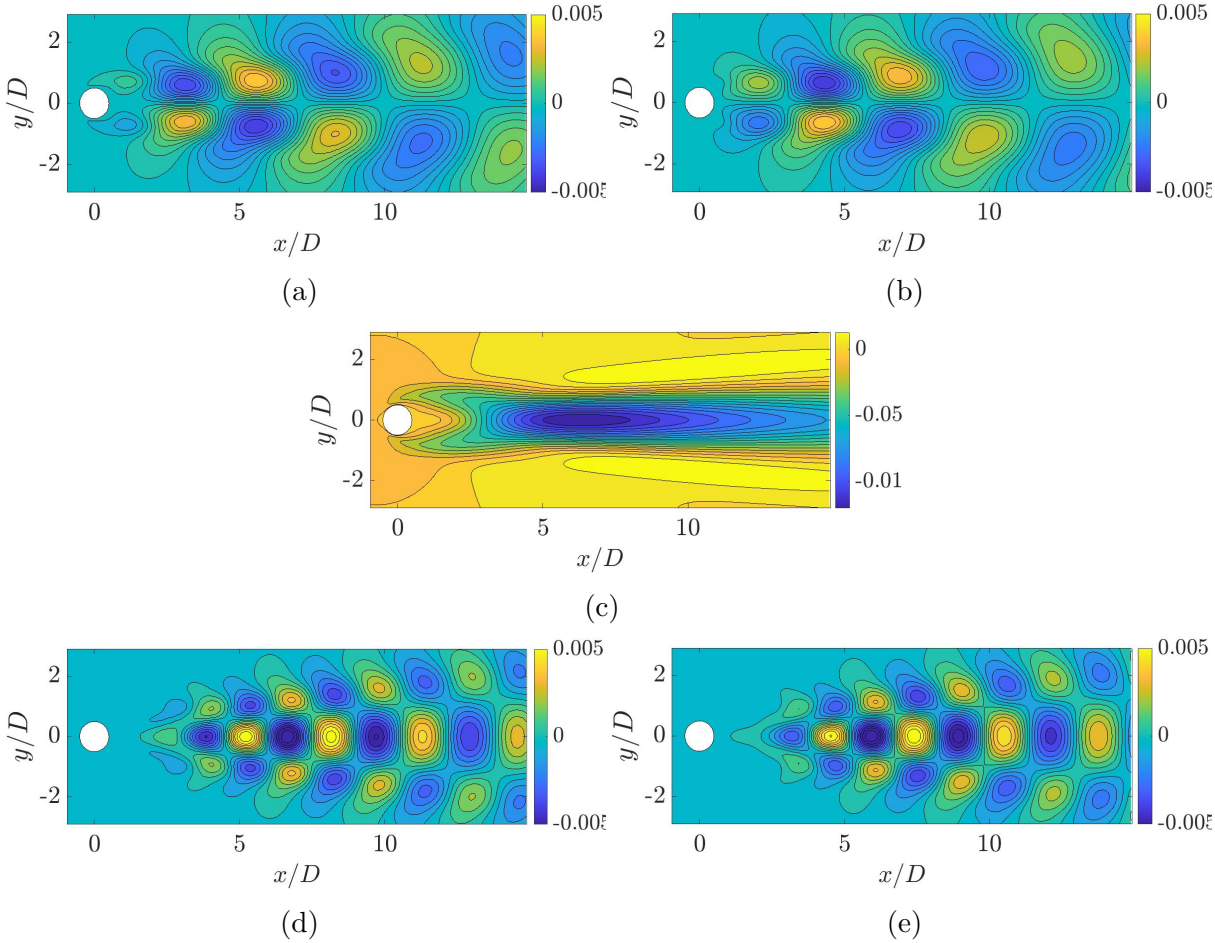
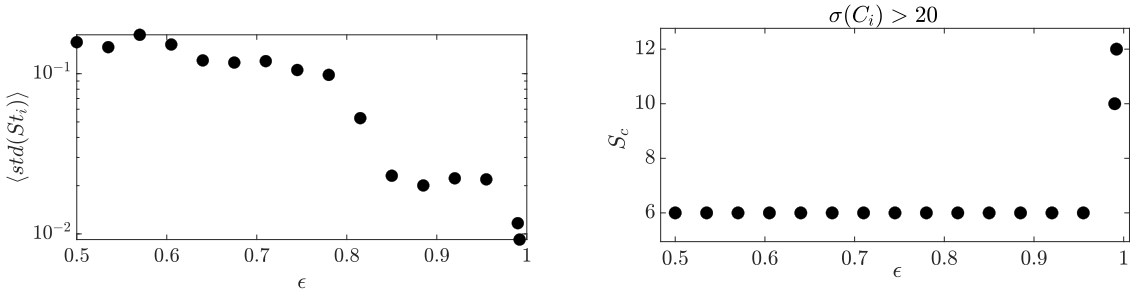



Figure 3.6: The streamwise velocity of representative modes: (a), (b) real and imaginary parts of the global mode at $\langle St \rangle = 0.12$; (c) the real shift mode at $\langle St \rangle = 0.008$; (d),(e) real and imaginary parts of the first harmonic of the global mode at $\langle St \rangle = 0.25$.

#### 3.3.1.1 Sensitivity to the mode similarity cut-off parameter $\epsilon$

To study the robustness of the identified clusters to $\epsilon$, Algorithm 1 was applied for 16 values in the range $0.5 \leq \epsilon \leq 0.992$. Intuitively, a low value of $\epsilon$ promotes cluster formation and coalescence while a value $\epsilon \approx 1$ inhibits clusters. To investigate this effect in more detail, Figure 3.7(a) shows, at each value of $\epsilon$, the standard deviation of Strouhal numbers of modes in non-singleton clusters, averaged across all clusters. It is clear that increasing $\epsilon$ corresponds to less spectral variation within clusters. Two noticeable drops occur at $\epsilon \approx 0.8$ and $\epsilon \approx 0.95$. In the former, for low spatial similarity cut-off parameters, within-cluster variation is increased due to the global mode's cluster also absorbing

high-frequency, low significance, modes. Importantly, the low significance of these modes implies that the representative global mode is not perceptibly modified. In the latter case, increasing $\epsilon$ beyond 0.95 leads to multiple clusters forming at the primary and first-harmonic shedding frequencies, decreasing within-cluster variation.




(a) (b)

Figure 3.7: Effect of the cut-off parameter $\epsilon$ on the clustering of Re=60 cylinder wake modes: (a) average within-cluster variation; (b) the number of significant clusters.

Finally, Figure 3.7 (b) denotes how the total number of significantly ranked clusters varies with $\epsilon$, in particular the number of clusters $S_c$ whose significance metric satisfies $\sigma(C_i) > 20$, corresponding to a cut-off at approximately the level of the second-harmonic. Interestingly, decreasing $\epsilon$ (which promotes cluster formation) does not correspond to an increase in the number of significant clusters demonstrating a level of algorithmic robustness.

We now consider the performance of Algorithm 1 when applied to more challenging data sets, and compare its ability to extract a sparse flow representation to the existing DMDsp methodology.

### 3.3.2 Flow past a square cylinder at Re $\approx 10^4$:

Flow past a square cylinder, with side-length $D = 25.4$ mm, at Reynolds number Re $\approx 10^4$ is considered. The underlying flow dynamics are similar to those discussed in §3.3.1, although this flow is in the turbulent regime. In the case of flow past a circular cylinder, a Reynolds number Re $\approx 10^4$ corresponds to the so-called *shear-layer transition*

*regime* [173] in which the wake is three-dimensional, with higher Reynolds stresses than at lower Reynolds numbers, and an associated reduction in length of the mean recirculation region. Kelvin-Helmholtz shear layer instabilities and von Kármán vortices are present at this Reynolds number. Similar behaviour for flow past a square cylinder has been observed in experiments and simulations in [175].

On the other hand, unlike a circular cylinder, the separation points for a square cylinder are fixed at the leading edge corners. Moreover, while in other rectangular cylinders with higher aspect ratios in which the flow may reattach to the cylinder, for the square cylinder considered here the flow does not undergo reattachment [40]. The flow past a square cylinder in this regime has been experimentally and numerically studied, with [175] providing visualisations of the flow and comparisons to flows past other cylindrical bluff bodies at the same Reynolds number. Studies have also detailed the variation of the Strouhal number as a function of the Reynolds number for the square cylinder [132].

We analyse an ensemble of 1984 PIV snapshots, sampled at a constant frequency of 50 Hz, of a two-dimensional window of the velocity field with dimensions $1D \leq x \leq 6D$ and $-2.4D \leq y \leq 2.1D$. As mentioned above, the cylinder has a side-length $D = 25.4$ mm and the snapshots are collected from an in-house water flume at Imperial College with cross-sectional dimensions 600mm $\times$ 600mm resulting in a blockage of 4.2%. The free-stream velocity was set to $U = 0.313$ ms$^{-1}$, and the free-stream turbulence intensity was measured to be 1.4%. Further details about the experimental set-up are available in [90]. The streamwise and transverse components of the time-averaged flow field are shown respectively in Figure 3.8 (a) and (b). Snapshots of the streamwise and transverse velocity fields are shown in Figure 3.8 (c) and (d).

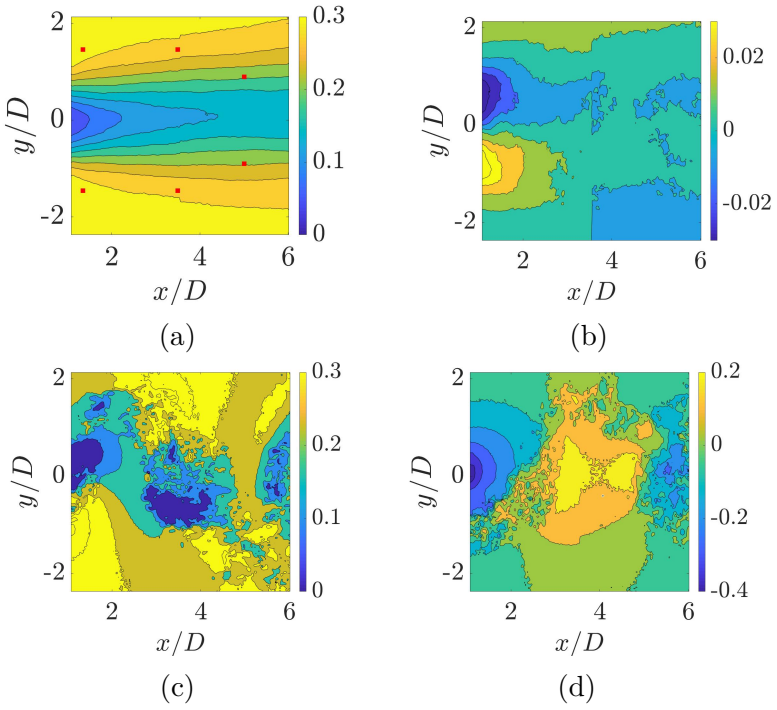



Figure 3.8: Mean streamwise, (a), and transverse, (b), velocity field for flow past a square cylinder at Re = $10^4$; (c) a selected snapshot of the streamwise velocity; (d) a selected snapshot of the transverse velocity field. Red markers in (a) are the positions of the virtual probes. Colorbars indicate the appropriate velocity component in $\mathrm{ms}^{-1}$.

Each snapshot contains streamwise and transverse velocity components at 49862 spatial points, corresponding to a uniform grid of 233 and 214 streamwise and transverse locations, respectively. Data is assembled into data matrices $X$ and $X'$, as discussed in §3.1, to facilitate computation of DMD modes $\phi_i$ and significance statistic $\sigma(\phi_i)$ using (3.8).

Figure 3.9 (a) shows mode significance against Strouhal number, hereafter referred to as a mode's *decomposition frequency*. To improve legibility, we only present modes in the range $0 \leq St \leq 0.5$. A number of highly energetic modes are observed for $0.13 \leq St \leq 0.14$, consistent with the findings in [132]. To further investigate whether the results in Figure 3.9(a) are consistent with the spectral information from the flow, the time series of the transverse velocity component is considered at six locations in the flow field. The virtual probe locations (x/D,y/D), shown with red squares in Figure 3.8(a), are

$$\{(1.35, 1.46), (1.35, -1.46), (3.5, 1.46), (3.5, -1.46), (5, 0.9), (5, -0.9)\}.$$

Power Spectral Densities (PSDs) of each time series, estimated using the Welch method, are plotted in Figure 3.9(b) and show a consistent peak in a frequency band similar to that extracted by DMD. However, Figure 3.9(b) also shows that, especially for probes further downstream, there is also a less pronounced peak at $St \approx 0.27$. This corresponds to the first harmonic of the vortex-shedding, a feature which is not obviously captured by considering $\sigma$ alone.

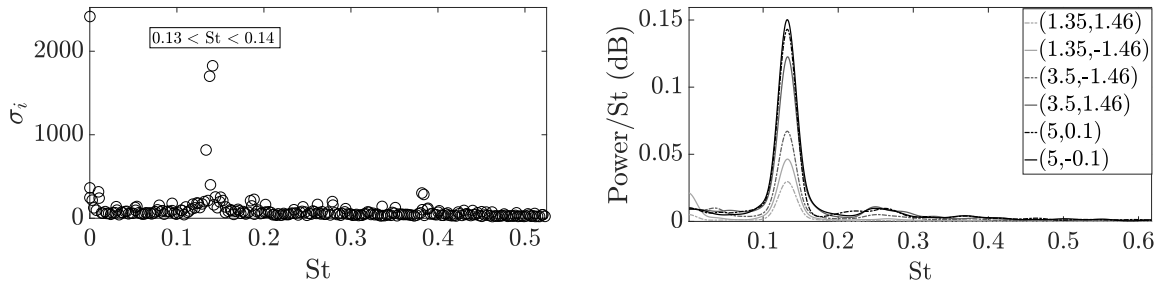
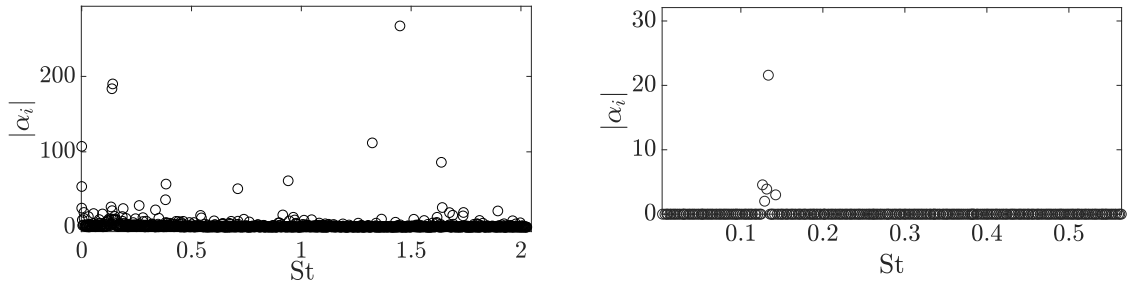



Figure 3.9: (a) Significance statistic $\sigma$ for each DMD mode against decomposition Strouhal number. (b) Power spectral density of the velocity time series against Strouhal number at virtual probe locations indicated in Figure 3.8 (a)

It is interesting to note that another standard significance measure, the mode amplitude $\sigma = |\alpha_i|$ (equivalently found by applying DMDsp with $\gamma = 0$), does not produce results consistent with the expected system frequencies of Figure 3.9(b). This is shown in Figure Figure 3.10 (a). While it is the case that modes at the primary shedding frequency are still highly-ranked, the highest-ranked modes have frequencies close to $St \approx 1.5$, significantly above the known dominant frequencies. In this case, taking the growth rate into account via (3.8) nullifies the effect of this high amplitude behaviour.




(a) (b)

Figure 3.10: Variation of the mode significance $|\alpha_i|$ from DMDsp with Strouhal number for parameter choice (a) $\gamma = 0$ and (b) $\gamma = 9 \times 10^3$.

We note, however, that by increasing the sparsity-promoting parameter $\gamma$, as shown in Figure 3.10 (b), DMDsp can be applied successfully, which we now discuss.

#### 3.3.2.1 Sparsity promotion using DMDsp.

DMDsp is applied, as in (3.5), for a range of sparsity-promoting parameter values $\gamma$. The cardinality of a sparse representation is the number of modes with non-zero amplitudes, and the performance loss (due to dimension reduction) is defined by

$$\Pi_{loss} := 100 \times \frac{\|X - \Phi D_\alpha \hat{V}\|_F}{\|X\|_F}, \tag{3.11}$$

where $X$ is the snapshot ensemble, $\Phi$ is the matrix whose columns are the DMD modes, $D_\alpha := \text{diag}(\alpha_i)$ is the diagonal matrix containing the mode amplitudes for a particular $\gamma$, and $\hat{V} := (\lambda_i^{j-1})_{i,j=1}^n$ is the Vandermonde matrix defined in terms of the DMD eigenvalues.

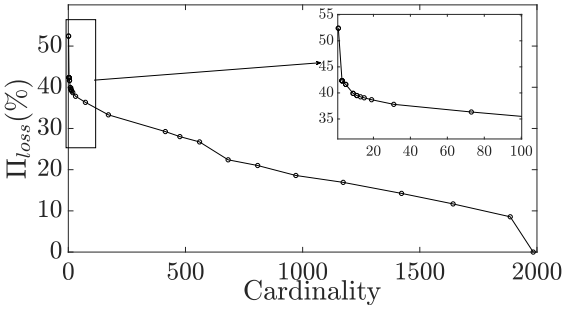



Figure 3.11: Cardinality of the retained set of modes by DMDsp, against performance loss.

The relation between performance loss and cardinality, as $\gamma$ varies, is shown in Figure 3.11. From the inset, it can be seen that an initial increase in the cardinality of the sparse flow representation leads to a steep decrease in $\Pi_{loss}$. However, once cardinality has reached 11 the rate of decrease in $\Pi_{loss}$ flattens but is still clearly decreasing. Indeed, the performance loss does not fall below 30% until more than 400 modes are included in the sparse set. This suggests that it may be challenging to definitively select an appropriate value of the sparsity-promoting parameter $\gamma$.

Nonetheless, it is the case that for a wide range of values of $\gamma$, DMDsp correctly identifies a pair of modes corresponding to the dominant vortex shedding frequency observed in Figure 3.9, and a mode corresponding to the mean flow. Furthermore, these features are consistently the highest-ranked.

However, a closer analysis of the modes retained by DMDsp shows that many modes contain similar spatial features and decomposition frequencies. We consider, for example, DMDsp applied at $\gamma = 9 \times 10^3$, which corresponds to the inflexion point at the cardinality of 11 in Figure 3.11. The DMDsp mode amplitudes are plotted shown in Figure 3.10 (b) where, to improve legibility, points corresponding to the mean mode and the complex conjugate modes are not shown.

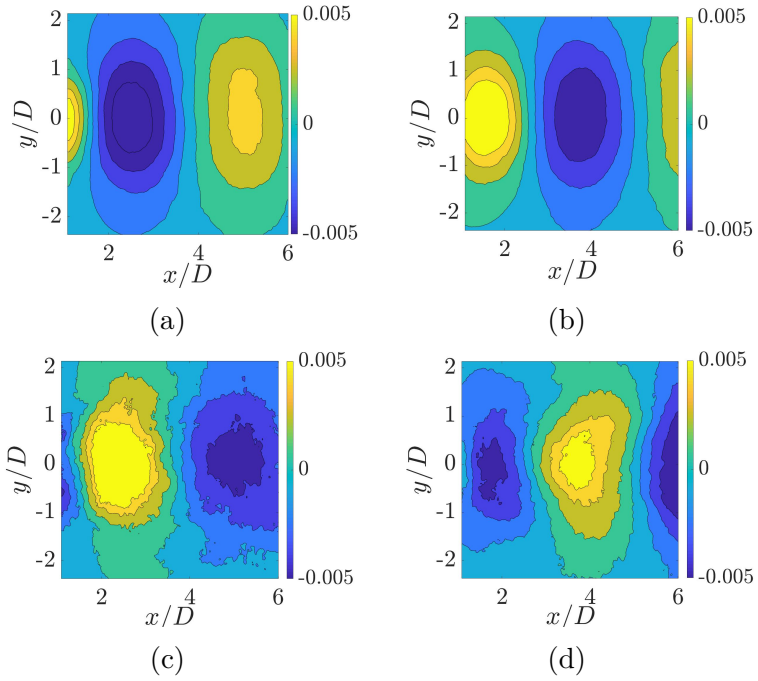



Figure 3.12: An illustrative example of the similarity in the spatial features of the modes retained by DMDsp at $\gamma = 9 \times 10^3$. The real (a,c) and imaginary (b,d) parts of the $v$ component of two of the modes with non-zero amplitude $\alpha_i$ in Figure 3.10 (b).

It can be seen that *only* modes in the known dominant frequency band $0.13 \leq St \leq 0.14$ have non-zero amplitudes. Furthermore, all modes with non-zero DMDsp amplitudes (in this case 10 out of the 11 identified by DMDsp) have very similar spatial and spectral features. An example of this is shown in Figure 3.12, where it should be noted that the phase-shift from one mode (shown in (a), (b)) to the other (shown in (c) and (d)) is irrelevant: both approximately represent the same advective feature.

The importance of this observation is that our chosen measure of cluster similarity, defined in terms of the subspace angle (3.6), was specifically chosen to view such pairs of modes as approximately equivalent. Indeed, if one is searching for a sparse flow model, it may be desirable to only include a single mode to represent such a coherent structure. In DMDsp, it may be possible to obtain distinct flow features by reducing the sparsity-promoting parameter $\gamma$, however, it is unclear how many modes resembling the already-identified features must be added (and for which value of $\gamma$) before this is achieved.

The purpose of Algorithm 1 is to seek sparse system representations that capture a range of dynamical features more efficiently and systematically.

#### 3.3.2.2 Sparse feature identification using mode clustering.

Algorithm 1 is also applied, using the same DMD modes as in §3.3.2.1 and the ranking statistic $\sigma$ defined in (3.8). The spatial similarity cut-off is $\epsilon = 0.94$ and no spectral cut-off is applied ($s_0 = 0$). Figure 3.13 shows the ranking of identified clusters, highlighting those which contain more than one underlying DMD mode.

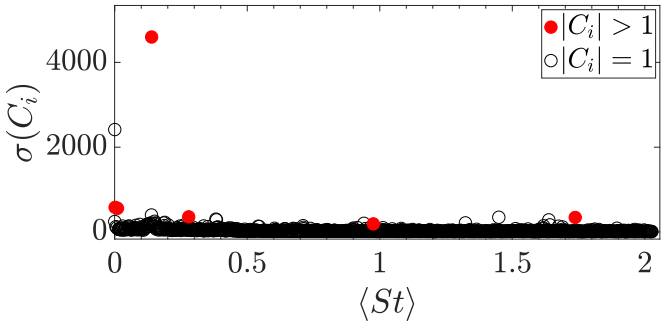



Figure 3.13: Cluster significance $\sigma(C_i)$ for cut-off parameter $\epsilon = 0.94$. Clusters with the cardinality of 1 are displayed with black circles, and those with $|C_j| > 1$ are indicated with red markers.

In agreement with DMDsp and the PSD of the velocity signal, Figure 3.13 shows that the most energetic cluster is associated with the vortex shedding feature at $\langle St \rangle \approx 0.135$ where, again, $\langle \cdot \rangle$ denotes the mean of the Strouhal numbers of the modes in a cluster. The representative mode of this cluster is shown in Figure 3.14 (a-b). Although less highly-ranked, clusters of modes are obtained with $\langle St \rangle \approx 0$ and $\langle St \rangle \approx 0.28$. The latter frequency is also observed in the PSD of the transverse component of the velocity in Figure 3.9 (b) and corresponds to the first harmonic of vortex shedding. The mode is shown in Figure 3.14 (c-d). The representative mode of the cluster at $\langle St \rangle \approx 0$, shown in Figure 3.14 (e), has features which resemble both the so-called "bubble-pumping" mode observed for axisymmetric three-dimensional bluff body wakes in [142], and also superficially resembles the shift mode observed for significantly cleaner DNS cylinder data in

§3.3.1. It is interesting that Algorithm 1 is able to extract the footprint of such a low-frequency mode, even from this more dynamically complex data ensemble.

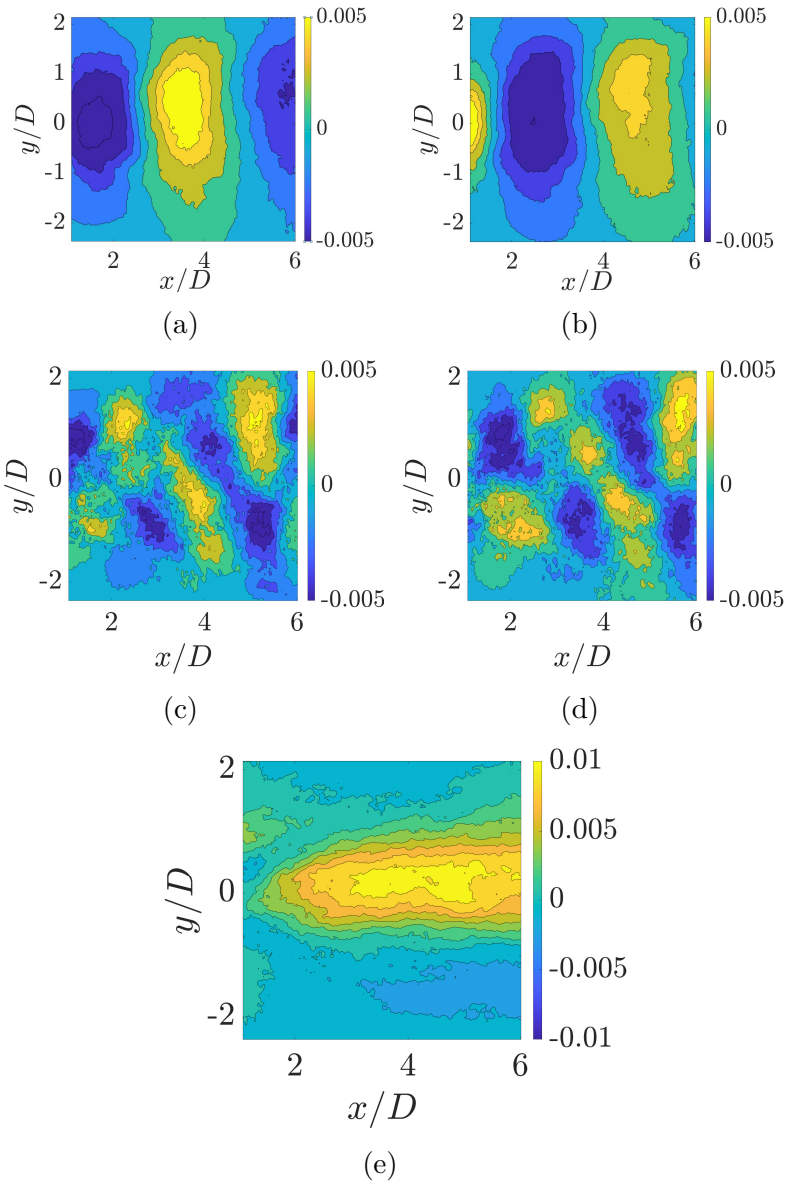



Figure 3.14: Representative modes for flow past a square cylinder: (a-b) real/imaginary parts of the $v$-velocity component of the shedding mode representative; (c-d) real and imaginary parts of the $v$-velocity component of the first harmonic representative; (e) $u$-velocity component of the shift mode representative.

Finally, we note that there are also two non-singleton clusters with $\langle St \rangle \approx 1$ and $\langle St \rangle \approx 1.7$. However, the representative modes of these clusters are dominated by length-scales smaller than $D/4$ and their significance statistic is lower than that of the clusters

described above. This highlights the fact that cluster formation does not necessarily imply dynamical significance. In this case, such spurious clusters can be explained by the underlying sampling frequency of 50Hz which, due to the Nyquist criterion, implies that features with $St > 1.6$ cannot be accurately extracted.

#### 3.3.2.3 Sensitivity to the mode similarity cut-off parameter $\epsilon$.

Similar to the analysis in §3.3.1.1, Algorithm 1 is applied for similarity parameters in the range $0.5 \leq \epsilon \leq 0.97$.

As before, we let $\langle \text{std}(St_i) \rangle$ denote the standard deviation of Strohal numbers within clusters, averaged over all non-singleton clusters at a particular value of $\epsilon$. Figure 3.15 (a) indicates that this metric is stable for $0.77 < \epsilon < 1$, motivating our choice of $\epsilon$ in the above analysis. Below this range, a step-change in within-cluster variation occurs.

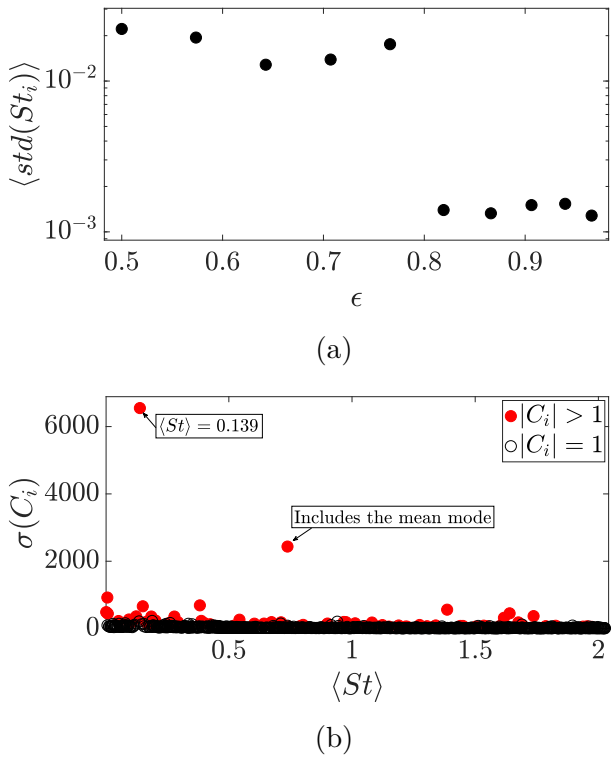



Figure 3.15: (a) average within cluster standard deviation of spectral content, against cut-off parameter $\epsilon$; (b) cluster significance $\sigma(C_i)$ at $\epsilon = 0.77$ with non-singleton clusters indicated by red markers.

Figure 3.15 (b) shows the cluster rankings for $\epsilon = 0.77$, corresponding to a value just prior to the jump. Compared to Figure 3.13 which denotes cluster rankings for $\epsilon = 0.94$, it is first clear that a lower value of $\epsilon$ promotes the formation of high-frequency clusters. Furthermore, at $\epsilon = 0.77$ a cluster is formed with $\langle St \rangle \approx 0.74$, which contains the mean-field mode *and* a mode with $St = 1.48$. This cluster is ranked highly due to the highly-ranked mean mode but, clearly, these two modes do not represent similar dynamic flow features. This behaviour can be explained, however, since the imaginary part of the mean mode is zero at all points in the flow field and the spatial features of the mode at $St = 1.48$ are very close to zero throughout the flow domain. However, the small value of $\sigma_i$ of the second mode means that it has minimal effect on the representative mode of the cluster. We note that (i) the stability of the within-cluster variation for $0.77 \leq \epsilon \leq 1$ indicates a sensible range from which to select $\epsilon$; and (ii) that spectral information, as discussed in §3.2.5, can be included to remove this behaviour even at low values of $\epsilon$.

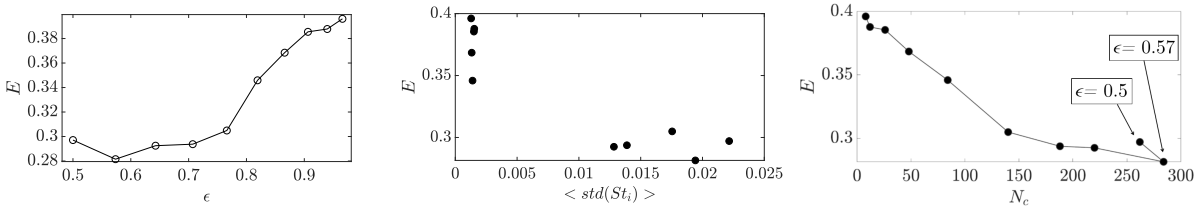
We next determine how the choice of $\epsilon$ influences the extent to which extracted flow features can be used to reconstruct the original data. Noting that non-singleton clusters typically have high cluster ranking, we consider projecting snapshots $X$ onto the representative modes of clusters with $|C_j| > 1$ and whose mean within-cluster frequencies satisfy $\langle St_j \rangle < 0.5$. Note that Figure 3.9 (b) indicates the main coherent structures lie within this frequency range.

Let $Q$ be a matrix whose columns are the chosen representative modes, and let $W = X - \bar{X}$, where $\bar{X}$ is the time-averaged flow field. We study the error between the fluctuating part of the flow field with its projection onto the representative modes, defined by

$$E = \frac{\|(I - QQ^{\dagger})W\|_F}{\|W\|_F}, \tag{3.12}$$

where $Q^{\dagger}$ is the pseudoinverse of $Q$. Figure 3.16(a) shows that increasing $\epsilon$ inhibits cluster formation, reduces the projection basis $Q$ and moderately increases $E$. However, as shown in Figure 3.16(b-c), this comes with the advantage of a significant reduction in both

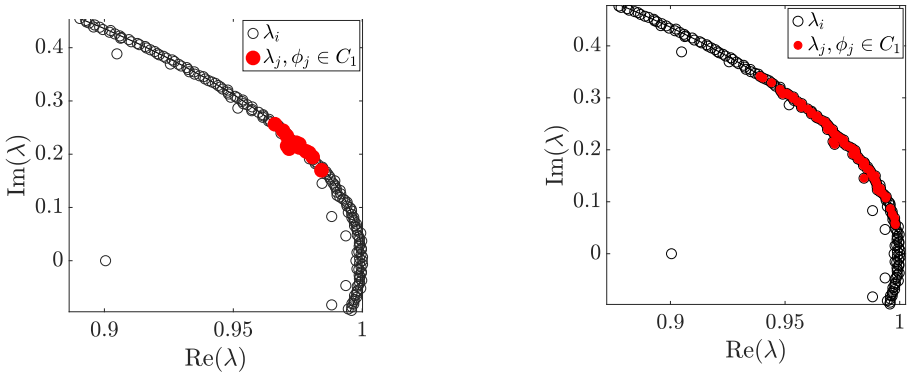
the number of modes required for the sparse representation, and the frequency-variation of the underlying DMD modes within each cluster. Again, for this data ensemble, this motivates selecting the mode similarity parameter in the range $0.77 < \epsilon < 1$.




(a) (b) (c)

Figure 3.16: The error $E$ between the estimated and true snapshots, plotted against (a) cut-off parameter $\epsilon$; and (b) averaged within-cluster standard deviation; (c) the total number of formed clusters.

Finally, Figure 3.17 indicates the underlying DMD modes belonging to the vortex shedding cluster (with $\langle St \rangle \approx 0.135$) for $\epsilon = 0.5$ and $\epsilon = 0.85$. Again, decreasing $\epsilon$ promotes clustering. However, we note that this is not necessarily detrimental, since many newly clustered modes may have very low significance and this is considered in Algorithm 1—in (3.7)—during that calculation of representative modes.

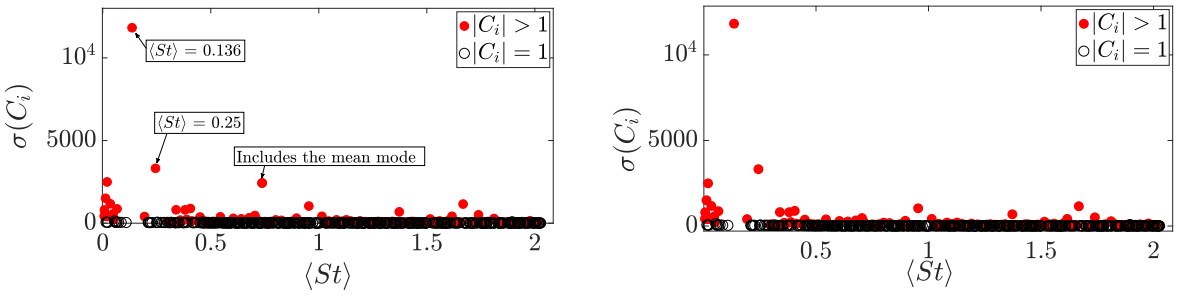



(a) $\epsilon = 0.85$ (b) $\epsilon = 0.50$

Figure 3.17: The effect of cut-off parameter $\epsilon$ on cluster cardinality. Red markers indicate modes which belong to cluster $C_1$ for the two choices (a) $\epsilon = 0.85$ and (b) $\epsilon = 0.50$.

#### 3.3.2.4 Using the spectral similarity cut-off parameter $s_0$.

Although it has been shown that lowering $\epsilon$ can lead to the formation of dynamically mixed clusters, employing the spectral cut-off, $s_0$, can mitigate this problem while retaining the benefit of reduced conservativeness in cluster formation. To see this, Figure 3.18 indicates cluster rankings for a very low spatial cut-off $\epsilon = 0.5$, both without and with spectral cut-off ($s_0 = 0.75, p = 2$). Using the spectral cut-off has two clear advantages for this data. First, the spurious cluster containing the mean mode identified at $\langle St \rangle \approx 0.74$ does not form. Second, in comparison with Figure 3.13, due to the decrease in $\epsilon$ the cluster containing the first harmonic of the shedding mode with $\langle St \rangle \approx 0.27$, is significantly increased in ranking. We note again that this feature was not identified as of obvious importance using classical mode ranking algorithms or DMDsp.




(a) $(\epsilon, s_0, p) = (0.5, 0, 0)$ (b) $(\epsilon, s_0, p) = (0.5, 0.75, 2)$

Figure 3.18: The influence of of spectral similarity cut-off at $\epsilon = 0.5$. (a) shows clusters formed without spectral similarity cut-off applied; (b) shows clusters with spectral cut-off parameters $s_0 = 0.75$ and $p = 2$.

### 3.3.3 Possible advantages of the graph-theoretic approach

Both the application of Algorithm 1 with $\epsilon = 0.94$ studied in §3.3.2.2, and the application of DMDsp with $\gamma = 9 \times 10^3$ in §3.3.2.1 lead to the same number, 11, of features extracted in a sparse representation (see Figure 3.11 and Figure 3.16). Furthermore, the reconstruction error $E$ is similar, at approximately 39%, in each case. In contrast, clustering extracts modes with a range of dynamical features (shedding, harmonic, shift mode), whereas modes extracted by DMDsp all lie in the narrow frequency range $0.13 \leq St \leq 0.14$.

### 3.3.4 Flow past an axisymmetric bluff body at Re $= 1.88 \times 10^5$:

We analyse a final test-case with significant dynamical complexity, flow past a three-dimensional bluff body at moderately high Reynolds number considered in [135]. The mean flow field and a particular wake snapshot are shown in Figure 3.19. The motivation for studying such a data set is highlighting the advantage of iterative cluster formation.

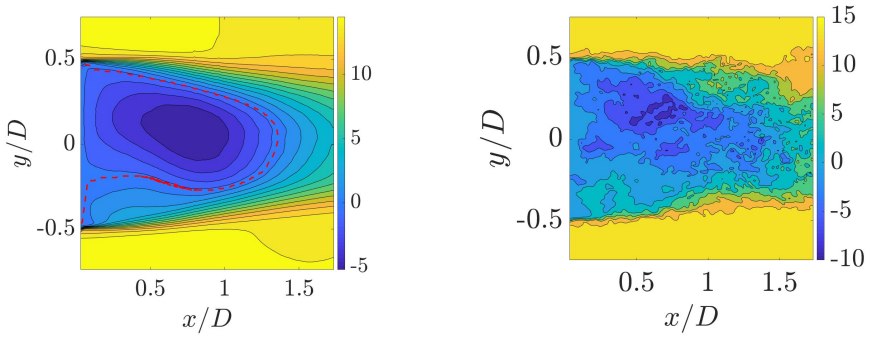



Figure 3.19: Flow past an axisymmetric bluff body at Re$= 1.88 \times 10^5$. (a) streamwise velocity component of the mean flow; (b) an indicative snapshot of streamwise velocity. The average position of the separation region is shown by a dashed red line in (a). Colourbars indicate streamwise velocity in ms$^{-1}$.

A sequential ensemble of 2732 time-resolved PIV snapshots of the velocity-field are considered, sampled at the 720 Hz, from a flow with free-stream velocity $U_\infty = 15$ ms$^{-1}$, past an axisymmetric bluff body with base diameter $D = 0.1965$ m and length-to-diameter ratio $L/D = 6.8$. The PIV field of view is a 2D diametric cross-section of the wake, perpendicular to the horizontal plane, with dimensions $0.05 \leq x/D \leq 1.72$ and $-0.7 \leq y/D \leq 0.7$. Full details of the experimental setup are given in [135]. An underlying mode ensemble is extracted using the Optimal Mode Decomposition algorithm of [174] and post-processed using the method outlined in §3.2.

To provide a benchmark with which to interpret mode clusters, after the application of Algorithm 1, we briefly review the known spatio-temporal features of axisymmetric bluff-body flows.

#### 3.3.4.1 Dynamical features of axisymmetric bluff body flows

Direct numerical simulations in [34] for flow past a similar axisymmetric body at $Re \leq 900$ indicate that, at lower Reynolds numbers, there is a reflectionally asymmetric vortex shedding structure. However, at $Re = 900$, after undergoing a transient "wake twisting" phase, the plane of asymmetry exhibits dramatic, intermittent changes in azimuthal orientation. (see Figures 4 and 6 in [34]). Flows past other axisymmetric objects are also of interest. For example, using hot-wire data in the wake of a circular disk for $1.5\times10^4 \leq Re \leq 3\times10^5$, [20] observed that, apart from vortex shedding at $St \approx 0.135$, the farthest point of the recirculation bubble from the base of the disc oscillates at $St \approx 0.05$ about its mean streamwise location. This extension and contraction of the recirculation region, was referred to as the *bubble-pumping mode*. A high-frequency peak in the power spectrum at $St \approx 1.62$ was also observed in the velocity time-series sampled in the vicinity of the separated shear layer.

We now briefly discuss the known dynamical features for the axisymmetric bluff body flow studied in this chapter. This flow was studied in [142], which collected 2000 base-pressure snapshots at a sampling frequency of 225 Hz, each containing data from 64 pressure tappings placed at regular azimuthal and radial distances. Specifically, the pressure tappings are located at eight radially equidistant, concentric circles on the base of the bullet-shaped bluff body. In [142], Fourier analysis was applied to decompose this ensemble of this flow's fluctuating base pressure into azimuthal mode-shapes. For example, the azimuthal mode $m = 0$ corresponds to all axisymmetric variations, and the modes $m = \pm1$ capture all sinusoidally asymmetric azimuthal variations of the base pressure. Such a decomposition allows one to consider the energetic contribution of each temporal frequency to the overall energy of each azimuthal mode using the pre-multiplied power spectra of the azimuthal modes. From this information, one can speculate about the dominant dynamical features present in the flow.

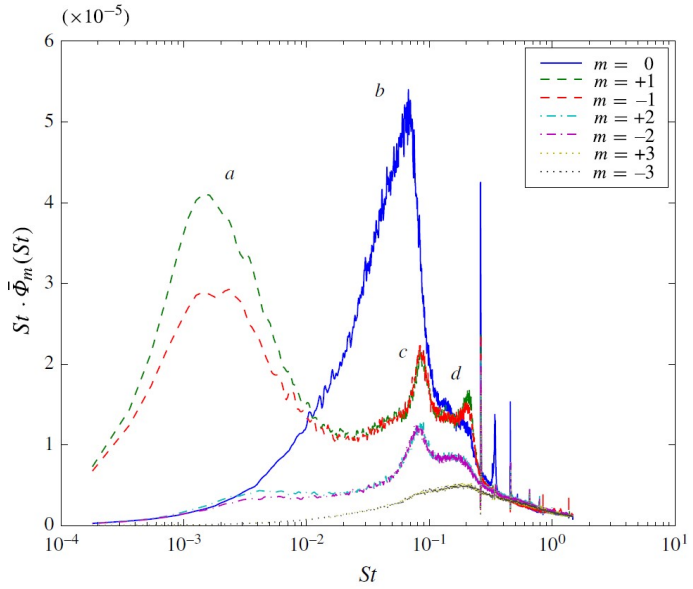



Figure 3.20: Pre-multiplied power spectra of the azimuthal modes of base pressure for flow past an axisymmetric bluff body, reproduced from [142, Figure 3], for $m = 0, \pm1, \pm2, \pm3$.

For completeness, Figure 3.20 reproduces [142, Figure 3] which shows the pre-multiplied power spectral densities of the azimuthal Fourier coefficients for modes $m = 0, \pm1, \pm2$ and $m = \pm3$. Here, the spectrum of the axisymmetric $m = 0$ mode has a clear peak (labelled $b$) at $St \approx 0.06$, which can be viewed as the base-pressure structure associated with the imprint of the wake's bubble-pumping mode. The spectral peak (labelled $d$) of the azimuthal mode $m = 1$ corresponds to an asymmetric flow structure, likely to be the global vortex shedding mode at $St \approx 0.2$. Finally, the azimuthal $m = 1$ mode also has spectral peaks (labelled $c$ and $a$, respectively) associated with the sub-harmonic of the vortex shedding mode at $St \approx 0.1$, and the random rotational changes in the orientation of the flow at $St \approx 0.002$. Note that the energetic content, i.e. the area under the curve, of the very low frequency (VLF) rotations is much higher than that of the vortex shedding events.

It is argued in both [20] and [142] that this flow is not instantaneously axisymmetric, with axisymmetry holding only in a time-averaged sense. Both studies show that the boundary of the recirculation region oscillates about its mean position at $St \approx 0.05$,

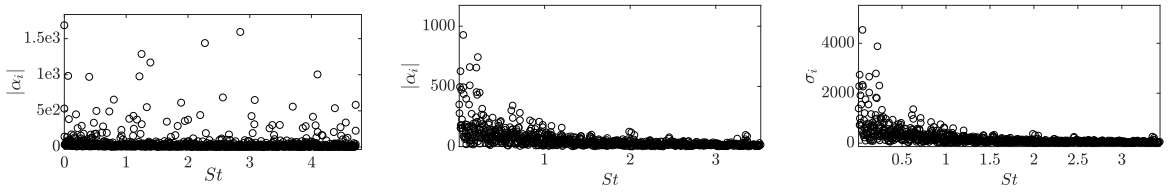
confirming that this frequency is associated with the bubble-pumping feature. However, we note that localised high-frequency features associated with the shear-layer instability were not observed in the base-pressure data used in [142], and the sub-harmonic of the vortex shedding was not detected in [20].

#### 3.3.4.2 Modal decomposition of velocity snapshots

The data ensemble analysed in this section consists of flow snapshots where, on average, the azimuthal orientation of the wake's plane of asymmetry is approximately constant, and parallel to the plane of view, with variance of $\pi/5$ rad. However, most variation is concentrated at two short intervals, each lasting about 0.1 s, corresponding to only approximately 5% of the data.

Underlying dynamic modes are computed using OMD with a relatively large reduction dimension $r = 2000$. Figure 3.21 indicates the OMD mode rankings, calculated using different methodologies. Appealing to mode amplitudes via the classical approach (3.4) performs poorly, with highly-damped, high-frequency features (e.g. at $St \approx 2.9$) ranked as significant. Figure 3.21 (b-c) instead show that rankings obtained either by modifying the analytical solution to (3.5) at $\gamma = 0$ in [86] to be suitable for OMD, or by using (3.8) perform better. As expected, both indicate significant flow features with frequencies in the range $0.02 \leq St \leq 0.25$.

While comparing Figure 3.21(b-c) shows that the (expected vortex shedding) mode at $St \approx 0.2$ is more clearly identified using the ranking (3.8), there are still many OMD modes in this frequency range. Our intention is now to determine whether mode clustering can be used to further distinguish dynamical features in the wake and provider a cleaner, sparse representation of the flow.

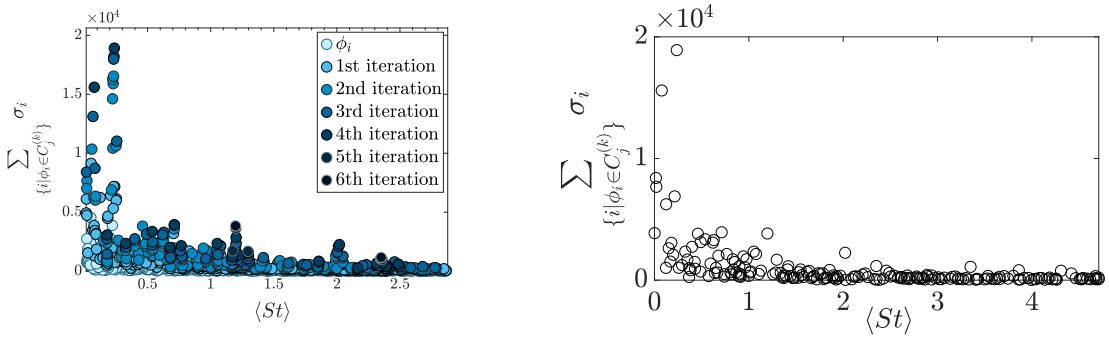



(a) (b) (c)

Figure 3.21: OMD mode ranking statistic $|\alpha_i|$ computed using (a) Equation (3.4); (b) Equation (3.5) with $\gamma = 0$; (c) Equation (3.8).

#### 3.3.4.3 Sparse feature identification using mode clustering.

Algorithm 1 is applied to the extracted modes, with $\epsilon = 0.5$ and using weights $\sigma_i$ computed from (3.8). No spectral cut-off is employed ($s_0 = 0$). Figure 3.22 (a) illustrates the evolution of the clusters after successive algorithm iterations, with Algorithm 1 terminating after six iterations. It is clear that the significant dynamic features emerge more distinctly after successive iterations. In particular, clusters which are expected to correspond to dominant coherent structures with $\langle St \rangle \approx 0.05$ and $\langle St \rangle \approx 0.2$ have converged by the fourth iteration of Algorithm 1.




(a) (b)

Figure 3.22: (a) Cluster ranking against mean Strouhal number for six iterations of Algorithm 1. (b) Ranking of the converged clusters at $\epsilon = 0.5$.

Once all converged clusters, including the OMD modes which remain as singletons, are found and all redundant clusters are cleared, Figure 3.22 (b) shows retained cluster rankings, highlighting again the significance of clusters with $\langle St \rangle = 0.23$ and $\langle St \rangle = 0.08$, consistent with the findings of [142].

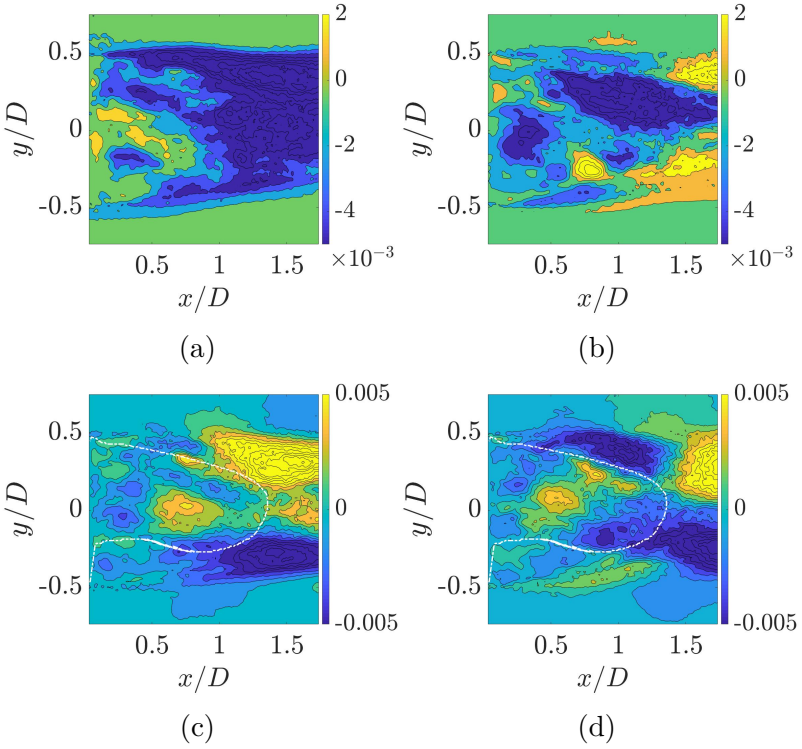
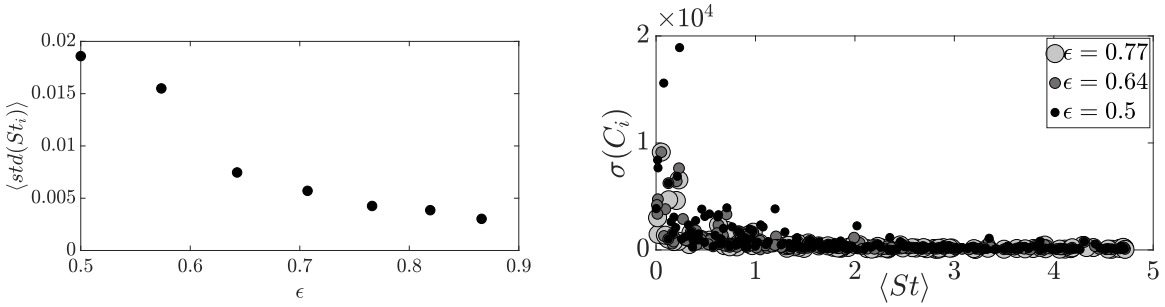



Figure 3.23: Real and imaginary components of representative modes obtained from Algorithm 1. (a-b) bubble-pumping mode at $\langle St \rangle \approx 0.08$; (c-d) vortex shedding mode at $\langle St \rangle = 0.23$.

The streamwise velocity component of the representative modes of these two clusters are shown Figure 3.23. The mode corresponding to $\langle St \rangle \approx 0.08$ in Figure 3.23 (a-b) can be associated with bubble-pumping. In particular, a large coherent region of streamwise velocity perturbation is located predominantly downstream of $x/D > 1$, indicating that this mode can describe the elongation or contraction of the recirculation region in the immediate vicinity of the base. The long timescale $\langle St \rangle = 0.08$ supports this claim. Vortex-shedding is associated with the shorter time-scale $\langle St \rangle = 0.23$ exhibited by the representative mode in Figure 3.23 (c-d). The region of positive perturbation at approximately (1.25,0.4) and the region of negative perturbation at $(1.25, -0.4)$ in Figure 3.23 (c), clearly represent advected vortical structures whose counterparts can be observed in Figure 3.23 (d) at approximately $(1.7, 0.4)$ and $(1.7, -0.4)$.

Finally, we briefly discuss algorithm sensitivity. Figure 3.24 (a) shows, in comparison to analysis of the square cylinder wake in §3.3.2, a more gradual decrease in the within-

cluster variability statistic $\langle \mathrm{std}(St_i) \rangle$ as $\epsilon$ increases. This reflects the inherent increase in complexity of the underlying flow. Figure 3.24 (b) presents the ranking of converged clusters for three different choices of $\epsilon$. While the numerical values of the rankings vary, the significant features identified by Algorithm 1 are consistent. We note that interrogation of the representative modes for the main flow features also shows little variability in their spatial form.




(a) (b)

Figure 3.24: Cluster statistics for flow past an axisymmetric bluff body: (a) Within-cluster Strouhal number standard deviation, averaged across clusters; (b) ranking statistics of converged clusters for cut-off parameter values $\epsilon = 0.5, 0.64$ and 0.77.

Of course, as expected in a complex flow such as the one presented here, the modes in each cluster show a higher degree of variation. As a result, the representative modes will span a smaller portion of the subspace spanned by all of the modes in the subspace. In §5 we will discuss a generalisation of representative modes where more than one mode can represent each cluster. We will also have a more detailed look at the contents of the clusters for the flow past the axisymmetric wake.

Finally, we comment that the application of Algorithm 1 using OMD modes and underlying ranking provided by (3.8), could be adapted to use alternative ranking methods or other modal decomposition techniques. For example, one could use the $\ell_1$ regularisation in DMDsp with a modest value of $\gamma$, and use the resulting amplitudes along with this clustering method. Determining the most appropriate choice of underlying modal analysis with which to promote robust clustering analysis will form the basis of future research.

### 3.3.5 Flow complexity and performance of the clustering algorithm

The data ensembles in §3.3.1, §3.3.2 and §3.3.4 are taken from flows whose dynamics are dominated by advective features. This dynamic similarity, and the significant Reynolds number variation between the datasets, allow us to compare the performance of the mode clustering Algorithm 1 as the dynamic complexity of the underlying flow increases.

Increasing the Reynolds number of a flow typically corresponds to the emergence of a broader spectrum of observed spatial and temporal scales. For low Reynolds number flows, such as the cylinder wake considered in §3.3.1, a small number of dynamical features can often describe, in terms of energetic contribution, a large percentage of a flow's behaviour. At a higher Reynolds number, the percentage energetic contribution of any individual structure, i.e. that described by a single mode, typically decreases. Consequently, it should be expected that automatic algorithmic identification of important dynamic features, e.g., shedding modes for the advective flows considered here, should become more challenging with increasing Reynolds number.

This expectation is confirmed by comparing the initial mode rankings for the cylinder wake at $Re = 60$ in Figure 3.5 (a) with the analogous plot in Figure 3.9 (a) for flow past the square cylinder at higher Reynolds number $Re \approx 10^4$. In both cases, the dominant shedding mode is identified by a clear peak in the significance statistic $\sigma$ observed between $0.12 \leq St \leq 0.14$ for each flow. However, with respect to the identification of harmonics of this mode, while peaks in $\sigma$ are clearly observable in Figure 3.5 (a) for both the first and second harmonics of the flow at $Re = 60$, even the first harmonic of the shedding mode cannot be clearly distinguished by an analysis of $\sigma$ alone in Figure 3.9 (a) for the square cylinder flow at $Re \approx 10^4$. Indeed, in this case there are many modes with non-negligible significance statistics in the range $0.2 \leq St \leq 0.4$.

Despite this difference, there are some notable similarities in the behaviour of the clustering Algorithm 1. First, as indicated in Figure 3.7 for $Re = 60$, and Figure 3.15 at

$Re = 10^4$, non-trivial clusters are formed when $\epsilon \leq 0.95$ in each case. Further, for both flows, if a high value of $\epsilon$ is employed, then non-trivial clusters are found which correspond to higher harmonics of the shedding mode, which can be observed in Figure 3.5 (b) at $Re = 60$, and Figure 3.13 for the flow at $Re \approx 10^4$. In both cases, significant clusters are found corresponding to harmonics which may not have been apparent from an initial observation of the underlying mode significant measure $\sigma$ alone. Importantly, the number of non-trivial clusters identified in each case, and the mean Strouhal number of each cluster, corresponds to the observed maxima in power spectral density plots of Figures 3.4 (b) and 3.9 (b).

A first contrast between the two cases is that, at $Re = 60$, clusters are found corresponding to six harmonics of the dominant shedding mode, while at $Re \approx 10^4$ only the first shedding harmonic is identified with a non-trivial cluster. This performance reflects the increasing dynamic complexity of the underlying flow with the Reynolds number. Still, it may also be explained by the different data-collection methods (DNS at $Re = 60$, PIV for $Re \approx 10^4$) since it is known that the presence of measurement noise can lead to a higher number of damped modes [45] and this may increase the difficulty of extracting coherent dynamical features. Furthermore, comparing the representative modes in Figure 3.6 and Figure 3.14, one can see that the representative modes in Figure 3.14 are noisier and contain a larger range of length scales.

A second difference in the performance of Algorithm 1 is observed as the similarity parameter $\epsilon$ is decreased. Recall that decreasing $\epsilon$ corresponds to relaxing the spatial condition for cluster formation, which can increase the number of modes belonging to each individual cluster. At $Re = 60$, Figure 3.7 (b) shows that the number of significant clusters is stable as $\epsilon$ decreases, a fact that can be explained by the relative dynamic and spatial simplicity of this flow. However, at $Re \approx 10^4$, decreasing $\epsilon$ can have the adverse effect of clustering dynamically dissimilar modes: specifically, the mean mode is clustered with an oscillatory mode at significantly higher $St$ as labelled in Figure 3.15 (b). While at face value this behaviour is not desirable, modifying the clustering algorithm to account

for spectral similarity (see §3.2.5) can mitigate this effect, while at the same time retaining the beneficial effects of using a smaller value of $\epsilon$ in terms of more relaxed spatial similarity criterion for cluster formation. As shown in Figure 3.18, a sufficiently low value of $\epsilon$ coupled with the use of a spectral similarity cut-off enables clear identification, in terms of cluster ranking, of the first shedding harmonic (labelled at $\langle St \rangle = 0.25$). This implementation outperforms the behaviour of the clustering algorithm applied with a high value of $\epsilon$ with no spectral similarity cut-off.

Finally, we note that the beneficial effect of decreasing $\epsilon$ to promote clustering is also observed when Algorithm 1 is applied to data from the three-dimensional bluff body flow of §3.3.4 at $Re = 1.88 \times 10^5$. At this higher Reynolds number, the spectrum of this flow is more broadband, and chaotic variations in the azimuthal position of the plane of symmetry significantly increase the level of experimental noise in this data ensemble in comparison with the flows studied in §3.3.1 and §3.3.2. The combination of these two factors prevents easy identification of dominant modes using standard importance statistics, as shown in Figure 3.21. In any given frequency range, a significant number of modes with nontrivial importance statistics can be observed. Interestingly, when using a spatial similarity of $\epsilon \geq 0.77$, Algorithm 1 does not extract the two expected dominant clusters, associated with bubble pumping and vortex shedding, due to the higher spatial variation between similar modes than in the lower Reynolds number cases of §3.3.1 and 3.3.2. However, as shown in Figure 3.22 (a), lowering the similarity threshold to $\epsilon = 0.5$ allows modes to form the two expected clusters. Finally, due to the increased complexity of this data ensemble, it is interesting to note that many iterations of Algorithm 1, six in this case, are required for convergence.

In conclusion, the algorithm's performance, in terms of forming clusters, worsens as Reynolds number increases. However, such performance degradation may be mitigated by lowering the spatial similarity parameter $\epsilon$ to promote mode clustering. In cases where a lower value of spatial similarity parameter $\epsilon$ is necessary, the algorithm's performance

may be improved by using more complex clustering criteria, such as hybrid considerations of spectral and spatial similarity.

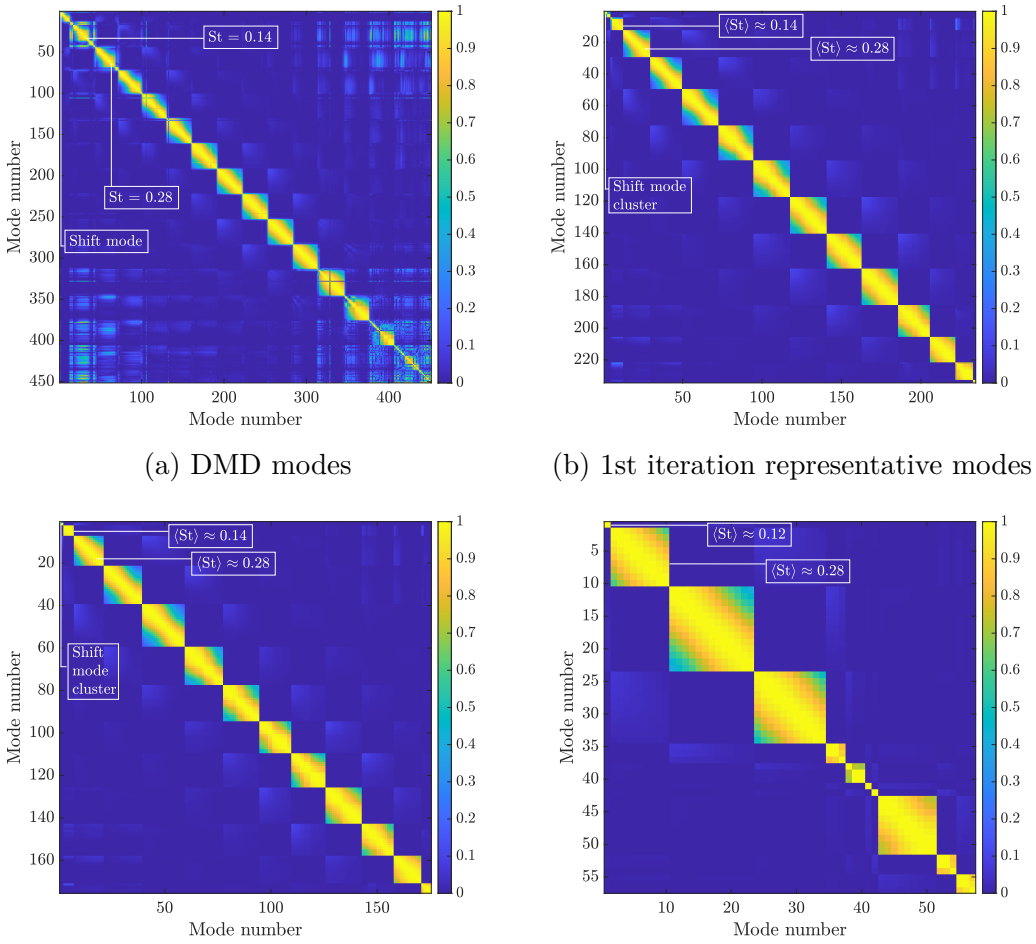
## 3.4 Intermediate results and adjacency matrices

Algorithm 1 extracts clusters by iteratively forming graphs of mode similarities and extracting the maximal cliques of the resulting graph. This section aims to give a more in-depth analysis of the algorithm's cluster-forming behaviour at intermediate (i.e. non-converged) iterations. To do this, we will present the adjacency matrices extracted after each iteration of the algorithm in addition to showing how the clusters coalesce at each sub-iterate of Algorithm 1. Throughout this section, whenever an adjacency matrix is presented, the underlying representative modes will always be sorted according to ascending temporal frequency. In addition, to provide full detail, all adjacency matrices presented here will show the subspace similarity angle statistic $\cos(\theta_{ij})$, as defined in (3.6), rather than simply the binary coefficients $d_{ij}$. With a small abuse of notation, we shall still refer to these matrices as adjacency matrices.

### Flow past a circular cylinder at Re $= 60$.

In §3.3.1, Algorithm 1 was applied to low-Reynolds number, supercritical, flow past a circular cylinder. We presented the converged clusters in Figure 3.5(b) and their representative modes in Figure 3.6. Figure 3.25(a) shows the adjacency matrix for the graph of the standard DMD modes of the data ensemble, with modes ordered by the Strouhal number of their DMD eigenvalue. In this example, $r = 899$ underlying DMD modes are initially computed. As discussed, we use the fact that oscillatory DMD mode shapes are extracted in complex conjugate pairs and show adjacency matrices for the DMD modes with non-zero Strouhal numbers. The spectral similarity criterion was not used in the cluster extraction process, and the spatial similarity cut-off was set to $\epsilon = 0.98$. Similarly,

Figures 3.25(b–d) show the adjacency matrices for the representative modes after the first, second and fifth iterations of Algorithm 1.




(a) DMD modes

(b) 1st iteration representative modes

(c) 2nd iteration representative modes

(d) 5th iteration representative modes

Figure 3.25: Adjacency matrices for Algorithm 1 applied to $r = 899$ DMD modes with cut-off parameter $\epsilon = 0.98$, for flow past a circular cylinder at Re= 60. Colour bars indicate the subspace angle statistic $\cos(\theta_{ij})$. Modes are ordered according to ascending temporal frequency.

The cylinder wake at a low Reynolds number has a well-defined global mode, and a typical DMD analysis will also give mode shapes corresponding to a number of harmonics of the global mode. In line with this expectation, the adjacency matrix for the original DMD modes, Figure 3.25(a), exhibits a clear block-diagonal structure with the Strouhal number of each successive block approximately constant and increasing (block by block)

by the shedding Strouhal number $St = 0.14$. Each block is observed to contain a significant number ($\sim 25$) of modes, each with similar temporal frequency and high spatial similarity.

Figure 3.25(b) shows the adjacency matrix after one iteration of Algorithm 1. Note that in this and subsequent subfigures, only the adjacency matrix of spatial similarities between *representative modes* of the current iteration's clusters are shown. Consequently, the dimension of the presented adjacency matrix decreases with each algorithm iteration and, for example, even the first iteration partitions the original DMD modes into 234 clusters.

It is notable that the block diagonal structure can still be observed in the spatial similarity of cluster representative modes in Figure 3.25(b). However, the number of representative modes in each block (i.e. those with similar Strouhal numbers) is smaller than for the original DMD adjacency matrix of Figure 3.25(a), where each block contains approximately 25 modes. For example, as indicated in Figure 3.25(b), there are only 8 clusters representing structures at the global mode frequency $St \approx 0.14$, there are 17 clusters representing its first harmonic at $St \approx 0.28$, and only two representative modes at low-frequency corresponding to the shift mode. Interestingly, at least for the clusters associated with the global mode and its first three harmonics, the size of each diagonal block (i.e. the number of clusters with very similar temporal frequencies) increases with the Strouhal number.

A possible explanation for this behaviour is that successively higher harmonics of the global mode result in higher spatial variability between the modes representing them, due to the energetic content of these features being dispersed amongst a higher number of modes. Consequently, these modes may initially form a larger number of overlapping, smaller clusters. This observation is supported by the structure of each diagonal block in Figure 3.25(b). It can be seen that the 8 clusters related to the global mode (i.e., the block labelled $St \approx 0.14$) all have representative modes with high spatial similarity (the block is almost uniformly yellow). However, for each subsequent 'harmonic block',

while the modes contained within each block have high spatial similarity, a clear banded structure in similarity statistic is present within each block, suggesting a weaker overall relation between its representative modes.

To explain the higher degree of variability within the blocks associated with the higher harmonics, one must note the root cause of cluster formation amongst DMD modes. The assumption of exponential temporal trajectories leads to many linear modes being needed to describe non-exponential behaviour [10]. Here, the data ensemble contains snapshots of the flow before, during and after reaching its limit cycle. Each mode associated with vortex shedding can best describe only a part of the dataset: certain modes are better suited to describe the initial vortex shedding structure, while others describe the transition onto the limit cycle. It is likely that this need for multiple modes causes increased variability of higher harmonics of the global mode, explaining why higher harmonic blocks contain modes which are more spatially dissimilar.

It should be noted that in Figures 3.25(a)-(c), only one individual cluster with $\langle St \rangle = 0.14$ is labelled, while in (d) all clusters at that frequency band have merged into a single cluster, resulting in a small change in average cluster frequency to $\langle St \rangle = 0.12$. Before further discussing the adjacency matrices of the later iterates of Figures 3.25(c)-(d), it is useful to consider the alternative view given by Figure 3.26 of the cluster formation process across the iterations of Algorithm 1.

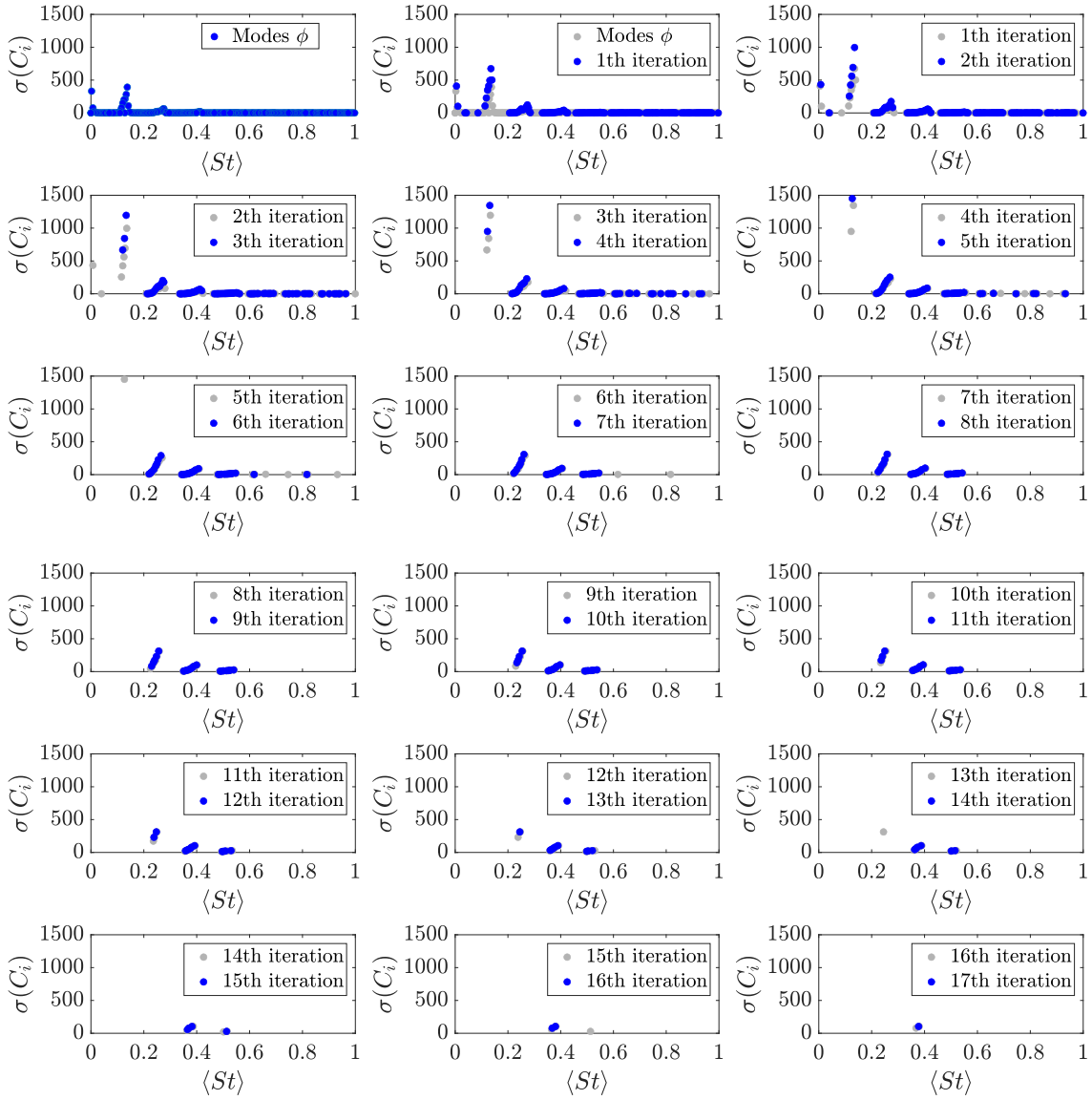



Figure 3.26: Cluster ranking against average cluster Strouhal number for successive iterations of Algorithm 1 applied with cut-off parameter $\epsilon = 0.98$ to a DMD mode ensemble with $r = 899$ for flow past a circular cylinder at Re $= 60$.

Figure 3.26 shows the 17 iterations required for convergence, presented from the top left (the original DMD modes) to the bottom right in Figure 3.26. At each iteration, for clarity, only two sets of modes are shown: the representative modes at the $(n-1)^{\text{th}}$ iterate in grey; and the clusters formed from these modes at the $n^{\text{th}}$ iterate in blue. From the $1^{\text{st}}$ to the $5^{\text{th}}$ generations, there are a number of highly-ranked clusters in a narrow frequency band near the global mode frequency $St \approx 0.14$. By the $5^{\text{th}}$ iteration, these

have converged into a single cluster. Note that converged clusters (i.e. those whose representative mode does not meet the spatial similarity threshold with any other cluster representative) are not plotted in any higher generation subfigure of Figure 3.26, meaning that the dominant global mode cluster is not shown from the $6^{\text{th}}$ generation plot onwards.

In subtle contrast to the global mode clusters, while clear groups of clusters can be observed forming close to the harmonic frequencies $St \approx n \times 0.14$, $n \in \mathbb{N}$, for higher harmonics, the frequency range of each cluster group is initially wider. For example, for representative modes whose frequency range is close to that of the first global harmonic $\langle St \rangle \sim 0.28 = 2 \times 0.14$, a fully distinct grouping only occurs at the $3^{\text{rd}}$ generation and persists until finally coalescing at the $13^{\text{th}}$ generation. That more iterates are required to converge this underlying flow feature confirms the previous observation of increased spatial mode variability. Nonetheless, it is interesting to note that based on the original DMD rankings (top-left panel of Figure 3.26), the first harmonic is only weakly distinguishable, yet, clear clusters related to this harmonic are formed with subsequent generations of Algorithm 1. Furthermore, the second (and higher) harmonics are barely distinguishable from the original DMD rankings but, again, clear clusters of modes are formed in distinct frequency bands as can be seen in Figure 3.26 from the $7^{\text{th}}$ generation and above.

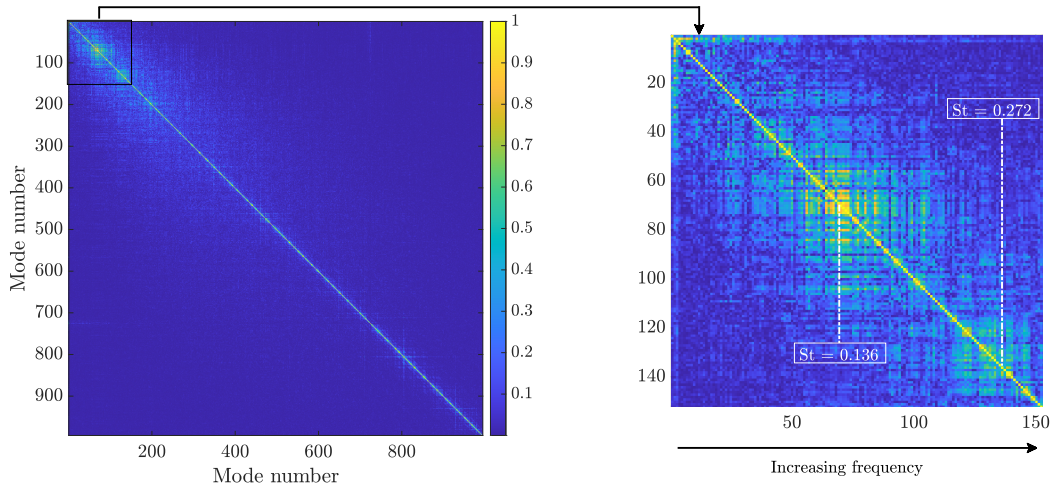
Returning finally to consider the adjacency matrices at higher iterations of Algorithm 1, Figures 3.25 (c) and (d) show the spatial cluster similarity for the iterates at which the shift mode cluster is fully converged (the $2^{nd}$ generation) and after the global mode cluster is fully converged (the $5^{th}$ generation). The converged representative modes of these features are shown in Figure 3.6. Between these generations, the size of the block corresponding to the global modes reduces from 6 modes in the $2^{\text{nd}}$ generation to a single mode in the $5^{\text{th}}$ generation. The size of the first harmonic block decreases from 14 to 6 across the same number of iterations, reflecting the slower convergence observed in Figure 3.26.

While this data ensemble has the advantage of clearly illustrating the cluster formation and coalescence process, studying more complex examples for which the underlying DMD

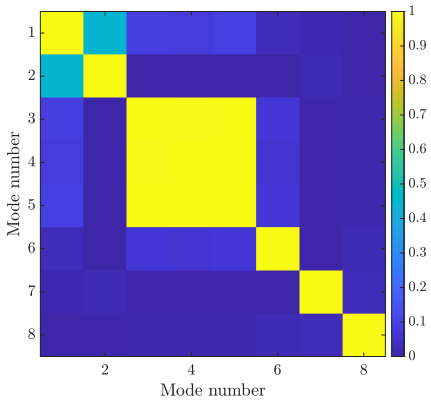
modes cannot themselves be clearly partitioned into well-defined clusters, can provide a more compelling motive for the use of Algorithm 1.

### Flow past a square cylinder at Re $\approx 10^4$.

In §3.3.2, the representative modes obtained using Algorithm 1 applied to a data ensemble of flow past a square cylinder at Re $= 10^4$ were presented in Figure 3.14. The original 2000 modes were extracted using DMD, and Algorithm 1 was applied at $\epsilon = 0.94$, without the spectral similarity filter. Figure 3.27 (a) shows the adjacency matrix of the original $10^3$ conjugate pairs of DMD modes of this ensemble, while Figure 3.27 (b) shows the adjacency matrix for the representative modes of clusters obtained by the first iteration of Algorithm 1.




(a) Original modes $\phi$




(b) First iteration representative modes

Figure 3.27: Adjacency matrices for the flow behind a rectangular cylinder at Re= $10^4$ corresponding to an application of Algorithm 1 with cut-off parameter $\epsilon = 0.94$ to a DMD mode ensemble with $r = 2000$ modes. Colour bars indicate the subspace angle statistic $\cos(\theta_{ij})$. Modes are ordered according to ascending temporal frequency. (a) adjacency matrices for the underlying DMD modes (b) adjacency matrix after one iteration of Algorithm 1.

In Figure 3.27 (a), one can notice a significant reduction in the regularity of the spatial similarity structure compared to that observed in Figure 3.25 (a) for a low Reynolds number bluff body flow. Consequently, it is to be expected that extracting clusters of similar modes, or even identifying dominant modes, should be more challenging for this data ensemble. Nonetheless, as indicated on the right-hand panel of Figure 3.27 (a), there are noticeable groups of modes with increased spatial similarity with Strouhal numbers

close to the expected global mode frequency at St $\approx 0.135$, and that of its harmonic at St $\approx 0.27$.

Moreover, Figure 3.27 (b) shows that after only one iteration, Algorithm 1 identifies three clusters of similar modes (listed as modes $3, 4$ and $5$ in Figure 3.27 (b)) with $\langle St \rangle \approx 0.135$, in addition to one cluster at $\langle St \rangle \approx 0.27$ (listed as mode 6 in the same figure) and another at $\langle St \rangle \approx 0$, which may correspond with a low-frequency bubble-pumping mode. Algorithm 1 terminates after the second iteration, with only the three representative modes related to the global mode (listed as $3, 4$ and $5$) being combined into a new cluster. All other clusters are viewed as distinct by Algorithm 1 and are converged at the first iteration of the algorithm.

The behaviour of Algorithm 1 is, therefore, somewhat counter-intuitive in that, at least for this example, a more dynamically complex data ensemble has led to fewer clusters forming in comparison to a less complex flow at a lower Reynolds number. However, this behaviour can be explained by the fact that at a higher Reynolds number, there is increased flow intermittency. Furthermore, the data sampling rate of 50Hz limits the range of time-scales for which coherent features can be accurately extracted. Consequently, lower energy flow structures, such as higher global-mode harmonics, may no longer be statistically identifiable from a given finite-length data ensemble. This observation corresponds to the smaller bandwidth of the adjacency matrix of Figure 3.27 (a) compared to the persistent block-diagonal structure of Figure 3.25. Hence, while fewer non-trivial clusters are formed for the case of the square cylinder, Algorithm 1 does efficiently form clusters of dynamically important modes. These observations are confirmed in Figure 3.28, which shows the change in cluster ranking over the two iterations of Algorithm 1. It can be seen that the ranking statistic of the converged global mode cluster (the blue marker in the right-hand panel, formed from the three global mode clusters at the first iteration) is over double that of the dominant DMD mode.

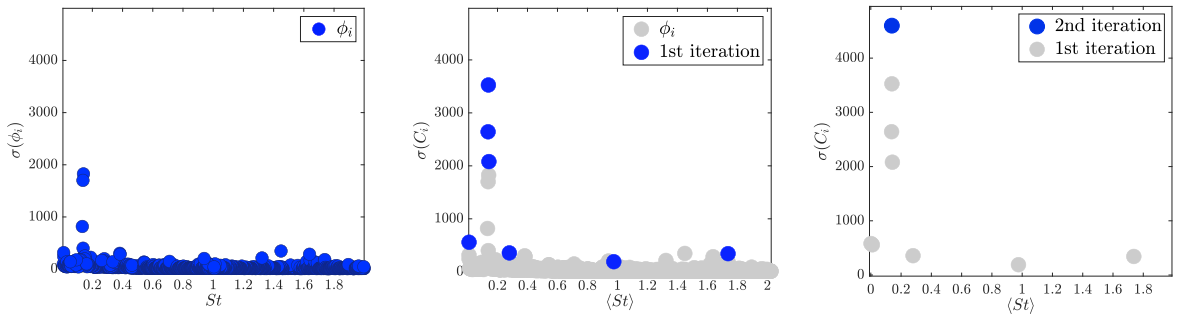


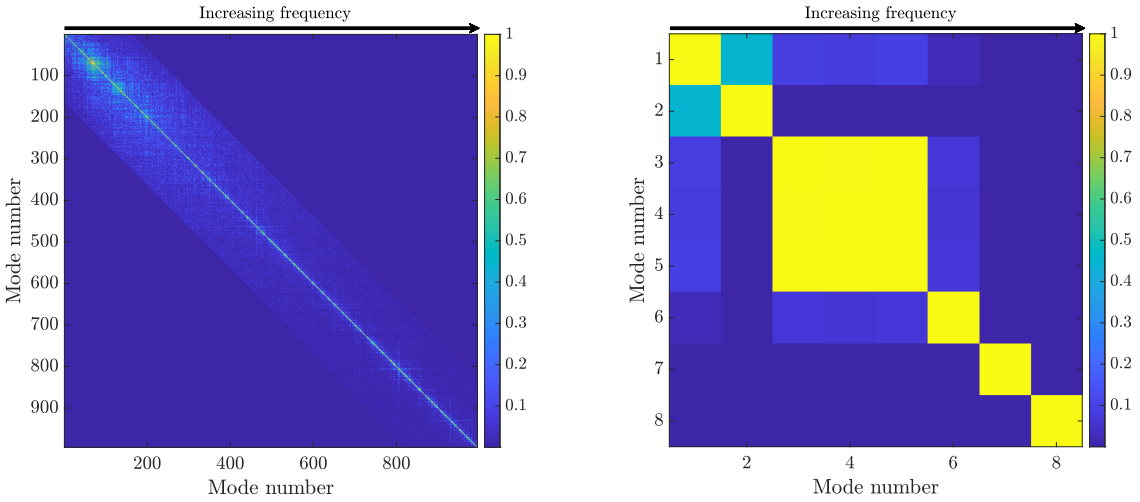

Figure 3.28: Cluster rank against average Strouhal numbers of clusters over the first and second iterations of Algorithm 1 applied to $r = 2000$ DMD modes with $\epsilon = 0.94$, for flow past a rectangular cylinder at Re$= 10^4$.

### The effect of using a spectral similarity cut-off

A further interesting property evident in Figure 3.27 (a) is that while the adjacency matrix is sparse, it contains many entries with negligible but non-zero values. This leads to numerical inefficiencies in the Bron-Kerbosch algorithm used for extracting the maximal cliques. Moreover, in §3.3.2.4 it was shown that depending on the choice of the spatial similarity cut-off $\epsilon > 0$, the mean mode can be clustered with a spurious high-frequency mode. This is a statistical anomaly arising from the fact that both modes are real-valued, meaning that their (zero) imaginary parts are highly correlated. To avoid this, we introduced a spectral similarity cut-off.

To demonstrate the effect of applying such a spectral similarity filter, Figure 3.29 shows the adjacency matrices with the spectral filter applied at $s = 0.75$ and $p = 2$ (see §3.2.5 for a detailed explanation of the parameters), leading to a clearly noticeable removal of non-zero mode similarities away from the leading diagonal. This implies that any connections with modes of much higher or lower temporal frequency is ignored. The fact that Figure 3.27 shows that the most significant connections are close to the main diagonal of the adjacency matrix, i.e. the modes with similar frequencies, suggests that apart from reducing the possibility of spurious cluster groupings, the effect of this filter on the final extracted clusters is not significant. Indeed, comparing the adjacency matrix of the 1$^{\text{st}}$

generation's converged clusters without the spectral similarity filter in Figure 3.29 (b) to those in Figure 3.27 (b) with such a filter, one can see that for this flow, at $\epsilon = 0.94$, this effect is indeed negligible.

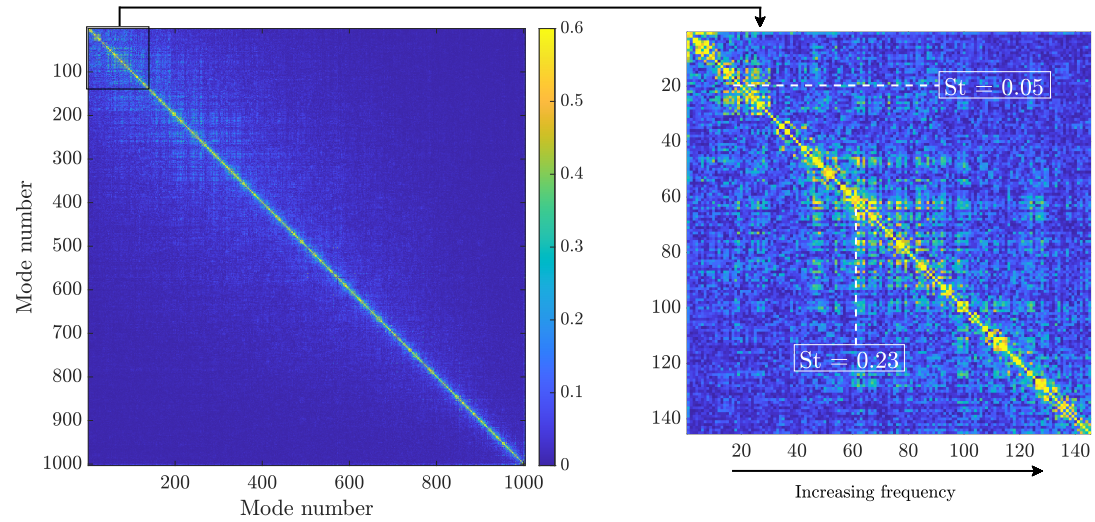



(a) Original DMD modes $\phi$ (b) First iteration representative modes

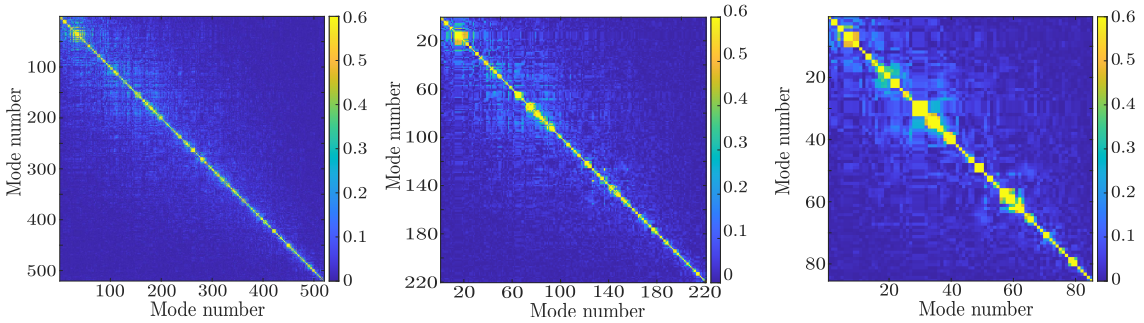
Figure 3.29: Adjacency matrices for the flow behind a rectangular cylinder, after applying the spectral filter with parameters $s = 0.75$ and $p = 2$. Colour bars indicate the subspace angle statistic $\cos(\theta_{ij})$. Modes are ordered according to ascending temporal frequency.

## Flow past an axisymmetric bluff body at Re $= 1.88 \times 10^5$.

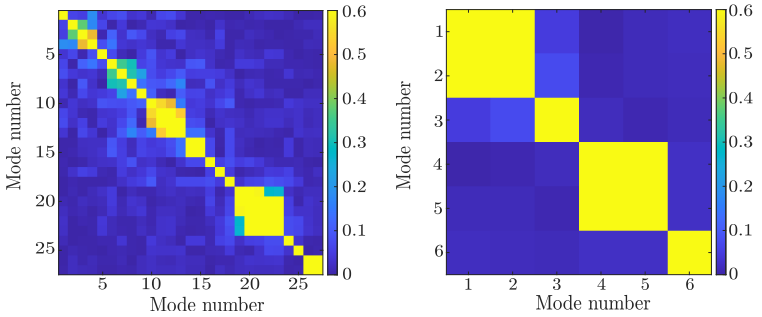
We now consider the case of turbulent flow past a bullet-shaped bluff body at Re $= 1.88 \times 10^5$, considered earlier in §3.3.4. There, OMD was applied to 2732 velocity field snapshots to extract $r = 2000$ modes. Subsequently, Algorithm 1 was applied with spatial similarity cut-off $\epsilon = 0.5$ to extract clusters of similar modes. Figure 3.32 shows the adjacency matrices for this application of Algorithm 1. Importantly, the lower signal-to-noise ratio for this data ensemble (in comparison to the lower Reynolds number examples considered previously) leads to lower absolute values of mode similarity, while the broadband nature of the flow leads to significantly less well-defined frequency bins. However, over successive iterates of Algorithm 1, it can be observed in Figure 3.30 that the adjacency matrices become more regular.




(a) Original OMD modes $\phi$




(b) First iteration modes (c) Second iteration (d) Third iteration




(e) Fourth iteration (f) Fifth iteration

Figure 3.30: Adjacency matrices for the flow behind an axisymmetric bluff body corresponding to: (a) the underlying OMD modes; (b-f) the five successive iterations of Algorithm 1 applied to $r = 2000$ OMD modes with cut-off parameter $\epsilon = 0.5$. Colour bars indicate the subspace angle statistic $\cos(\theta_{ij})$. Modes are ordered according to ascending temporal frequency.

To aid the interpretation of the adjacency matrices, Figure 3.31 also shows how clusters coalesce and form at each iteration of Algorithm 1, by plotting the mean cluster Strouhal number against the cluster ranking statistic.

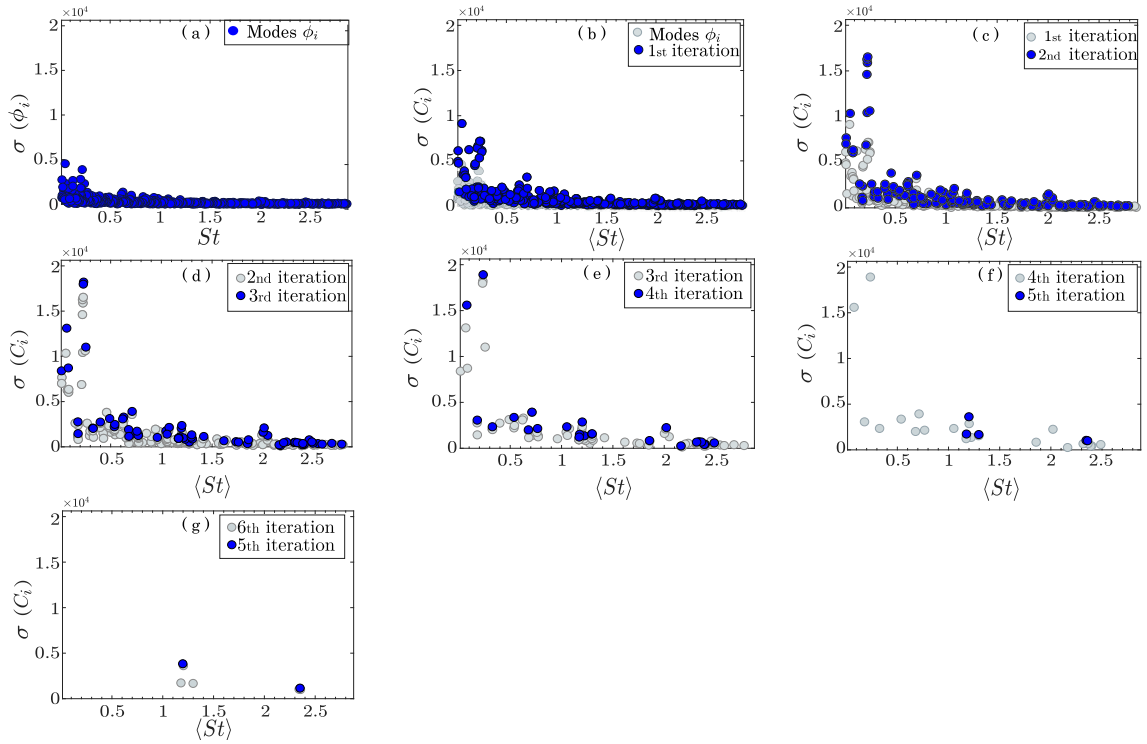
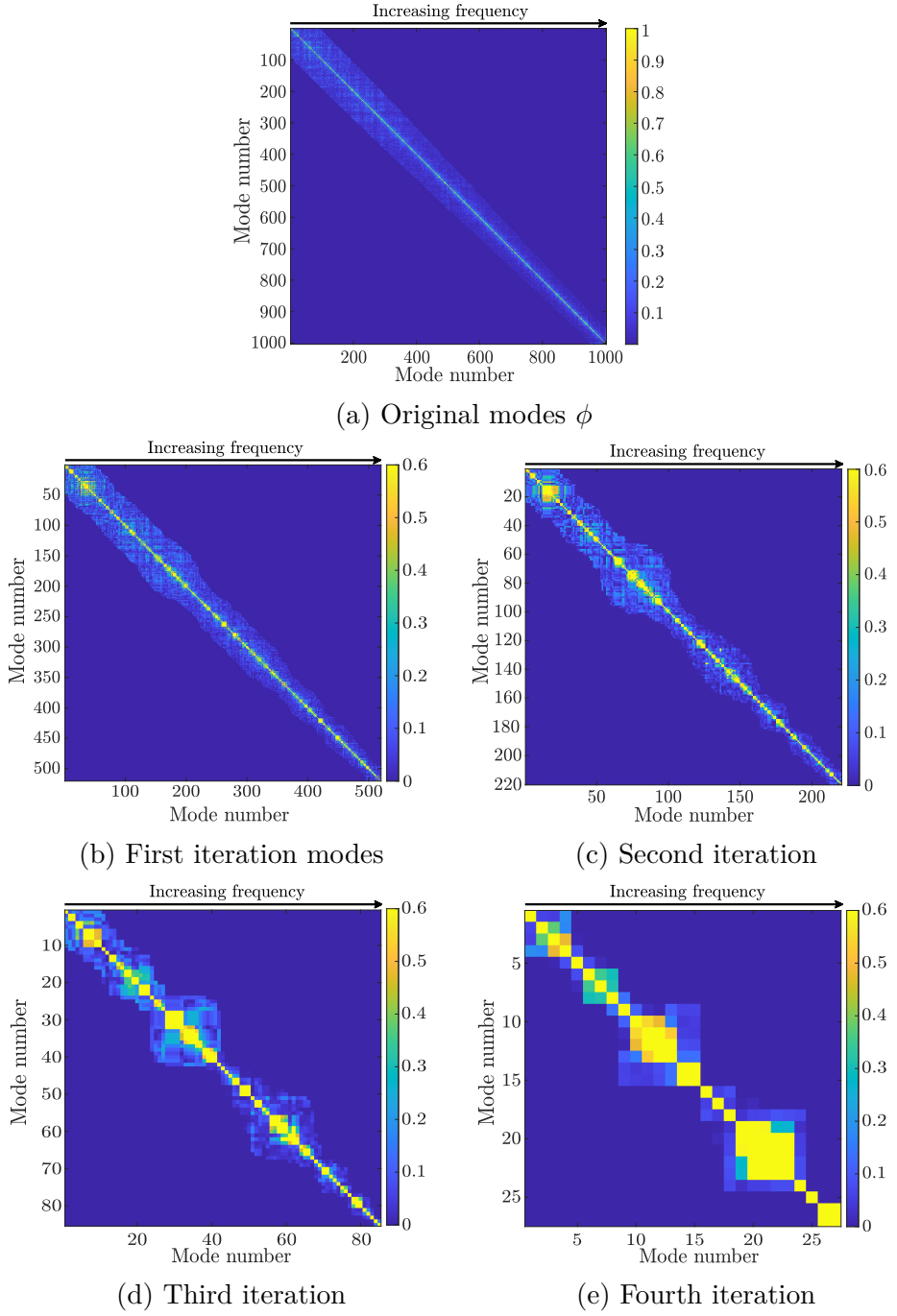



Figure 3.31: Cluster ranking against average Strouhal numbers. Shown for (a) the underlying $r = 2000$ OMD modes; and (b-g) the six iterations of Algorithm 1 applied with cut-off parameter $\epsilon = 0.5$.

As an indicative example, we now track the formation of the vortex-shedding cluster in detail. For the initial set of OMD modes, it can be seen in Figure 3.31 (a) that there are many significant modes with frequencies close to the global mode frequency $St \approx 0.2$. This corresponds to the incoherent block of higher subspace angle similarity, formed approximately from mode numbers 40 to 100, which can be observed in Figure 3.30 (a). Next, after the first iteration, Algorithm 1 extracts ten clusters with $\langle St \rangle \approx 0.2$, which can be seen in Figure 3.31 (b). The representative modes of these clusters form the small block of high mode-similarity around the mode number 20 in Figure 3.30 (b). In the second iteration, these clusters coalesce to form a block of seven highly similar clusters

around mode number 15 in Figure 3.30 (c). In a similar way, the following two iterations see the algorithm further consolidate the underlying modes into one representative mode at $\langle St \rangle = 0.23$. The representative mode for this cluster is seen as mode number 5 in Figure 3.30 (e). As expected, this representative mode does not meet the similarity threshold ($\epsilon = 0.5$) with any other representative mode, and as a result, the cluster converges at the fourth iteration. The converged cluster contains 22 modes, all of which belong to the original sparse block of higher similarity between mode numbers 40 to 100 in Figure 3.30 (a).

For completeness, Figure 3.32 shows the same adjacency matrices, after applying the spectral filter at $s = 0.75$, $p = 2$. Notice that the filter eliminates many negligible similarities and leads to better-defined clusters, although, as in the example of flow past a square cylinder, it is found that using the spectral similarity filter does not significantly alter the resultant modes in each cluster. Indeed, for the third and fourth iterations at least, there are very clear similarities in the structure of the adjacency matrices without spectral cut-off in Figure 3.30 (d-e), and those in Figure 3.32 (d-e) with the spectral cut-off applied. We note that since the vortex shedding and bubble-pumping clusters converge at the fourth iteration, regardless of whether or not the spectral similarity cut-off is applied, we choose to show in Figure 3.32 only the first four iterations of the algorithm.




(a) Original modes $\phi$

(b) First iteration modes

(c) Second iteration

(d) Third iteration

(e) Fourth iteration

Figure 3.32: Adjacency matrices for the flow behind an axisymmetric bluff body. Shown for (a) the underlying $r = 2000$ DMD modes and (b-e) the first four iterations of Algorithm 1 applied with cut-off parameter $\epsilon = 0.5$ and with a spectral filter with parameters $s = 0.75$ and $p = 2$. Colour bars indicate the subspace angle statistic $\cos(\theta_{ij})$. Modes are ordered according to ascending temporal frequency.

### 3.4.1 The effect of the decomposition dimension r

One of the foremost motivating factors in developing the post-processing algorithm presented in this chapter was that the outputs of modal decomposition techniques, particularly DMD variants, can vary significantly based on arbitrary parameters such as the decomposition dimension $r$. Indeed, the observation from Figure 3.1 that the spectral content of the entire mode ensemble obtained using DMD is directly influenced by $r$ provided initial motivation for developing Algorithm 1. The idea is that instead of reducing $r$ to select a small number of dynamically important modes, one should seek to post-process (or cluster) the hopefully more accurate modes obtained with a large value of $r$. From this viewpoint, it can be argued that Algorithm 1 is designed to be applied with as high a value of $r$ as possible.

Despite this, it is of interest to determine whether Algorithm 1 behaves robustly when applied to initial mode ensembles formed by selecting different values of the DMD mode reduction parameter $r$. To do this, we consider the example of flow past a circular cylinder at $\mathrm{Re} = 60$ studied in §3.3.1. Previously, Algorithm 1 was applied to $r = 899$ snapshots of this flow. Here we apply DMD at different truncation values and then post-process the resultant modes using Algorithm 1. In all cases, the spatial similarity cut-off was kept at $\epsilon = 0.98$, and the spectral similarity cut-off was not used. Lower numbers of underlying DMD modes of $r = 450$ and $r = 220$ were chosen to be roughly around half and a quarter of the original DMD truncation dimension.

Figure 3.33 shows the mean Strouhal number against cluster rank for clusters formed by Algorithm 1 for the three considered values of $r$. It can be seen that the most significant clusters remain similar for the wide range of underlying decomposition dimensions $r$ considered. Interestingly, the Strouhal numbers and the significance levels remain similar regardless of the cut-off parameter $r$. This indicates that the converged clusters contain similar modes.

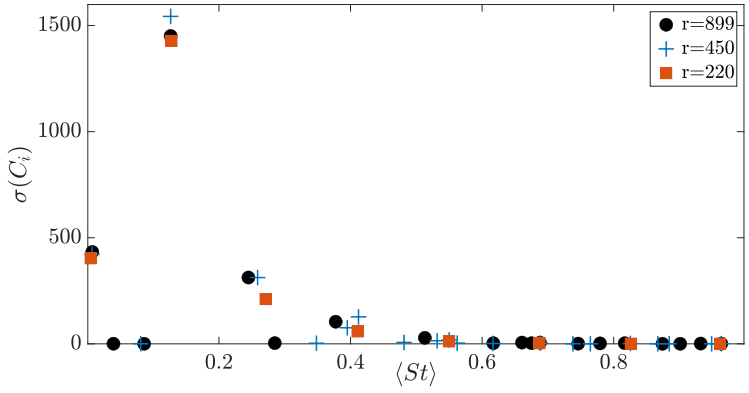



Figure 3.33: Cluster ranking against average Strouhal number of the converged clusters obtained from the application of Algorithm 1 to an underlying DMD mode ensemble with $r = 220, 450$ and $r = 899$ modes.

## 3.5 Conclusion

A method of post-processing dynamic modes to extract a sparse set of dynamically relevant clusters of modes was presented. The method groups modes into clusters, using the graph-theoretic notion of maximal cliques, and iteratively combines similar clusters to form clusters of higher cardinality. Through this iterative process, a sparse representation of the flow can be found until the remaining clusters are distinct. Each cluster can then be represented by a single mode, facilitating analysis of the coherent flow structure it represents. Algorithm 1 has been benchmarked on the much-studied wake of a two-dimensional circular cylinder before being applied to more challenging experimental data sets of turbulent bluff-body flows. In each case, features are shown to be extractable corresponding to accepted coherent wake structures. For the more dynamically complex flows, clustering was shown to (i) promote sparse flow representations with a greater dynamic variety than existing methods and (ii) extract coherent flow structures which were not clearly present when using classical mode ranking methodologies.

Chapter 4

# Output-Regulated Optimal Mode Decomposition

Recent advances in modal decomposition techniques, such as Dynamic Mode Decomposition (DMD), facilitate the extraction of coherent structures and reduced order models of an underlying dynamical system from high-dimensional data ensembles. However, the reduced order models that DMD and other similar techniques extract are typically focused on approximating an underlying dynamical system without consideration of possible inputs and outputs to it. In this chapter, we consider the problem of reduced-order system identification for systems where only a limited number of system outputs can be used in real-time.

We present an algorithm that simultaneously extracts a linear reduced order model for the underlying dynamics and an output matrix which maps a high-dimensional state to a low-dimensional vector of measurements. The method identifies an appropriate low dimensional subspace, which is appropriate for approximating the dynamics of the system, and suitable for being mapped to the measured output signal. As a result, the subspace of data spanned by the modes extracted by the presented algorithm corresponds better to the measurement signal than those computed via a standard methodology (e.g. DMD or OMD). This feature is of importance for applications to high-dimensional systems and, in

particular, those for which sensor information may be localised spatially. In this setting, typically encountered in fluid mechanics, it may be the case that less energetic features of the underlying dynamics may be better correlated with a given sensor as a result of spatial proximity. From the perspective of generating observable ROMs, it is therefore important to consider this issue in the extraction of dynamic modes. This motivates the study presented in this Chapter.

## 4.1 Introduction

Consider state $(x_i)_{i\geq 1} \subset \mathbb{R}^m$ and output $(y_i)_{i\geq 1} \subset \mathbb{R}^\ell$ sequences arising from a system

$$\begin{aligned} x_{i+1} &= f(x_i) \\ y_i &= h(x_i) \end{aligned} \tag{4.1}$$

where $f : \mathbb{R}^m \to \mathbb{R}^m$, and $h : \mathbb{R}^m \to \mathbb{R}^\ell$ are continuous functions. We consider the problem of estimating the system dynamics, $f$ and $h$, from ensembles of state and measurement data, in systems where there are no control inputs. In particular, assume that $N+1 \in \mathbb{N}$ snapshots $\mathcal{X} = \{x_i\}_{i=1}^{N+1} \subset \mathbb{R}^m$ of the system's state are available for analysis and denote the corresponding output sequence $\mathcal{P} = \{y_i\}_{i=1}^{N} \subset \mathbb{R}^\ell$.

As in §3, we concentrate on the cases where the underlying state dimension is high in comparison to the number of snapshots available for analysis and the dimension of the output signal $(y_i)_{i\geq 0}$. That is, we assume that $p \gg N \gg \ell$. This is valid in fluids examples where, typically, the state snapshot $x_i \in \mathbb{R}^p$ is an instantaneous discretised velocity field. For example, $x_i$ may contain the fluid's velocity components at a large number $p \gg 1$ (typically $p \geq 10^4$) of spatial locations in the flow domain. This is the case both for direct numerical simulations of the Navier-Stokes equations, and for data ensembles obtained from particle image velocimetry. Due to computational cost and experimental complexity, the number of spatial data points required to accurately describe the flow is significantly greater than the number of snapshots that can practicably be collected. As

a result, the dimensions of the collected snapshots dictate that $p \gg N$. Measurement signals typically arise from a small number of velocity, shear-stress (velocity derivative) or pressure sensors placed at particular locations in or on the boundaries of a flow. As a result, it is also the case that $p \gg \ell$. Furthermore, one may reasonably expect that the available measurements may only correlate well with a, possibly low-dimensional, subspace of state-space.

The approach taken towards reduced-order modelling in this Chapter will follow the philosophy of extracting *coherent structures* as motivated in Chapter 3. As was argued there, many techniques have been developed for data-driven identification of coherent spatio-temporal structures, whose use can enable low-order approximations of high-order dynamics (i.e. to identify approximations to the mapping $x_{i+1} = f(x_i)$). However, none of the methods described so far enables the identification of structures (and models) at the same time as taking into account information about an output sequence $(y_i)_{i\geq 0}$. In this chapter, we will extend the DMD and OMD algorithms to address this deficiency.

To do this, we begin with the simplest approach of linear modelling which underpins DMD and the majority of techniques described in §3. In particular, we will seek to find matrices $\mathcal{M} \in \mathbb{R}^{p\times p}, \mathcal{C} \in \mathbb{R}^{\ell\times m}$ for which

$$x_{i+1} \approx \mathcal{M}x_i, \quad \text{and} \quad y_i \approx \mathcal{C}x_i,$$

thus extending the approach underpinning Dynamical Mode Decomposition described in §2.3.

To approximate $\mathcal{M}$ and $\mathcal{C}$ from the available data ensembles, $\mathcal{X}$ and $\mathcal{P}$, we follow the philosophy of DMD and construct the state matrices

$$X = \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ x_1 & x_2 & \cdots & x_N \\ \downarrow & \downarrow & & \downarrow \end{pmatrix} \in \mathbb{R}^{p\times N}, \qquad X' = \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ x_2 & x_3 & \cdots & x_{N+1} \\ \downarrow & \downarrow & & \downarrow \end{pmatrix}, \in \mathbb{R}^{p\times N}$$

whose columns are shifted by one discrete time-step which, as before, we assume is a constant $\Delta t > 0$. In addition, we also assemble the measurable output sequence into

$$P = \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ y_1 & y_2 & \cdots & y_N \\ \downarrow & \downarrow & & \downarrow \end{pmatrix} \in \mathbb{R}^{\ell \times N}.$$

Again, we are met with the same challenge faced when extracting DMD modes: for a high-dimensional state $p \gg N \gg 1$, even fitting for $\mathcal{M}$ alone by solving the naive least-squares optimisation problem $\min_{\mathcal{M}} \|X' - \mathcal{M}X\|^2$ is over-determined. It is therefore necessary to reduce the dimension of the fitted model and this is performed by introducing an encoding step that projects the high-dimensional state to a lower-dimensional state.

We now follow the philosophy of OMD from §2.4 and employ a linear projection matrix $L \in \mathbb{R}^{r \times p}$ for which $L^\top L = I$ and $r \ll p$. Subsequently, model fitting for both the state evolution and the output equation can be achieved by constructing matrices $M \in \mathbb{R}^{r \times r}$ and $C \in \mathbb{R}^{\ell \times r}$ such that

$$X' \approx LML^\top X, \quad \text{and} \quad P \approx CL^\top X, \tag{4.2}$$

with reduced order models $f(x) \approx \mathcal{M}x = LML^\top x$ and $h \approx \mathcal{C}x = CL^\top x$ corresponding to the underlying dynamics (4.1). Identification of such a model, shown schematically in Figure 4.1, implies a reduced-order state $v_i = L^\top x_i$ whose evolution is contained in the $r$-dimensional subspace $\mathrm{Im}(L) \subset \mathbb{R}^p$. Consequently, at a schematic level, the aim of this Chapter is to identify a projection-matrix, state-matrix, output-matrix triple $\{L, M, C\}$ which approximates the available data according to (4.2).

Identifying the triplet $\{L, M, C\}$ *simultaneously* is a challenging task, and existing methodologies are usually focused on the identification of only one decision variable in isolation. For example, if one applies the Dynamic Mode Decomposition (DMD) algorithm from §2.3, the $M$ is constructed by first computing the rank-reduced singular-value

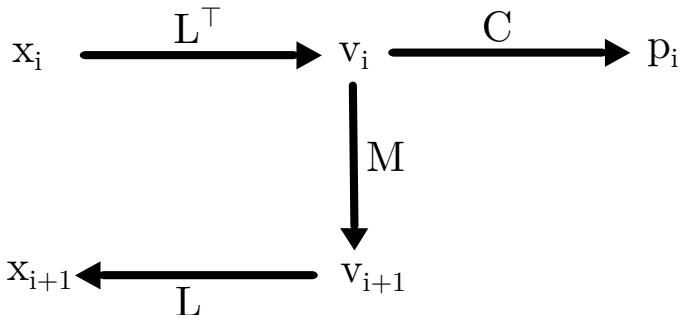



Figure 4.1: Transformation matrices corresponding to the output system (4.1) with full-order state $x_i$, reduced-order state $v_i$, and measurement data $p_i$. The aim is to identify $\{L, M, C\}$.

decomposition $U\Sigma V^\top = X$ to obtain $U \in \mathbb{R}^{p\times r}, \Sigma \in \mathbb{R}^{r\times r}, V \in \mathbb{R}^{p\times r}$, then fixing the projection matrix as $L = U$ and subsequently obtaining $M \in \mathbb{R}^{r\times r}$ by solving

$$M = \operatorname{argmin}_A \left\| X' - UAU^\top X \right\|_F^2 = U^\top X' V \Sigma^{-1}. \tag{4.3}$$

It is important to note that while an output mapping $C$ can be fitted as a post-processing step, there is no guarantee that the implied subspace $\mathrm{Im}(U)$ over which the DMD model $v_i = L^\top x_i$ evolves is appropriately correlated with the observation process $y_i = h(x_i)$. Many variants of DMD focus on modifying the model $UMU^\top : x_i \mapsto x_{i+1}$ to improve the approximation of the state for more challenging modelling scenarios (e.g. in the presence of additive noise or system nonlinearity) —see [75, 45, 166, 130]— and do not explicitly consider the inclusion of an output process.

Before introducing our new approach in §4.2, we now briefly discuss some related data modelling approaches which solve similar, yet different, problems and explain the challenges of extending them to the problem we consider in this chapter.

### 4.1.1 Related approaches for control system identification

There do exist classical methods of reduced-order-modelling which base their approach to order-reduction on broader dynamical considerations, such as the observability and

controllability. We have reviewed a number of these techniques in §2.2 and here recall that for a discrete system, stable, and linear control system

$$\begin{aligned} x_{i+1} &= Ax_i + Bu_i \\ y_i &= Cx_i \end{aligned} \tag{4.4}$$

where $A$, $B$ and $C$ are all *known*, balanced truncation can be used to calculate a reduced-order model that ignores the parts of the state space that are both difficult to reach and difficult to observe [123]. Balanced truncation finds a transformed state $z_i = Tx_i$ using the transformation matrix $T$ which leads to the transformed dynamics system

$$z_{i+1} = T^{-1}ATz_i + T^{-1}Bu_i,$$

for which the observability and controllability Gramians are identical. Truncating the matrices $T$ and $T^{-1}$ will then lead to the *balanced* ROM. To find the transformation matrix $T$, [123] relies on calculating the Gramians, $W_O$ and $W_C$, of the system as defined by (2.3) and (2.4). However, when only snapshot and output data are available, the matrices $A, B, C$ are not known and the respective Gramians are not possible to compute, making balanced truncation impossible to apply directly in a data-driven setting.

Data-driven methods do exist which seek to approximate the benefits of balanced truncation, such as balanced POD [145] described in §2.1, but these require snapshot information relating to the impulse response of both the system itself and its adjoint. The need for the adjoint simulations means that they cannot be applied to experimental data in which no a priori information about $f$ or $h$ is available. Similar issues exist if one attempts to apply classical methods such as ERA [87], which rely on knowledge of the underlying system's impulse response and so will also be available for analysis of output-only systems of the form (4.1).

DMD with control (DMDc) [137] is a more recent extension of DMD that concentrates on systems with known input signal $u$ and allows for simultaneous modelling of a dynam-

ics matrix $A$ and an input matrix $B$, based on snapshots data drawn from states $x_i$ and inputs $u_i$ only. This modification of DMD seeks to satisfy an equation of the form

$$X' \approx AX + B\mathfrak{U} \tag{4.5}$$

where $\mathfrak{U}$ contains snapshots of input $u$. By computing two Singular Value Decompositions

$$\begin{bmatrix} X \\ \mathfrak{U} \end{bmatrix} = \begin{bmatrix} \tilde{U}_1 \\ \tilde{U}_2 \end{bmatrix} \tilde{\Sigma}\tilde{V}^\top, \qquad X' = \hat{U}\hat{\Sigma}\hat{V}^\top$$

DMDc finds the low dimensional matrices

$$M = \hat{U}^\top X'\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}_1^\top\hat{U} \quad \text{and} \quad \tilde{B} = \hat{U}^\top X'\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}_2^\top$$

for which the dynamics of the high-dimensional data can be approximated by a reduced-order system of the form $z_{i+1} = Mz_i + \tilde{B}u_i$. While this is the control-input-only analogue of the problem we will address in this chapter, it is far from obvious how one can convert the DMDc approach to solve the mode identification problem for output-only systems of the form (4.1)

## 4.2 Methodology

As evident from the above discussion, a common feature of DMD-inspired methods is their use of POD (or SVD) as the means to project high-dimensional snapshot data onto a lower-dimensional subspace. In the context of Figure 4.1, these algorithms assume that the optimal matrix $L$ is the matrix of POD modes of the state. As explained in §2.1, POD modes, $U$, of the state snapshots, accurately replicate those snapshots through minimising $\left\|(I - UU^\top)X\right\|_F^2$. If we use the POD modes of the state for a mapping between the output and the low-dimensional state, $P = CU^\top X$ presupposes that the measurements $P$ must be mapped to the subspace where most of the energy of $X$ is contained. However,

this subspace may not be optimal for estimating $P$. On the other hand, concentrating on replicating the measurement snapshots will lead to a projection matrix $L$ that may not lead to an optimal $LML^T : x_i \to x_{i+1}$. To create a ROM where the accuracy of the reconstructed snapshots and the mapping of the state $x$ to the measurement $y$ are balanced, there is a need for a better projection method.

For this reason we appeal here to the OMD methodology, described in §2.4. Since the philosophy underpinning OMD is to allow the projection matrix $L$ to be a free optimisation parameter, this opens the door to adapting the OMD algorithm for the purposes of output-regulated coherent structure identification. In the following, we will detail this approach, illustrating how we modify the OMD formulation to simultaneously find the optimal projection matrix $L$, dynamics matrix $M$ and output matrix $C$. In §4.2.1 and §4.2.2, we discuss the algorithm for finding the optimal projection matrix $L$. We will then apply the algorithm on synthetic and experimental datasets in §4.3.

As mentioned above, a reduced-order DMD model concentrates on replicating the most energetic features of the system through a linear model. As a result, one can not rely on the standard implementation of DMD to find features that may be less energetic, but more observable to the measurement $y$. We start, therefore, from OMD, which finds the optimal projection matrix and the optimal reduced-order dynamics matrix simultaneously.

Given the measurable output $y$, we want to find the $r$ ranked subspace of the collected data that balances how well the snapshots are recreated by the ROM, with how well the output $y$ is reconstructed through a linear output matrix $C$. We start from the OMD formulation as described in [174], and explained in §2.4. For completeness, recall that OMD solves the optimisation problem

$$\begin{aligned} \underset{L,M}{\text{minimize}} \quad & J_1 := \|X' - LML^\top X\|_F^2 \\ \text{s.t.} \quad & L^\top L = I, \\ & M \in \mathbb{R}^{r\times r},\ L \in \mathbb{R}^{p\times r} \end{aligned} \tag{4.6}$$

where $r = \text{rank}(M)$ is prescribed by the user as a parameter, $L$ is the optimal orthogonal projection matrix of rank $r$ and $M$ is the low-dimensional dynamics matrix. Solving (4.6) enables the user to simultaneously find the optimal projection matrix $L$ and the dynamics matrix $M$. In practice, this optimisation problem is solved by finding a relationship between a matrix $L$ and its corresponding optimal $M$ matrix, and reducing the optimisation problem to one of finding the optimal $L$. The OMD modes $\Phi$ and eigenvalues $(\mu_i)_{i=1}^r$ are then found diagonalising $M = Q\Gamma Q^{-1}$ and letting

$$\Phi = LQ \quad \text{and} \quad \mu_i = \frac{\log(\lambda_i)}{\Delta t}. \tag{4.7}$$

Our aim is to adjust the OMD formulation with respect to a certain output. We will, therefore, refer to our method as output-regulated optimal mode decomposition, OMDor for the remainder of this thesis. OMDor creates a similar model for the state's progression while we penalise the error in mapping the state to the observable. We augment the cost function in (4.6) with the least squares fit between the true measured values of the outputs contained in $P$ and the estimated outputs $\tilde{P} = CL^\top X$ to get

$$\begin{aligned} \underset{L,M,C}{\text{minimize}} \quad & J := \|X' - LML^\top X\|_F^2 + \alpha\|P - CL^\top X\|_F^2 \\ \text{s.t.} \quad & L^\top L = I, \\ & M \in \mathbb{R}^{r\times r},\ L \in \mathbb{R}^{p\times r},\ C \in \mathbb{R}^{\ell\times r} \end{aligned} \tag{4.8}$$

where $\alpha$ and $r$ are user-defined parameters.

The intention is that for $\alpha > 0$ the choice of spatial projection $L$ will be able to balance dynamic modelling considerations, with the extraction of a subspace $\text{Im}(L)$ and an output mapping $C$ for which the output of a reduced-order state evolving on $\text{Im}(L)$ has a high correlation with the observed measurements in $P$.

### 4.2.1 Transforming the optimisation problem (4.8)

We apply techniques from matrix differentiation used to solve the original OMD optimisation problem in [65] in order to transform the optimisation problem (4.8) into a form that can be solved by standard gradient-based algorithms.

We seek to express the minimisation problem as a function of $L$ alone by first finding, for a fixed $L \in \mathbb{R}^{p\times r}$, matrices $M$ and $C$ for which

$$\frac{\partial J}{\partial M} = 0 \quad \text{and} \quad \frac{\partial J}{\partial C} = 0.$$

For completeness, we note that for a matrix $B = (b_{ij}) \in \mathbb{R}^{n\times m}$ and a scalar-valued function $J : \mathbb{R}^{n\times m} \to R$, the definition of the derivative of $J$ with respect to $B$ is

$$\frac{\partial J}{\partial B} = \left(\frac{\partial J}{\partial b_{ij}}\right)_{i=1,j=1}^{n,m}.$$

That is, $\partial J/\partial B$ is itself an $n \times m$ matrix which contains the element-wise partial derivatives of $J$.

To address (4.8), we must first differentiate $J$ with respect to matrix $C$. To do so, we aim to expand the cost function using the definition $\mathrm{tr}(AA^\top) = \|A\|_F^2$, noting first that

$$\begin{aligned}
\|P - CL^\top X\|_F^2 &= \mathrm{tr}\left((P - CL^\top X)(P - CL^\top X)^\top\right) \\
&= \mathrm{tr}\left(PP^\top\right) - 2\mathrm{tr}\left(P(CL^\top X)^\top\right) + \mathrm{tr}\left((CL^\top X)(CL^\top X)^\top\right) \\
&= \mathrm{tr}\left(PP^\top\right) - 2\mathrm{tr}\left(PX^\top LC^\top\right) + \mathrm{tr}\left(CL^\top XX^\top LC^\top\right) \\
(\text{since } \mathrm{tr}(A^\top) = \mathrm{tr}(A)) &= \mathrm{tr}\left(PP^\top\right) - 2\mathrm{tr}\left(CL^\top XP^\top\right) + \mathrm{tr}\left(CL^\top XX^\top LC^\top\right) \\
&= \mathrm{tr}\left(PP^\top\right) - 2\mathrm{tr}\left(PX^\top LC^\top\right) + \mathrm{tr}\left(C^\top CL^\top XX^\top L\right),
\end{aligned}$$

where in the last line the cyclic property of the trace operation $\mathrm{tr}(ABC) = \mathrm{tr}(CAB)$ is used. Now, using the facts that

$$\frac{\partial}{\partial B}\text{tr}(BX) = X^\top \quad \frac{\partial}{\partial B}\text{tr}(B^\top BX) = B(X + X^\top)$$

for any fixed matrix $X$ of appropriate dimensions, it follows that

$$\frac{\partial J}{\partial C} = \alpha\frac{\partial}{\partial C}\left\|P - CL^\top X\right\|_F^2 = 2(PX^\top L - CL^\top XX^\top L),$$

By setting this derivative to 0, we find that for a given projection matrix $L$, the optimal matrix $C$ can be found using

$$C^\star(L) = PX^\top L(L^\top XX^\top L)^{-1}. \tag{4.9}$$

Similarly, as

$$\frac{\partial J}{\partial M} = \frac{\partial}{\partial M}\|X' - LML^\top X\|_F^2,$$

we can use the same formula as that used in [174] for the optimal matrix $M$ as a function of matrix $L$:

$$M^\star(L) = L^\top X'XL(L^\top XX^\top L)^{-1}. \tag{4.10}$$

Finally, by substituting (4.9) and (4.10) into (4.8), we will reduce the problem from that of finding the optimal matrices $L$, $M$ and $C$, to one of finding the optimal projection matrix $L$. In particular,

$$\begin{aligned}
J(L) &:= J(L, M^*(L), C^*(L)) \\
&= \|X'\|_F^2 - 2\text{tr}(X'X^\top L(M^*)^\top L^\top) + \text{tr}(LM^*L^\top XX^\top L(M^*)^\top L^\top) \\
&\quad + \alpha\|P\|_F^2 - 2\alpha\ \text{tr}(PX^\top LX(C^*)^\top) + \alpha\ \text{tr}(C^*L^\top XX^\top L(C^*)^\top) \\
\text{(by [174])} &= \|X'\|_F^2 - \|L^\top X'Q(L)\|_F^2 \\
&\quad + \alpha\left(\|P\|_F^2 - 2\text{tr}(PX^\top LX(C^*)^\top) + \text{tr}(C^*L^\top XX^\top L(C^*)^\top)\right),
\end{aligned}$$

where $Q(L) := X^\top L(L^\top XX^\top L)^{-1}L^\top X$. The final terms in the above equation involving $C$ can be similarly written in terms of $Q(L)$ by noting that

$$\begin{aligned}
-2\mathrm{tr}(PX^\top LX(C^*)^\top) + \mathrm{tr}(C^*L^\top XX^\top L(C^*)^\top) &= -\mathrm{tr}(PX^\top LX(L^\top XX^\top L)^{-1}L^\top XP^\top) \\
&= -\mathrm{tr}(PQ(L)P^\top) \\
(\text{as } Q(L)Q(L)^\top = Q(L)) &= -\mathrm{tr}(PQ(L)Q(L)^\top P^\top) \\
&= -\|PQ(L)\|_F^2.
\end{aligned}$$

Consequently, we arrive at

$$J(L) = \|X'\|_F^2 + \alpha\|P\|_F^2 - \|L^\top X'Q(L)\|_F^2 - \alpha\|PQ(L)\|_F^2.$$

Since $P$ and $X'$ are both independent of $L$, the optimisation problem (4.8) is equivalent to

$$\begin{aligned}
\underset{L}{\text{maximize}} \quad & \|L^\top X'Q(L)\|_F^2 + \alpha\|PQ(L)\|_F^2 \\
\text{s.t.} \quad & L^\top L = I,\ L \in \mathbb{R}^{p\times r}.
\end{aligned} \tag{4.11}$$

### 4.2.2 Solving the optimisation problem (4.11)

Below, in Algorithm 2, we propose a gradient-based iterative algorithm to solve the maximisation problem (4.11). To do this, it is necessary to compute the gradient of

$$g(L) := \|L^\top X'Q(L)\|_F^2 + \alpha\|PQ(L)\|_F^2.$$

To differentiate $g(L)$, we first use the result from [174] that

$$\begin{aligned}
\frac{d}{dL}\|L^\top X'Q(L)\|_F^2 = &-2\mathcal{B}L(L^\top\mathcal{B}L)^{-1}(L^\top\mathcal{A}^\top L)(L^\top\mathcal{A}L)(L^\top\mathcal{B}L)^{-1} \\
&+ 2[\mathcal{A}L(L^\top\mathcal{B}L)^{-1}(L^\top\mathcal{A}^\top L) + \mathcal{A}^\top L(L^\top\mathcal{A}L)(L^\top\mathcal{B}L)^{-1}],
\end{aligned} \tag{4.12}$$

where $\mathcal{B} = XX^\top$ and $\mathcal{A} = X'X^\top$.

To differentiate the component of the cost which includes the output data $P$ we can apply similar methods. First, we expand the norm

$$\begin{aligned}\|PQ(L)\|_F^2 &= \mathrm{tr}(PQ(L)P^\top)\\ &= \mathrm{tr}(PX^\top L(L^\top \mathcal{B} L)^{-1}L^\top X P^\top)\\ &= \mathrm{tr}(\Gamma L(L^\top \mathcal{B} L)^{-1}L^\top \Gamma^\top)\\ &= \mathrm{tr}(L(L^\top \mathcal{B} L)^{-1}L^\top \Gamma^\top \Gamma)\end{aligned}$$

where $\Gamma := PX^\top$. Differentiating the above quartic expression in $L$ requires the use of the chain rule which, for matrix-valued differentiation, gives

$$\frac{d}{dL}\|PQ(L)\|_F^2 = \frac{d}{dL}\mathrm{tr}(LX_1) + \frac{d}{dL}\mathrm{tr}(X_2(L^\top \mathcal{B} L)^{-1}X_3) + \frac{d}{dL}\mathrm{tr}(X_4 L^\top X_5) \tag{4.13}$$

where the matrices

$$X_1 = (L^\top \mathcal{B} L)^{-1}L^\top \Gamma^\top \Gamma, \quad X_2 = L, \quad X_3 = L^\top \Gamma^\top \Gamma, \quad X_4 = L(L^\top \mathcal{B} L)^{-1}, \quad X_5 = \Gamma^\top \Gamma$$

are all viewed as fixed matrices for the purposes of differentiation in (4.13). The simpler calculations are the first and last expressions in (4.13) which give

$$\frac{d}{dL}\mathrm{tr}(LX_1) = X_1^\top = \Gamma^\top \Gamma L(L^\top \mathcal{B} L)^{-1}L^\top, \tag{4.14}$$

and

$$\frac{d}{dL}\mathrm{tr}(X_4 L^\top X_5) = \frac{d}{dL}\mathrm{tr}(LX_4^\top X_5^\top) = X_5 X_4 = \Gamma^\top \Gamma L(L^\top \mathcal{B} L)^{-1}L^\top. \tag{4.15}$$

For the middle term, the identity

$$\frac{d}{dX}\mathrm{tr}((X^\top CX)^{-1}A) = -CX(X^\top CX)^{-1}(A + A^\top)(X^\top CX)^{-1}$$

can be used to give

$$\begin{aligned}\frac{d}{dL}\mathrm{tr}(X_2(L\mathcal{B}^\top L)^{-1}X_3) &= \frac{d}{dL}\mathrm{tr}((L\mathcal{B}^\top L)^{-1}X_3X_2)\\ &= -\mathcal{B}L(L\mathcal{B}^\top L)^{-1}(X_3X_2 + X_2^\top X_3^\top)(L\mathcal{B}^\top L)^{-1}\\ &= -2\mathcal{B}L(L\mathcal{B}^\top L)^{-1}(L^\top\Gamma^\top\Gamma L)(L\mathcal{B}^\top L)^{-1}. \end{aligned} \tag{4.16}$$

Combining the calculations from (4.12), (4.14), (4.15) and (4.16) gives the analytical expression for the derivative of the optimal cost $g(L)$,

$$\begin{aligned}\frac{dg}{dL} = &-2\mathcal{B}L(L^\top\mathcal{B}L)^{-1}(L^\top\mathcal{A}^\top L)(L^\top\mathcal{A}L)(L^\top\mathcal{B}L)^{-1}\\ &+ 2[\mathcal{A}L(L^\top\mathcal{B}L)^{-1}(L^\top\mathcal{A}^\top L) + \mathcal{A}^\top L(L^\top\mathcal{A}L)(L^\top\mathcal{B}L)^{-1}]\\ &+ 2\alpha\left[\Gamma^\top\Gamma L(L^\top\mathcal{B}L)^{-1} - \mathcal{B}L(L^\top\mathcal{B}L)^{-1}(L^\top\Gamma^\top\Gamma L)(L^\top\mathcal{B}L)^{-1}\right], \end{aligned} \tag{4.17}$$

where, to recall from the above, $\Gamma = PX^\top$, $\mathcal{B} = XX^\top$ and $\mathcal{A} = X'X^\top$.

With this expression in hand, it is possible to adapt the original OMD algorithm from [174] (which is just the case when $\alpha = 0$) to apply to this output-regulated ($\alpha > 0$) form of OMD as well. Algorithm 2, which we refer to as *Output-Regulated OMD* (OMDor), presents this generalisation of the OMD algorithm. For the most part, the algorithm remains unchanged, apart from the alternative expression for $dg/gL$ derived above. However, there are two other areas where the algorithm differs from that used in OMD.

First, in OMD, by default, the matrix of POD modes of the snapshot matrix $X$ is set as the initialisation for $L_0$. However, the nature of the problem OMDor allows for more educated initial guesses. We introduce in §4.2.3 several strategies for choosing an appropriate initialisation $L_0$. Second, in line 8 of Algorithm 2, the function $g(L_{k-1}(\theta))$ is non-convex and expensive to evaluate. In §4.2.5, we propose a Bayesian optimisation routine to ensure global minimality while keeping the function evaluations to a minimum.

**Algorithm 2** An iterative algorithm for finding the optimal matrix $L$

Let $\alpha = 0$, fix an update length $\delta\alpha$.
**while** $\alpha \leq \alpha_{max}$ **do**
    Initialise $L_0 \in \mathbb{R}^{p\times r}$ ▷ See §4.2.3.
    Compute initial gradient $G_0 = (I - LL^\top)\frac{dg}{dL}(L_0)$
    Set search direction $H_0 = G_0$
    Let $k = 1$
    **while** $g(L_k) - g(L_{k-1}) >$ tolerance **do**
        compute $\theta_{min} \in [0, 1]$ for which $-g(L_{k-1}(\theta))$ is minimal, over the geodesic curve

$$L_{k-1}(\theta) = L_{k-1}V\cos(\Sigma\theta)V^\top + U\sin(\Sigma\theta)V^\top$$

    where $H_{k-1} = U\Sigma V^\top$
        update the subspace basis $L_k = L_{k-1}(\theta_{min})$
        update the gradient $G_k = (I - L_kL_k^\top)\frac{dg}{dL}(L_k)$
        update the search direction $H_k = G_k + \Delta_k$
        $k = k + 1$
    **end while**
     **return** $L(\alpha) = L_k$
    $\alpha = \alpha + \delta\alpha$
**end while**

### 4.2.3 Choosing the initialisation $L_0$.

In OMD, by default, the initial guess for the projection matrix, $L_0$, is the truncated POD mode matrix calculated using the state snapshots $X$. In OMDor, since we will have to apply the method at various values of $\alpha$, we may have better strategies at our disposal. We discuss three such strategies and their respective advantages and disadvantages.

**1: Using $L_0 = U_t$:** The first strategy is to set $L_0$ to a truncated set of POD modes, $U_t$, for all values of $\alpha$. The major disadvantage of this method is that, considering the non-convex nature of the objective function, the algorithm is not guaranteed to increase its emphasis on fitting the measurement $(y_j)_{j=1}^N$ as $\alpha$ increases. By substituting $L_0 = U_t$ in (4.17), we find that

$$\begin{aligned}\frac{dg(U_t)}{dL} =& 2\left[X'V_t\Sigma_t^{-1}U_t^\top XX'^\top U_t + (I - U_tU_t^\top)XX'^\top U_tU_t^\top X'V_t\Sigma_t^{-1}\right] \\ &+ 2\alpha(I - U_tU_t^\top)XP^\top PV_t\Sigma_t^{-1}\end{aligned} \tag{4.18}$$

where $V_t$ and $\Sigma_t$ are the truncated matrices in the decomposition $U_tU_t^\top X = U_t\Sigma_tV_t^\top$. The second term in (4.18) is solely responsible for adjusting the direction of the search to account for the measurements $P$ and the weight parameter $\alpha$.

Using the orthogonalisation

$$P^\top P = V_p\Sigma^2V_p^\top$$

where $V_p$ are the temporal POD vectors of matrix $P$, one can see that if $V_p^\top V_t = 0$, i.e. if the measurements are temporally orthogonal to the retained POD time series of the collected snapshots $X$, the initial search direction will be independent of the measured output and the value of $\alpha$.

The matrix $V_t$ is associated with the $r$ most energetic POD modes. Therefore, this can limit the algorithm's ability to enhance the mapping to the measurable output, if the output only correlates with low-energy events that are truncated out of matrix $V_t$. Although in later iterations this direction of search can change, if the relative improvement of the overall cost function is lower than the prescribed cut-off, the algorithm will terminate and will output a result similar to that of OMD, even if $\alpha > 0$. This is undesirable.

The advantage of initialising the algorithm with $L_0 = U_t$ and disregarding the results of the optimisation at previous values of $\alpha$, however, is that it can allow the algorithm to escape local minima, as the search at each iteration is independent of the previous results. This independence also allows the algorithm to be parallelised easily (in the sense

of running the algorithm on multiple computational cores), making it the fastest of the presented initialisation methods.

**2: Using $L_0(\alpha) = L(\alpha - \delta\alpha)$** By definition, Algorithm 2 systematically increases the value of $\alpha$ via the update $\alpha \mapsto \alpha + \delta\alpha$. This successively prioritises finding reduced order states which can correlate well with the measured outputs. Therefore, a second approach is to initialise the matrix $L_0$ in step 3 of the algorithm, at each iteration, with the solution to the optimisation at the previous value of $\alpha$. That is, $L_0(\alpha) = L(\alpha - \delta\alpha)$. This approach has the advantage of guaranteeing that $\|P - \tilde{P}\|_F$ will either remain constant or decrease as $\alpha$ increases. As a result, this approach is suitable for finding the trends in the optimisation results of the algorithm as we vary different parameters.

However, starting the search from the optimal solution at $\alpha - \delta\alpha$ has the adverse effect of making the algorithm less likely to escape a local optimum.

**3: Combining both approaches** A third method is to first run the Algorithm 2 by initialising $L_0$ to be the POD modes at each value of $\alpha$ to find the value of $\alpha_{min}$ at which the algorithm finds the best mapping between the outputs $(y_i)_{i\geq 0}$ and $x$ and hence the minimum residual $\|P - \tilde{P}\|_F$. We can then rerun the algorithm for values of $\alpha > \alpha_{min}$, but this time set $L_0(\alpha) = L(\alpha - \delta\alpha)$ to guarantee that the algorithm can only improve on its previous performance. This method allows us to search for $L$ over a wider region while enabling us to detect a trend in what features of original snapshots the model chooses as $\alpha$ varies.

### 4.2.4 Selecting an appropriate value of $\alpha$.

A key feature of Algorithm 2 is that the weight parameter $\alpha$ is gradually increased from $\alpha = 0$ (i.e. the original OMD algorithm) to successively larger values of $\alpha$ which promotes mode shapes correlated with a given observable. While a nominal stopping criteria $\alpha_{\max}$ is included in Algorithm 2, in practice, one must choose a particular value of $\alpha$ at which

to extract (and then use) the obtained matrices $L, M$ and $C$. Such a choice is naturally a balancing act, since low values of $\alpha$ prioritise dynamic fitting to snapshot data (i.e. lower costs $\|X' - LML^\top X\|$), while a large $\alpha$ prioritises modes which correlate well to the chosen observable (i.e. low costs $\|P - CL^\top X\|$). Hence, prioritising one aspect of the fit necessarily reduces the accuracy of the other, and it is therefore not possible to recommend a single "correct" value of $\alpha$ for a given application. However, as will be seen in the subsequent examples, it is typically observed that abrupt changes to the dynamic and observation cost components ($\|X' - LML^\top X\|$ and $\|P - CL^\top X\|$, respectively) occur as $\alpha$ passes certain threshold values. Further, after each threshold is reached the value of each cost component then plateaus. This behaviour may be viewed as exploring the Pareto front associated with the two competing components of the cost function. A simple rule for selecting an appropriate value of $\alpha$ is, therefore, to consider the output matrices immediately after each abrupt change in the cost components, and a user may subsequently select the outputs most closely associated with their desired use (i.e., either prioritising dynamic or output modelling, or balancing the two).

### 4.2.5 Finding $\theta_{min}$ using Bayesian optimisation

In line 8 of Algorithm 2, we minimise the objective function (4.11) over the geodesic curve. However, the equation $-g(L_{k-1}(\theta))$ is computationally expensive to evaluate, making gradient-based optimisation algorithms unsuitable for this sub-component of the OMDor algorithm. The original OMD algorithm [174] relies on a line-search algorithm that finds the first local minimum in the cost function using a combination of a forward-backwards search and a quadratic fit in a trust region close to $\theta = 0$.

However, depending on the state and output signals, the addition of the output regularisation term to the optimisation problem in (4.8) is typically observed to introduce additional local maxima to the cost function, which increases the non-convexity of the function $-g(L_{k-1}(\theta))$. As a result, it is vital to include a more robust line-search algorithm to ensure there is a better estimate of the global minimiser of the function $-g(L_{k-1}(\theta))$.

Bayesian optimisation is a leading example of a global optimisation algorithm which relies on adaptive sampling to efficiently solve non-convex optimisation problems. For example, in machine learning, Bayesian optimisation is often used to find optimal hyper-parameters of the model [158, 89], including finding optimal neural network architectures [88]. In the context of flow control, a recent study used Bayesian optimisation for finding the optimal wall normal blowing for drag reduction in turbulent boundary layers [111].

For our application, the objective function is parameterised as a scalar-valued function $g$ of a scalar parameter $\theta$. As a result, although the evaluation of the objective function remains expensive, the dimensions and the complexity of the minimisation in line 8 of Algorithm 2 remain low. Therefore a generic implementation of Bayesian optimisation will suffice for our application. We use MATLAB's Bayesian optimisation package which uses Gaussian kernels for approximating the objective function, to find a low-cost estimate of that global minimum.

The use of Bayesian optimisation to find $\theta_{min}$ will make the OMDor algorithm more robust against the introduction of new scales to the objective function. However, careful consideration of the optimisation parameters is needed to ensure the results converge in time. For example, the sampling must be limited to a sensible range of $\theta$ values and the initial sampled points most be arranged in accordance with the expected behaviour of function $g$.

To illustrate how OMDor allows the practitioner to find the subspace of data that is most relevant to output $y$, in §4.3 we apply OMDor on a synthetic dataset where the subspace of the data that $y$ best correlates to is known. We would like to see if the OMDor model can capture this subspace of data even when more energetic features are present in the dataset.

## 4.3 Results

As discussed in §2, each coherent physical feature in the flow is represented by a mode shape and a corresponding time series associated with the feature. There are features whose influence covers the entire spatial domain of a flow are global modes, e.g. the vortex-shedding mode in the flow past a circular cylinder at $Re > 46$ [157]. However, certain temporal features may only be locally observable. For example, in flows past axisymmetric bluff-bodies, the portion of the flow-field immediately past the bluff-body, i.e. the recirculation region, is dominated by low-frequency events, while higher frequency events associated with vortex shedding dominate the flow-field further downstream[20, 34, 142]. Assuming that a feature is spatially localised (or only coherent locally) it is not unreasonable to assume that the spatial feature is only observable using measurements taken from the corresponding subset of the flow domain. Moreover, in data-driven modelling of such flows, the varying energy content of the features may lead to a varying degree of detectability in the collected data.

An advantage of using OMDor for modelling a flow-feature associated with the output $(y_i)_{i\geq 1}$, is that the model should automatically detect the region of the flow-field where the feature can be observed, even if the flow-feature is not energetically dominant in the collected data. To illustrate the potential of OMDor in modelling such localised flow-features, we apply the algorithm on a synthetic dataset where there are two sinusoidal flow-features, with differing energy levels, length scales and temporal frequencies, in two separate regions of the flow-field. The goal is for OMDor to be able to correctly identify the spatio-temporal feature associated with the measured output $(y_i)_{i\geq 1}$ even if that feature is low energy in the collected data.

### 4.3.1 Application of OMDor on a synthetic sinusoidal flow

To create the synthetic dataset, we first form a discrete, uniform, spatio-temporal mesh with $0 \leq s \leq 1$ and $0 \leq t \leq 2$, where $\Delta s = 0.001$ is the spatial increment and $\Delta t = 0.01$

is the time-step between each two consecutive snapshots. Suppose now that the value of the state $x$ at the spatial location $s$ and at time $t$ is defined by

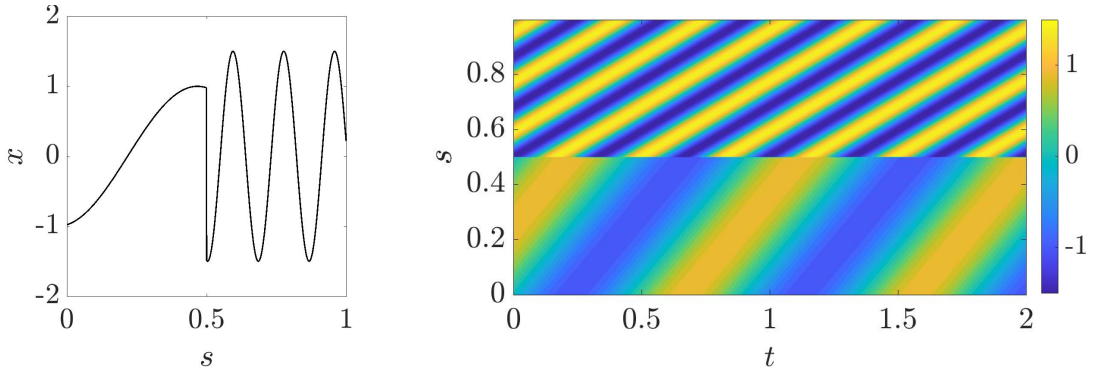
$$x(s,t) = \begin{cases} A_1 \sin\left(\omega_1 \pi s - f_1 t\right), & 0 \leq s \leq 0.5, \\ A_2 \sin\left(\omega_2 \pi s - f_2 t\right), & 0.5 < s \leq 1. \end{cases} \tag{4.19}$$

For the following, initial, discussion of the performance of Algorithm 2, the chosen parameters will be $A_1 = 1$, $A_2 = 1.5$, $\omega_1 = 2$, $\omega_2 = 11$, $f_1 = 7$ and $f_2 = 19$. The spatial and temporal frequencies $\omega_i$ and $f_i$ parameters are chosen to be different prime numbers to avoid non-trivial and unintended spatio-temporal relations to exist between the two distinct sections of data $s \in [0, 0.5]$ and $s \in [0.5, 1]$. A discussion about the effects of varying these parameters will be presented in §4.3.2.

Substituting the above parameters in (4.19) we define a dataset with the governing equations

$$x(s,t) = \begin{cases} \sin(2\pi s - 7t), & 0 \leq s \leq 0.5, \\ 1.5\sin(11\pi s - 19t), & 0.5 < s \leq 1. \end{cases} \tag{4.20}$$

Figure 4.2 (a) shows a snapshot of state $x$ at $t = 2$ and (b) shows the value of $x$ as a function of space and time.




(a) A snapshot $x(s,t)$. (b) $x(s,t)$

Figure 4.2: An overview of the synthetic data ensemble. (a) presents an illustrative snapshot; (b) presents the evolution of the data over the time window $t \in [0, 2]$.

The periodic nature of the dataset implies that there exist two pairs of complex conjugate modes that can describe the entire data set. These modes are spatially orthogonal due to the chosen decomposition of the data (one conjugate pair in each half of the spatial domain). Consequently, applying OMD with $r = 4$, i.e. a set of 4 OMD modes, has the potential (if the algorithm converges to this solution value of $L$) to find these modes and describe the full dataset [174].

It is, therefore, intuitive that lowering the parameter $r$ below 4 will adversely affect the ability of the OMD modes to describe the full dataset. As discussed in §3, the correct choice of parameter $r$ is not always obvious. Furthermore, there may, in a complex data sets, be many spatially-localised features of interest. In this minimal example, we have created the simplest possible situation in which only a subset of the available features may be observed using measurements from a local segment of the spatial domain. The question we want to address with this example is, in a decomposition where only a subset of the flow features can be captured, *does the OMDor formulation lead to capturing the subspace of the flow field that is relevant to the measurable output?*

To that end, we first extract an OMD model of rank $r = 2$. Note that the feature with higher spatial and temporal frequency has a higher amplitude and as a result, we expect OMD at $r = 2$ to prioritise the more energetic flow feature. In line with this expectation, applying OMD with $r = 2$ to the snapshots of the state $x$ in (4.20), we find a pair of eigenvalues $\mu = 0 \pm 18.9i$, where the temporal frequency $f_r = 18.9$ is close to the expected temporal frequency $f_2 = 19$. The amplitude difference between the two segments of the OMD mode $\phi_i$ in Figure 4.3 confirms that the OMD model is biased towards reconstructing the more energetic subspace of the data. While there is an imprint of the lower frequency mode, this is clearly of insignificant amplitude in comparison with the higher frequency feature located in the subdomain $0.5 < s < 1$.

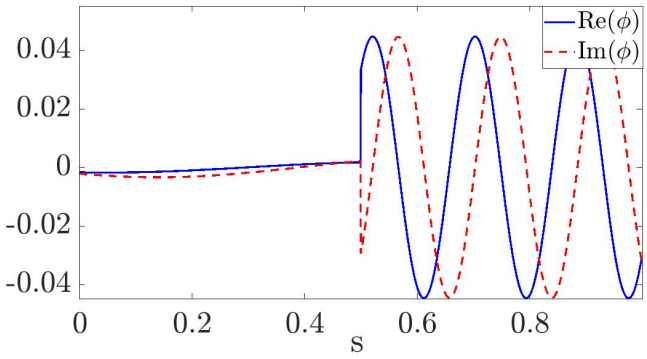



Figure 4.3: The single mode obtained by applying OMD with $r = 2$. Note that this also corresponds to the setting $\alpha = 0$.

Suppose now that we have a measurable system-output that corresponds to the low-energy segment of the data. For example, suppose we are able to measure the state $x$ at 5 equispaced spatial locations in the domain $0 \leq s \leq 0.5$, where temporal oscillations at frequency $f_1 = 7$ are dominant.

A discussion on varying the number of measurements will be presented in §4.3.2. However, we note that for real flows the number of measurements is often limited by the physical restrictions of the experimental setup (e.g. using boundary sensors to avoid influencing the flow unexpectedly, or practical restrictions on the number of sensors which can be reasonably installed).

Using these measurements as the output $y$ of the system, we can now apply OMDor at various values of $\alpha$. To evaluate the performance of our method, we separate the two terms in the cost function $J$ of (4.8) and define the state error

$$E_x = \frac{\|X' - LML^\top X\|_F}{\|X'\|_F} \times 100\% \tag{4.21}$$

to be the percentage error between the predicted and real snapshots of the state, while the output error

$$E_p = \frac{\|P - CL^\top X\|_F}{\|P\|_F} \times 100\% \tag{4.22}$$

is the percentage error between the measured and predicted snapshots of output $y$. Figure 4.4 shows how the values of $E_x$ and $E_p$ evolve as we change the value of $\alpha$. For $\alpha < 160$, the ROM replicates the state with the error of $E_x \approx 55\%$. We expect an initial state error of around 50%, as the parameter $r = 2$ and the ratio of amplitudes mean that a ROM only covering the more energetic section of the data can at best replicate 60% of the state ensemble's energy. Figure 4.4 also shows that for $\alpha < 160$ the output error is almost 100% as the lower-energy feature, which occupies the subset of the domain in which the measurement "sensors" are located, is not described by the ROM. For values of $\alpha$ above 160, there is a marked decrease in $E_p$ as well as a sharp, but smaller, rise in $E_x$. The model detects the low-energy, low-frequency feature which corresponds to the measurement $y$.

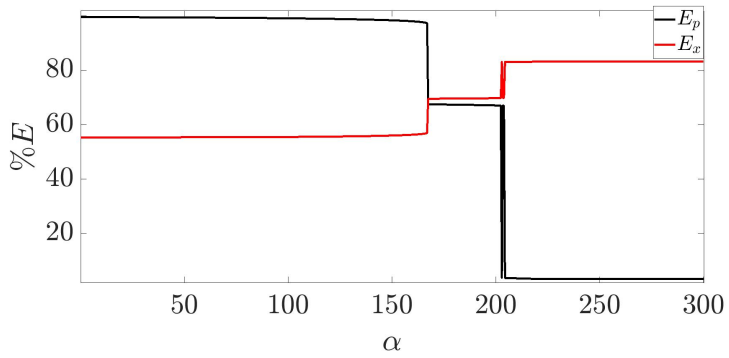



Figure 4.4: The variation of error statistics $E_p$ and $E_x$ with $\alpha > 0$.

Comparing the OMDor mode at $r = 2$ and $\alpha = 250$ in Figure 4.5, with the OMD mode at $r = 2$ and $\alpha = 0$ in Figure 4.3, we can see that at higher $\alpha$ the ROM emphasises the reconstruction of the region of the flow which is better mapped to the output $y$. This can explain the decrease in $E_p$ for $\alpha > 160$. The eigenvalue of the mode in Figure 4.5 is $\mu = 0.0 + 7.0i$, which confirms that at $\alpha = 250$ the ROM prioritises replicating the lower energy features at $f_1 = 7$. Note also that while there is, again, an imprint of the high-energy feature in the subdomain $0.5 < s < 1$, their amplitude is negligible compared to the low-frequency feature. This implies that OMDor modes prioritise capturing the dynamics of the subdomain $0 < s < 0.5$. This observation is confirmed by considering

Figure 4.6 which shows the projected data $LL^\top X$, where $X$ is the matrix of snapshots of the state $x$, for the three cases considered.

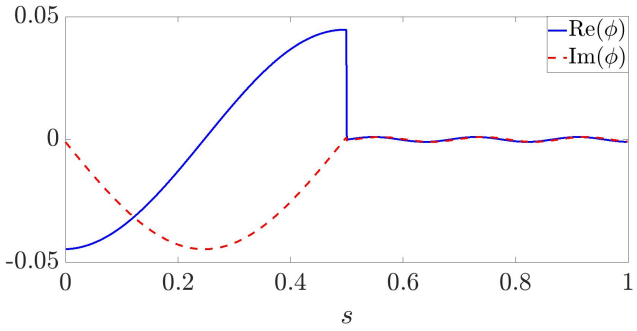
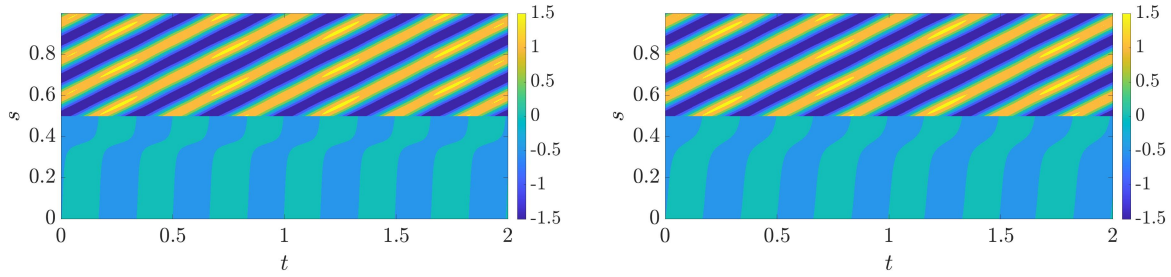



Figure 4.5: The single OMDor mode obtained with parameters $r = 2$ and $\alpha = 250$.

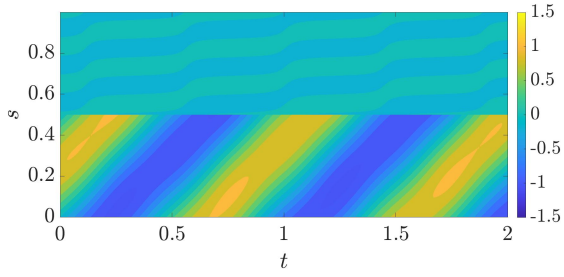
Figure 4.6 compares the subspace of the data that the projected data $LL^\top X$ covers in three cases. The first, (a) case is when $L$ is assumed to be the first 2 POD modes, as in truncated DMD. The second case is when $L$ is found using OMD and the third case is when $L$ was found using OMDor at $\alpha = 250$.




(a) DMD




(b) OMD (OMDor at $\alpha = 0$)




(c) OMDor at $\alpha = 250$

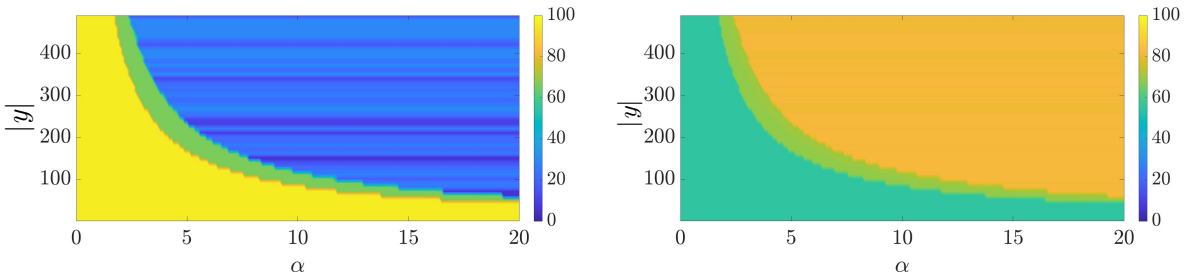
Figure 4.6: Projection of the underlying data ensembles $LL^\top X$: (a) DMD, with $L$ given by the POD modes of the ensemble; (b) OMD, with $L$ obtained with $\alpha = 0$; (c) OMDor, with $L$ obtained with $\alpha = 250$.

The example above demonstrates that OMDor is capable of selecting a feature that is significantly better correlated with the available measurements $y$, even if that feature is less energetic. However, the specific amplitudes and frequencies in (4.20) were arbitrarily chosen. Moreover, for this example we could select a different number of velocity readings as our output signal $y$. In §4.3.2 we illustrate how varying each of these parameters affects the performance of OMDor.

### 4.3.2 A parametric study of OMDor

For this analysis, we set the number of features that OMDor can extract at $r = 2$. This ensures that the model can only choose one of the two periodic features present in the data. We start by analysing the effects of increasing the number of measured outputs. Applying OMDor to the same dataset as in (4.20), we vary the number of virtual probes in the lower half of the spatial domain, i.e. $s \leq 0.5$. Since there is only one temporal feature in the subdomain where the probes are positioned, we expect to see a similar performance for OMDor regardless of the number of probes. However, it is reasonable to expect that the desirable OMDor performance may occur at different values of $\alpha$, depending on the number of probes available. Figure 4.7(a) shows that output error drops to comparable levels for all values of cardinality $|y| > 50$. However, as the number of probes decreases, the value of $\alpha$ at which the drop in $E_p$ happens increases. For example, the results in Figure 4.5 used 5 probes and were obtained at $\alpha = 250$.

In Figure 4.7(b) it can be seen that the state error, $E_x$, increases as the output error $E_p$ decreases. Note, however, that the model is always forced to choose between one of the two existing features and therefore the state error at $\alpha = 0$ is around 55%.



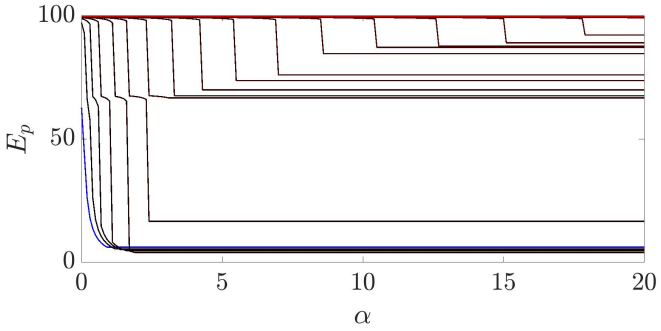

(a) $E_p$ (b) $E_x$

Figure 4.7: The influence of the cardinality of the measured output $|y|$ and of the parameter $\alpha$: (a) on the output error $E_p$; (b) on the state error $E_x$.

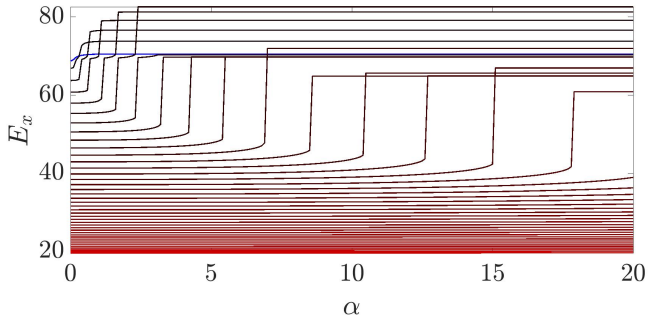
At $|y| = 500$, when the entire lower half of the flow field is considered as the measurement signal $y$, the model switches features at $\alpha \approx 2$. Interestingly, the ratio between the energy of the entire flow field and the energy of the lower half of the flow field is 1.8. This is also consistent with the trend that the number of probes and the value of $\alpha$ at which the drop in $E_p$ happens are inversely related.

To further analyse the effects of changing the energy ratio on the performance of OMDor, we fix the number of virtual probes to the maximum number possible, $|y| = 500$, i.e. $y = x(s,t)$ for $0 \leq s \leq 0.5$. We also fix the amplitude of the lower half of the flow field to $A_1 = 1$, and vary the amplitude of the upper half of the flow field, $A_2$, between 1 and 5. In Figure 4.8, the line corresponding to $A_2 = 1$, i.e. equal energy in the lower-half and upper-half, is drawn in blue, and as $A_2$ increases, the lines are shaded in gradations between black (low $A_2$) and red (high $A_2$). Figure 4.8 (a) shows that as the amplitude $A_2$ increases, the value of $\alpha$ at which the model switches to the low-frequency feature increases. Figure 4.8 (a) shows that as $A_2$ increases, at first the final output error decreases, with the $A_2 = 1.4$ line reaching a steady minimum at 3.8%, compared to 6% at $A_2 = 1$. However, as $A_2$ further increases, the minimum output error that OMDor can achieve increases. Figure 4.8 (b) shows that as the amplitude of the upper half increases, the initial value of $E_x$ at $\alpha = 0$ decreases, as the model can replicate more of the underlying data by choosing the more energetic mode. At amplitudes $A_2 > 1.5$ both $E_x$ and $E_p$ remain high, once $\alpha$ is large enough for the model to switch features. However, for moderate amplitude

ratios, OMDor seems to be capable of extracting features more relevant to the measured output $y$.




(a) $E_p$



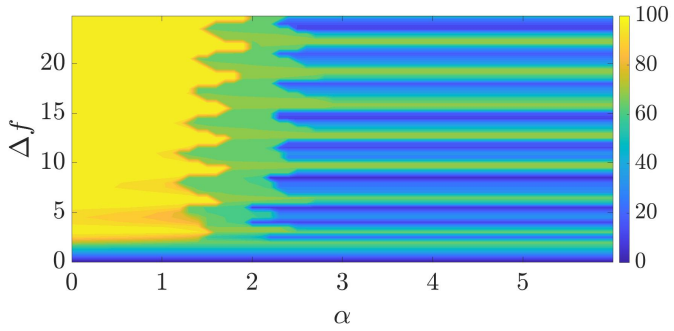

(b) $E_x$

Figure 4.8: The effect of the amplitude $A_2/A_1$ ratio (a) the output error $E_p$; and (b) on the state error $E_x$. The data amplitude ratio varies from $A_2/A_1 = 1$ (black) to $A_2/A_1 = 5$ (red).

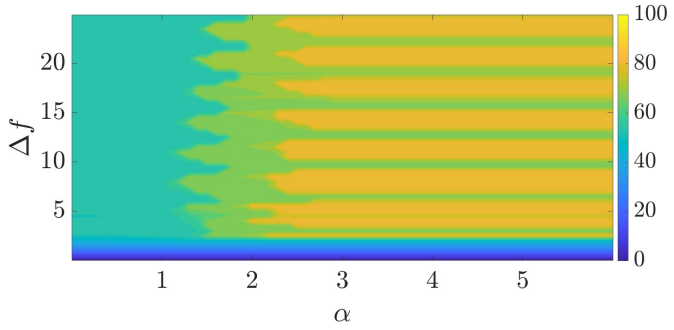
Finally, we will analyse the effects that the frequency difference between the two features will have on the performance of OMDor. We fix the temporal frequency of the lower half, $f_1 = 7$, and increase the frequency difference between the two sections $\Delta f = f_2 - f_1$. As expected, Figure 4.9(a) shows that for low $\Delta f$, OMDor can detect both features at all values of $\alpha$, as they are temporally similar. However, as the frequency difference increases, the initial value of $E_p$ at $\alpha = 0$ increases, with the model preferring to fit to the more energetic feature. Increasing, $\alpha$ beyond 2 improves the mapping between the state and output.

However, an interesting pattern emerges where the steady minimum error that OMDor can achieve, changes in a periodic fashion with increasing $\Delta f$. For the current analysis, the second initialisation technique in §4.2.3 was used and we conjecture that this has led

to the optimisation problem returning local minimums. For example, while Figure 4.9(a) shows that at $\Delta f = 9.75$ the best performance of the algorithm is at $E_p = 75\%$, using the third initialisation technique in §4.2.3 at $\Delta f = 9.75$ we find that the model finds the desired feature, with the output error dropping to 4% for all values of $\alpha > 2.3$. However, the third initialisation technique is more computationally expensive and therefore was not suitable for the full parametric analysis.




(a) $E_p$




(b) $E_x$

Figure 4.9: The influence of frequency difference $\Delta f$ and of the parameter $\alpha$: (a) on the output error $E_p$; (b) on the state error $E_x$

Sinusoidal flows such as the one analysed here are good examples of a noise-free dataset from a periodic system. However, in most real-life non-linear systems which exhibit quasiperiodic behaviour, energy is distributed throughout a range of time-scales despite the existence of a few dominant frequencies. Moreover, the dominant oscillations in such datasets often show temporal intermittence and phase shifts. An example of an oscillatory mode's time series in a turbulent flow, is presented in Figure 4.10, which corresponds to

the real part of the representative vortex shedding mode found in §3, of the flow past an axisymmetric bluff body.

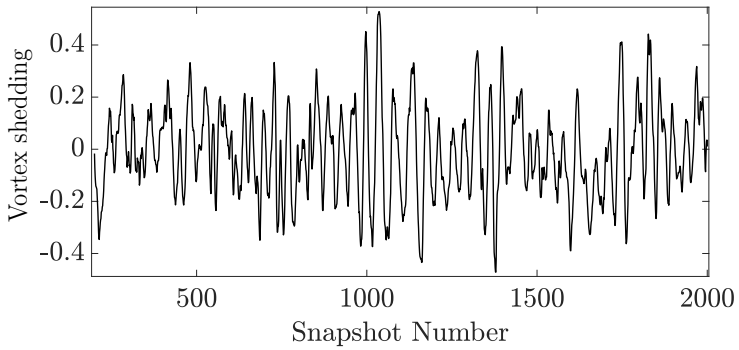



Figure 4.10: Indicative time series of a vortex shedding mode for flow past an axisymmetric bluff body.

We now therefore apply OMDor on a synthetic dataset where such phase drifts and frequency variations are present.

### 4.3.3 Application on a synthetic dataset with phase-drift and frequency-noise.

To create a more realistic test case, we introduce a random phase-drift and temporal-frequency variation to each segment of the data. To generate the random phase drift, we set the initial values of phase drift in both segments to $\theta_1(0) = 0$ and $\theta_2(0) = 0$. The phase drifts, $\theta_1$ and $\theta_2$, for each segment of the data are randomly generated by using a discrete random walk

$$\begin{aligned} \theta_1(i) &= \theta_1(i-1) + 0.03a \\ \theta_2(i) &= \theta_2(i-1) + 0.03b, \end{aligned} \tag{4.23}$$

where $a \sim \mathcal{N}(0,1)$, and $b \sim \mathcal{N}(0,1)$ are independent normally distributed random variables.

Similarly, the frequency disturbances, $\xi_1$ and $\xi_2$, are randomly generated using

$$\begin{aligned} \xi_1(i) &= 0.05\epsilon \\ \xi_2(i) &= 0.05\gamma, \end{aligned} \tag{4.24}$$

where $\epsilon \sim \mathcal{N}(0,1)$ and $\gamma \sim \mathcal{N}(0,1)$. The resultant state is defined by

$$x_1(s,t) = \begin{cases} \sin(2\pi s - ((7+\xi_1(t))t + \theta_1(t))) & 0 \leq s \leq 0.5 \\ 1.5\sin(11\pi s - ((19+\xi_2(t))t + \theta_2(t))) & 0.5 < s \leq 1. \end{cases} \tag{4.25}$$

Figure 4.11 shows an instance of $x_1(s,t)$ in the presence of these uncertainties. For these snapshots, the noise-to-signal ratio is

$$\frac{\|x(s,t) - x_1(s,t)\|_F^2}{\|x(s,t)\|_F^2} \times 100 = 153\%,$$

where $x(s,t)$ is found using (4.20).

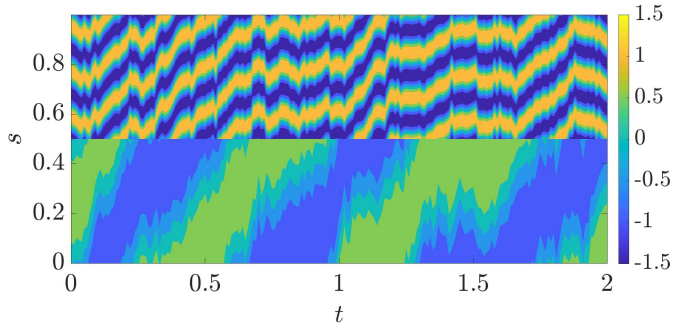



Figure 4.11: Indicative time series of $x(s,t)$ in the presence of frequency noise and phase shift

Applying OMDor at various values of $\alpha$ to the snapshots shown in Figure 4.11, while setting $y$ to be the signal, $x_1$, at the same 5 equispaced locations in $0 \leq s \leq 0.5$, which we used in §4.3.1, we find the values of $E_p$ and $E_x$ in Figure 4.12 (a). Despite the value of the noise-to-signal ratio being higher than unity, Figure 4.12 (a) illustrates that OMDor improves the mapping between the output and the state. In particular, if a standard appliation of OMD is used (i.e. $\alpha = 0$) the best linear fit between OMD weight coefficients and measurement signals has an error of almost 90%. However for $\alpha \geq 200$, OMDor is

able to find modes whose temporal coefficients are significantly more correlated with the available measurements, with the best linear mapping between OMD temporal coefficients and measurements, in this case, having an error of less than 20%.

The randomised nature of the disturbance signals implies that snapshots will vary every time we generate them. To analyse the efficacy of OMDor, we generate 100 series of randomised snapshots using (4.25) and apply OMDor at $0 \leq \alpha \leq 300$ using the $x_1$ readings at the same 5 equispaced locations for the output signal $y$. Figure 4.12 (b) shows the mean value of $\langle E_p \rangle$ and $\langle E_x \rangle$ at each value of $\alpha$. Highlighted, are the 90% confidence interval of each error plot. The decreasing trend seen in $\langle E_p \rangle$ in Figure 4.12 (b), confirms that applying OMDor to this noisy dataset results in better mapping between the state and the output signals. The wide confidence interval at $\alpha > 110$ is due to the wide range of phase and frequency noises that (4.23) and (4.24) can produce. Moreover, the probe positions remained unchanged for all 100 runs, even though it was observed that in the presence of uncertainties, there were probe arrangements that performed better than the considered equidistant arrangement.

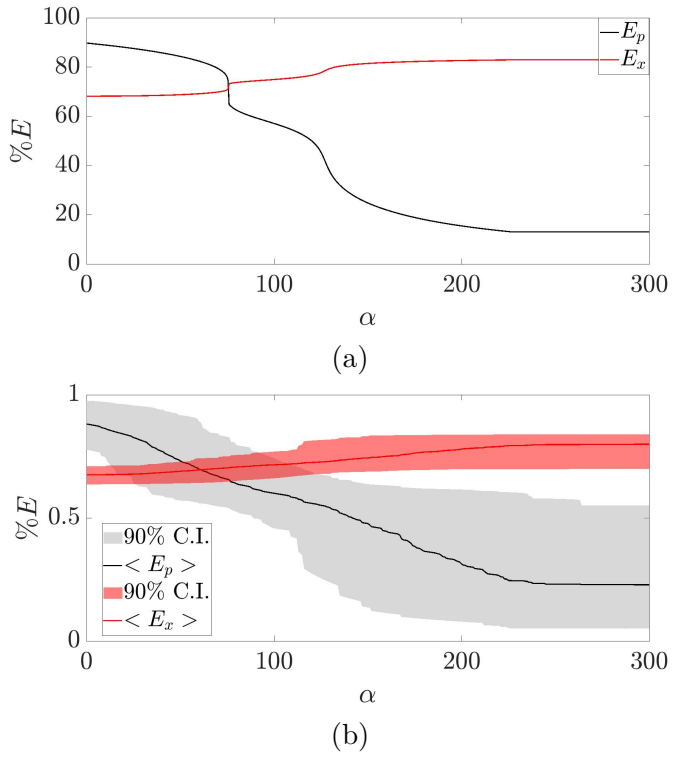



Figure 4.12: (a) Variation of error statistics $E_p$ and $E_x$ with $\alpha$ for a data ensemble with phase drift and frequency noise. (b) average mean values $\langle E_x \rangle$ and $\langle E_p \rangle$ and 90% confidence intervals, averaged over 100 runs.

It is worth noting that both the phase-drift signals and the frequency noise signals contain arbitrary constants which determine the energetic content of those signals and will affect the noise-to-signal ratio.

We have now shown that OMDor is able to improve the detection of low-energy local features in a synthetic dataset, with or without noise. However, for both cases, the limited number of features in the underlying data meant that the optimisation problem had a clear choice of switching between the modes it chooses to model. To consider a system with a large number of degrees-of-freedom, we will now concentrate on the dynamically complex flow past an axisymmetric bluff body considered previously in §3.

### 4.3.4 Application of OMDor to flow past an axisymmetric bluff body

We will apply OMDor on snapshots of the velocity field from the turbulent wake of an axisymmetric bluff body at $Re = 1.88 \times 10^5$, which was previously studied in §3.3.4. Each snapshot contains the streamwise and the spanwise components of the velocity at a mesh of $(186 \times 163)$ spatial location in the flow field. In addition to the snapshots of the velocity field, we have access to the corresponding snapshots of 64 pressure tappings at the base of the bluff-body. The captured flow field contains a recirculation region where low-frequency features are dominant, namely the bubble pumping feature at $St \approx 0.05$. Here, $St$ refers to the Strouhal number found using the free-stream velocity and the diameter of the bluff body's base. The analysis in §3 shows that at a high value of $r$, OMD extracts a range of similar modes in this region with $St = \mathcal{O}(10^{-2})$. Outside the recirculation region, there is a dominant vortex-shedding feature. The flow also exhibits sudden and randomly (in time) distributed, out-of-plane azimuthal rotations. It has been shown that the azimuthal reorientation of the flow happens at a very low frequency [142], $St \approx 0.002$, and we have isolated 2732 sequential snapshots of the velocity field over which the orientation remains nearly unchanged, with the centre of pressure showing a variance of $\pi/5$ rad.

Due to the highly unsteady nature of the flow, which possesses energetic structures across a wide range of length and time scales, and the noisy nature of the experimental data, the features obtained through OMD or DMD are highly dependent on the prescribed dimension of the reduced-order-model, $r$. OMD extracts modes in pairs of complex conjugate oscillatory modes or real-valued non-oscillatory modes, e.g. a mode associated with the mean flow. Since there are two dominant oscillatory features present in the velocity dataset, we know that to capture the dominant features of the flow, we will need to set the parameter $r \geq 5$. Applying OMD at $r = 5$ can therefore correspond to searching for the ROM of the lowest possible dimension that is able to extract the dynamical features with an established physical interpretation in the literature. However, applying OMD at

$r = 5$ we find that the extracted modes' eigenvalues do not match the expected dominant features. Table 4.1 shows that the original OMD algorithm, i.e. OMDor with $\alpha = 0$, extracts a feature at $St \approx \pm 0.006$, which is not close to one of the expected frequencies.

| $\alpha$ | 0 | 200 |
|---|---|---|
| $St_1$ | 0 | 0 |
| $St_2$ | $\pm 0.20$ | $\pm 0.19$ |
| $St_3$ | $\pm 0.006$ | $\pm 0.03$ |

Table 4.1: Strouhal numbers of the identified features at $r = 5$, with different values of $\alpha$

Now, we apply OMDor to the same dataset at the same value of $r = 5$, but this time the balancing parameter $\alpha = 200$. For measured output $y$, we use a set of pressure readings recorded by 64 pressure tappings at the base of the bluff body, arranged in a radially and circumferentially equidistant pattern, as explained in [135]. As we have access to many more snapshots of the pressure readings, we first subtract the long-term average pressure field from the pressure snapshots. The frequencies stated in the second column of Table 4.1 show that OMDor now detects the correct low-frequency feature, which is known to be observable from base pressure measurements [142]. Note also that detecting the bubble-pumping mode does not come at the cost of corrupting the frequency content of the vortex-shedding mode that is identified at $\alpha = 0$ as well.

The above comparison shows the merit of using the OMDor algorithm on experimental flows. To give a more thorough account of the effect of varying the balance parameter $\alpha$, we applied the OMDor algorithm with $0 \leq \alpha \leq 1000$, using the same pressure fluctuations as the measured output $y$. Furthermore, the algorithm was applied at three different values of the decomposition dimension $r$, to assess the effects of the decomposition dimension on OMDor's performance. The resulting loss functions are depicted in Figure 4.13. At $r = 5$, this figure shows that for $\alpha > 80$ the measurement error, $E_p$, is reduced from the steady value of 75% to values fluctuating around 61%, with no significant increase in $E_x$. This implies that at a very limited dimension, OMDor modes are able to better model

the measured output $y$, without a discernible decrease in their ability to represent the underlying flow features.

Changing the decomposition dimension $r$ to 50, we see that when $\alpha > 210$ there is still an improvement to measurement error $E_p$, which decreases from approximately 50% to 45%. Similar to the previous case, there is no significant change to the state error $E_x$. It is also interesting to note that $r = 50$ the onset of the decrease in $E_p$ occurs at a higher value of $\alpha$ than the corresponding decrease observed when $r = 5$.

Repeating the same output regularisation at $r = 500$, one can see from Figure 4.13 that the initial values of $E_x$ and $E_p$ for $\alpha = 0$ are much lower than the previous two cases. Here, in the range $0 \leq \alpha \leq 1000$, increasing the balance parameter $\alpha$ does not lead to an improvement in the output mapping. This behaviour can be attributed to the fact that in the case that the underlying mode ensemble is large, it is possible to obtain modes which enable an optimal linear relation between mode weights and observables without degrading the ability of the obtained model to capture the flow dynamics. This observation suggests that OMDor is primarily beneficial if a user is seeking a flow model of low or moderate dimension.

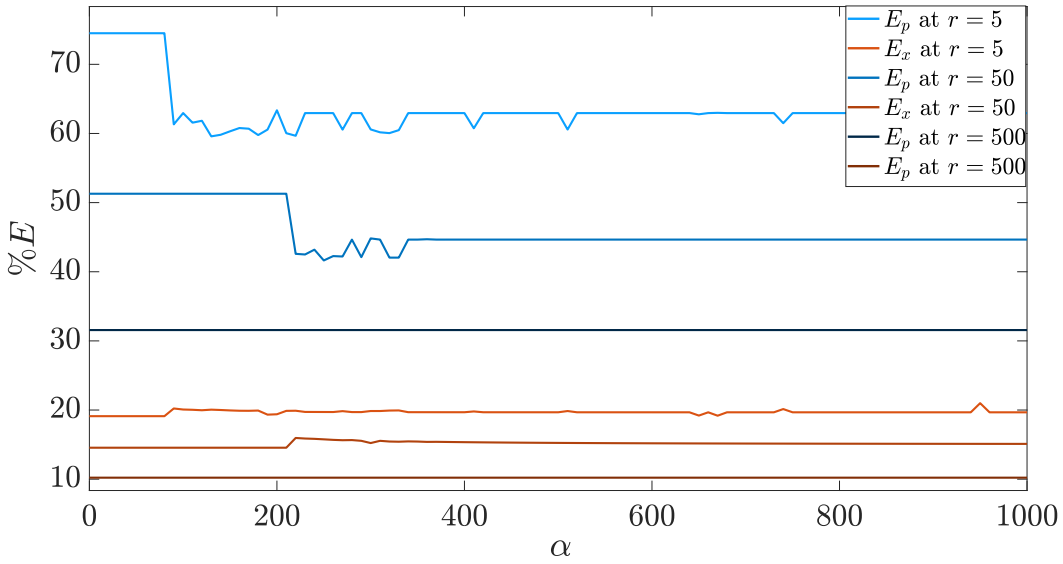



Figure 4.13: Error statistics $E_p$ and $E_x$ for OMDor at parameter values $r = 5, 50$ and $r = 500$ and for $0 \leq \alpha \leq 1000$.

Finally, we comment on the generally high value of $E_p$ in Figure 4.13. Although, predictably, the linear mapping between the base pressure and velocity field remains imperfect, OMDor clearly improves this mapping as well as the extracted dynamical features. Such an improvement may be beneficial if this output mapping is used as part of a dynamic estimation algorithm, for example. We note here that in theory OMDor can be extended to cases where the assumed mapping function $\mathcal{C}$ is non-linear. In this case, based on the relationship between velocity and pressure, a quadratic mapping function may be of particular interest.

## OMDor, with velocity measurements

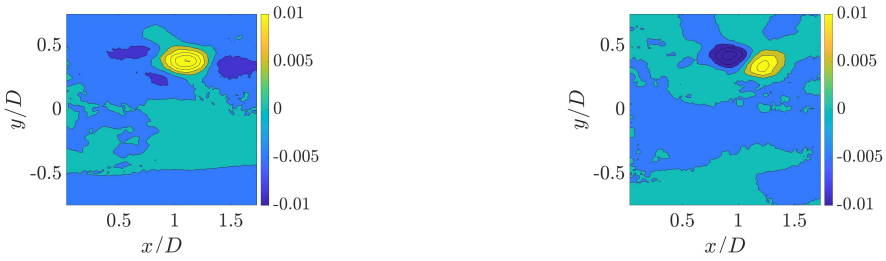
We now apply OMDor to this dataset with another measurable output. From the analysis in §3 it is known that there is a vortex-shedding feature that exists downstream of the recirculation bubble, at $x/D > 1.25$. There is also a lower energy structure that corresponds with the first harmonic of the shedding mode. Although this feature is dynamically relevant and can be extracted at higher values of $r$, i.e. when the reduced order model has more degrees of freedom, applying OMD at $r = 5$ we will not be able to extract this feature.

We set the measured output $y$ to the velocity readings from the potion of the time-varying flow field where $0.2 \leq y/D \leq 0.5$ and $0.85 \leq x/D \leq 1.25$. This is the portion of the velocity field that is slightly upstream of the peak of the vortex shedding mode, see Figure 3.6(c), although there is some spatial overlap between the vortex shedding feature and this region. Applying OMDor at $r = 5$ and $\alpha = 350$. The analysis was carried out at a range of $\alpha$ values and the value of $\alpha = 350$ was chosen to correspond with the best performance of the algorithm for these measurements. Table 4.2 lists the retained modes at the two different values of $\alpha$. Note that this region also includes part of the vortex shedding mode, and as a result, the retained mode's Strouhal number is lower than the expected $St \approx 0.4$.

| $\alpha$ | 0 | 350 |
|---|---|---|
| $St_1$ | 0 | 0 |
| $St_2$ | $\pm 0.20$ | $\pm 0.34$ |
| $St_3$ | $\pm 0.006$ | $\pm 0.02$ |

Table 4.2: Strouhal numbers of the identified features at $r = 5$, with different values of $\alpha$

The extracted OMDor modes include a mode at $St = 0.34$ whose spatial features can be seen in Figure 4.14. Note that this mode has a length-scale of 0.25, which is consistent with the fact that the vortex-shedding structures in Figure 3.6 have a length-scale of 0.5




(a) Re (b) Im

Figure 4.14: The harmonic of the vortex shedding mode retained by OMDor in application to an axisymmetric bluff body flow.

We note that the choice of measured output in this case, i.e. the velocity readings at a spatial subdomain containing 1215 grid points, is not realistic in terms of real-time use. In future work, a parametric study can clarify the role of the number, and the spatial distribution, of the velocity "sensors" for this flow.

## 4.4 Conclusion

A method for extraction of dynamical features from high-dimensional snapshots of the state of the system was proposed. In this method, the ability of the extracted modes in reconstructing the flow field is balanced with the ability of the resulting reduced order state to map the low dimensional state to a small set of model outputs. The model was applied to both synthetic and experimental data where localised coherent structures are

present. The results show that in both cases the application of OMDor enhances the ability of the extracted model to capture the subspace of the dataset most related to the measured outputs. This is true even in the case where more energetic dynamical features are present in the dataset. A parametric study was also carried out using the synthetic data, analysing the effect of practical limitations such as the dimension of the measured output, and the spectral and energetic contents of the flow field.

Chapter 5

# Preserving interpretability in data-driven order reduction

## 5.1 Order reduction and interpretable models

One of the primary challenges of reduced order modelling is to find a compromise between the interpretability of the reduced order state and the accuracy of the predictions afforded by the reduced order model (ROM). To expand upon this point of view, it is useful to draw a distinction between techniques which are motivated by linear theory—both in terms of feature extraction and dynamic modelling—and those which aim to move beyond these approaches and utilise nonlinear data-compression, feature extraction and dynamic modelling.

Classical approaches to data-driven model reduction in fluid mechanics typically appeal to linear operator theory in order to motivate their decomposition of the underlying flow into identifiable features. Examples discussed in §2.1, §2.3 and §2.4 include Proper Orthogonal Decomposition (POD), Dynamic Mode Decomposition (DMD) and Optimal Mode Decomposition (OMD). In particular, POD modes are simply the eigenvectors of a data ensemble's covariance matrix and provide an energy-optimal linear decomposition of a given data ensemble. DMD extracts features defined in terms of the eigenvectors of the

system matrix of a discrete-time linear system, fitted to optimally capture the evolution of the data's temporal POD coefficients over a single time-step. Finally, OMD generalises DMD in the sense that a linear POD-like projection operator is identified simultaneously with the discrete-time model fit.

By appealing to linear theory, each of these techniques has the advantage of a strong theoretical underpinning but, more importantly, there is transparency behind the selection procedure of the features (i.e., the POD, DMD or OMD modes) in each case. For example, if we apply POD to extract features from a data ensemble, we know that such features are an energy-optimal decomposition of the data. Consequently, given prior knowledge of the dynamic features of a given data-ensemble, a user can decide with confidence whether POD is an appropriate technique for feature extraction. However, a drawback of classical, linear, techniques such as POD, DMD or OMD, is that when the true dynamics of the underlying process are complex and nonlinear, ROMs built using the associated modes may not give the most efficient reduced-order representation of the true underlying dynamics. For example, while Koopman theory [93, 118, 146] (see also §2.3.1) implies that nonlinear dynamics can be approximated with models built using DMD modes, an arbitrarily large number of modes may be required for accuracy [10]. Consequently, a ROM built from modes or features identified by linear methods may require a larger number of states than necessary to capture dynamics arising from nonlinear phenomena.

The attempt to move beyond linearly-motivated feature extraction techniques can be viewed from two different perspectives, which can be made clear by revisiting the form of the ROM arising from the Optimal Mode Decomposition algorithm. In OMD, as detailed in §2.4, the model approximating the underlying high-dimensional dynamics has the form

$$LML^{\top} : \{\text{data at time } t_i\} \longmapsto \{\text{data at time } t_{i+1}\},$$

where $L \in \mathbb{R}^{p\times r}$ is a linear projection operator mapping from the (high) data dimension $p$ to a low-order subspace of dimension $r \ll p$. The matrix $M \in \mathbb{R}^{r\times r}$ describes the time-

evolution dynamics of this ROM arising from OMD. Clearly, being defined by matrices, both the action of the projection $L$ (data-compression) and dynamics matrix $M$ are linear processes. Either or both of these two components can potentially be replaced by nonlinear mappings or processes, thus generalising the classical approach to data compression and dynamical modelling. We now discuss these two generalisations.

**Nonlinear techniques for data compression.**

Techniques for nonlinear data compression are well-established in data science. The adoption of nonlinear data compression in fluid mechanics has only truly accelerated in the past decade, but it is now the case that powerful techniques from deep learning and computer-vision-inspired methods such as convolutional neural networks have now been brought to bear on the problem of feature identification in fluids [126, 56, 57, 133].

While the techniques for nonlinear compression are complex and varied, the fundamental modelling concepts are simple to explain. In particular, given a data ensemble comprising vectors $\{\boldsymbol{y}_i\}_{i=1}^N \subset \mathbb{R}^p$ of (high) dimension $p$, a user first defines a (small) reduction dimension $r$, then seeks to identify nonlinear functions $f : \mathbb{R}^p \mapsto \mathbb{R}^r$ and $g : \mathbb{R}^r \mapsto \mathbb{R}^p$ known, respectively, as the *encoder* and *decoder* mappings. To identify these mappings, at a given fixed reduction dimension $r$, nonlinear data-compression techniques seek to minimise residuals of the form

$$\sum_{i=1}^{N} \|\boldsymbol{y}_i - (g \circ f)(\boldsymbol{y}_i)\|_2^2 . \tag{5.1}$$

A clear advantage of this approach over linear techniques (e.g., over OMD in which $f = L^\top$ and $g = L$) is that the use of nonlinear encoder/decoder mappings can reduce the fitting residual for any chosen reduction dimension $r$. Thus, there is the beneficial potential to describe (i.e. compress) the data at a given accuracy level with fewer dimensions than are required by linear methods. As explained in §2.5.1, if functions $f$ and $g$ are both linear, the optimal results are equivalent to projection onto the POD modes.

A significant drawback of nonlinear data compression techniques, especially important in the context of fluids analysis, is that there is no longer a clear definition of a physically meaningful "feature". While in POD, DMD and OMD, the underlying appeal to linear theory provides natural features or modes arising directly from eigenvector analysis of the underlying matrices, the use of nonlinear (and typically highly complex) encoder and decoder functions prevents simple identification of distinct coherent features. Recent studies have attempted to rectify this issue, for example, by constructing a decoder function $g : \mathbb{R}^r \mapsto \mathbb{R}^p$ that generates $r$ vectors of length $p$ whose linear combination can reconstruct the input snapshot [126, 57]. The temporal evolution of these vectors forms a nonlinear analogue of "modes" from linear theory. Other approaches include analysis of the temporal evolution of the compressed vector $f(\boldsymbol{y}_i)$ [133]. However, while these autoencoders achieve lower reconstruction errors for a given value of $r$, each of their extracted "modes" contains elements of multiple linear modes and further analysis is necessary to interpret the state of the flow.

**Nonlinear techniques for data-driven system identification.**

The second way in which nonlinear modelling can extend classical data-driven approaches is to provide improved fits to the temporal evolution observed in a typical time-resolved data ensemble. Typical ensembles in fluid mechanics are high-dimensional, meaning that a data-compression technique (either linear or nonlinear) must first be applied before model fitting can be attempted (c.f. the discussion in Section 2.3).

The simplest approach towards nonlinear data-driven modelling, given a time-resolved data ensemble is as follows. First reduce dimensions using a given, and fixed, data compression technique. Next, extract the corresponding $r$-dimensional time series of reduced-order coefficients. Finally, fit an ordinary differential equation to these time series of any chosen complexity. Such an approach includes the recently-proposed Sparse Identification of Nonlinear Dynamics (SINDy) algorithm [32] in which POD is used for data-compression, and nonlinear fitting to the reduced-order time series is performed by

fitting multivariate polynomials of arbitrary degree (although it should be noted that the fit itself is performed using a standard regularised least-squares optimisation algorithm). The nontrivial, but still classical, component of the SINDy algorithm is to impose an $\ell_1$-regularisation term to promote sparsity and to prevent overfitting.

The final generalisation to fully nonlinear feature identification is to attempt to implement both nonlinear data compression (i.e. identification of encoder and decoder functions) and nonlinear system identification (i.e. identification of a time-stepping model) *simultaneously*. In a sense, this approach can be viewed as a nonlinear counterpart to OMD's generalisation of DMD in which the POD projection used by DMD is replaced by an arbitrary projection matrix $L$, which is identified simultaneously with the dynamics matrix $M$. Such a nonlinear compression/identification approach has been recently proposed by in [37], where a feedforward neural network simultaneously finds the nonlinear mapping to a lower dimensional state and ensures that the evolution of the extracted state can be expressed by a sparse dynamical model.

As a summary of the above discussion, a schematic overview of the different choices of linear and nonlinear data compression and system identification methods is given in Figure 5.1.

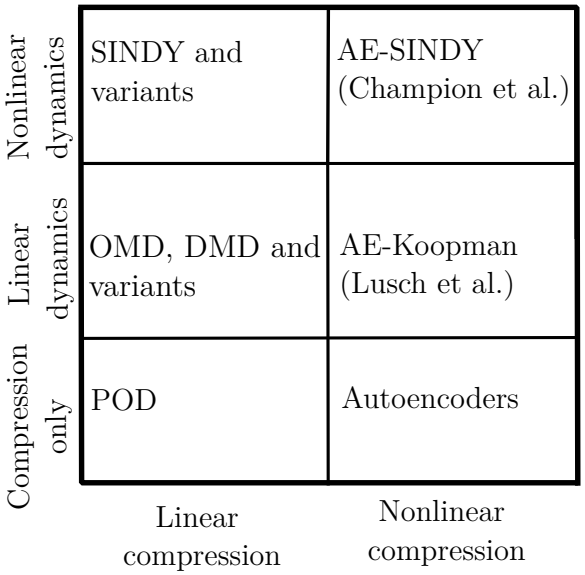



Figure 5.1: Categories of order reduction and model discovery methods

**What is the 'most appropriate' choice of compression or dynamic model?**

Given the ever-expanding range of model reduction and dynamics modelling tools available, it is obvious to ask which technique is best to apply in a given application. Naively, given that the underlying systems that we would like to model in typical fluid mechanics applications are nonlinear, it should be expected that nonlinear compression and nonlinear system identification will always provide better ROMs than classical, linear, methods. However, this may not always be the case.

Moving to nonlinear identification necessitates an increase, often dramatic, in the number of optimization variables which, in turn, leads to the possibility of overfitting and poor model performance off training data. Many successful implementations of nonlinear machine learning avoid overfitting due to the availability of arbitrarily large sets of training data. In fluid mechanics, however, there is typically a large cost associated with data collection, either in terms of computational cost or experimental limitations. Consequently, data is often incomplete (e.g. only certain velocity components or fields of view are available) and is only available at a small number of operating conditions (e.g., Reynolds numbers, actuator inputs or flow geometries). Importantly, the use of the word 'small' should be viewed in the context of all the possible dynamic behaviours that could be observed in a particular flow and, in many practical applications, it would not be unreasonable to replace 'small' with 'vanishingly small'.

In light of this typical lack of data in a fluid mechanics context, it should not automatically be expected that appealing to the most general nonlinear model fitting techniques will succeed. Indeed, while some promising progress has been made in cases where the sparse coordinates are known—such as handling noise [150] and partially captured states [30]—simultaneous order reduction and model discovery remains limited to fully captured, noise-free datasets [37].

It is therefore arguable that a middle ground must be found if the benefits of nonlinear system identification are to be enjoyed in a wide range of fluids applications. Recent

studies have concentrated on using deep learning to the Koopman invariant subspace of nonlinear models [160, 134, 110](see §2.5.4 for an in-depth review of these methods.) However, generally, these methods do not scale well with high dimensional and noisy data, due to prohibitively computationally expensive optimisation problems associated with them.

A recent study has considered the possibility of improving the performance of linearly motivated DMD analysis by replacing the initial POD-based compression with nonlinear compression techniques. However, in this approach, the extracted spatial features seem to suffer from a lack of interpretability [49].

In this chapter, motivated by the challenge of maintaining the interpretability of extracted features from data compression, we explore the possibility of extracting a sparse state to represent a fluid flow which maintains a connection to an existing interpretable encoding of the flow. To this end, rather than discarding the first, linearly-motivated, class of flow feature extraction techniques entirely, we aim to bootstrap from them.

Our approach is to first find an interpretable decomposition of the given data, using a linearly-motivated modal decomposition algorithm. Subsequently, we post-process the expansion of the underlying data-ensemble in the chosen modal basis to attempt to find a lower dimensional state that can be mapped to a, hopefully lower-dimensional, decomposed state. Enforcing a relationship between the lower dimensional state and the original interpretable modes ensures that the low dimensional state is by definition interpretable. Finally, we investigate whether the features identified by our new approach can enable more accurate dynamic ROMs to be created for the underlying data ensembles.

Since the first step in our method is to apply a linear modal decomposition technique, we first revisit these classical methods and discuss them in terms of both the interpretability of their features and the efficiency with which these enable the decomposition of the underlying data ensemble.

### 5.1.1 The efficiency of linearly motivated modal decomposition techniques in the context of extracting interpretable states

In previous chapters, we saw that the process of finding patterns of dynamical behaviour from ensembles of flow field snapshots typically involves identifying spatially coherent structures that exhibit interpretable temporal behaviour. To motivate the new nonlinear methodology considered in this chapter, we very briefly discuss the *efficiency* of the most common linearly-motivated decomposition methodologies, POD and DMD.

As explained in Chapter 2.1, given a data ensemble $X \in \mathbb{R}^{p\times n}$ whose columns are flow snapshots $x_i$ sampled at times $t_i$, POD solves the optimisation problem

$$\begin{aligned} \underset{\Phi}{\text{minimize}} \quad & \|X - \Phi\Phi^\top X\|_F^2 \\ \text{s.t.} \quad & \Phi^\top\Phi = I,\ \Phi \in \mathbb{R}^{p\times r}. \end{aligned} \tag{5.2}$$

The POD modes $\phi_i$ are the columns of $\Phi$, and an approximation of the original data set is then given by

$$x_j = \sum_{i=1}^{r} a_i(t_j)\phi_i + \nu_i, \qquad j = 1, \ldots, n \tag{5.3}$$

where $\nu_i$ are fitting residuals and $a_i(t_j)$ are the POD time series coefficients defined at each of the sample times $t_j$. These are computed by the method described in Section 2.1.

Now, POD is optimally efficient in the sense that for a given reduction dimension $r$, the POD modes $\{\phi_i\}_{i=1}^r$ are the unique vectors (flow fields) for which the fitting residual error $\sum_{i=1}^r \|\nu_i\|_2^2$ is minimised. However, such energetic fitting efficiency may not be the only criteria of interest for modelling.

Indeed, the primary motivation for the introduction of DMD is the fact that optimal energy efficiency (i.e. POD) may not be desirable if one would like to model flow structures which are purely oscillatory in time. This was first observed in one of the earliest papers employing DMD [146], where two dominant global modes of a jet in crossflow are analysed: one corresponding to a high-frequency shear layer mode of the jet; and the

other relating to low-frequency wall vortex oscillations. While a POD analysis extracts modes which can be argued to reasonably represent each feature, an analysis of their associated temporal information (via the coefficients $a_i(t_j)$) shows that both POD modes contain multiple frequencies, including ones which are clearly distinct from those known, via spectral analysis of DNS data, to be associated with the features of interest.

A consequence of the above observation is that if it is expected that flow features exist which are purely oscillatory, then POD may not necessarily provide the most efficient representation of such features. Indeed, the fact that the POD modes corresponding to the oscillatory structures of a jet in crossflow contain multiple frequencies indicates that multiple POD modes may be required to capture a single globally oscillatory structure in any resulting reduced-order model. Clearly, such models would not be of the smallest possible dimension and, in this sense, can be viewed as inefficient.

Dynamic Mode Decomposition was introduced to address this inefficiency specifically. If the user believes that an important structure in the flow can be represented by a linear process (a pure oscillator, or an oscillator with exponential growth or decay) then it makes sense to attempt to extract modal structures whose associated temporal coefficients (i.e. the analogues of POD's $a_i$ coefficients) exhibit exactly this linear behaviour. As explained in Section 2.3, DMD provides a decomposition of a data ensemble $X$ via

$$x_j = \sum_{i=1}^{r} \alpha_i \phi_i^{\mathrm{DMD}} \lambda_i^{j-1} + \nu_i^{DMD}, \tag{5.4}$$

where the DMD modes $\phi_i^{\mathrm{DMD}}$ are (complex) linear combinations of POD modes, and the eigenvalues $\lambda_i$ are chosen to optimally reduce the fitting residual $\sum_{i=1}^{n} \|\nu_i^{DMD}\|_2^2$. This decomposition implies that temporal coefficients of each DMD mode are, by (5.4), the terms $\alpha_i \lambda_i^{j-1}$, which endows each structure with a single frequency and growth rate. It follows that DMD is an optimally efficient method of decomposing a data-ensemble into structures with linear dynamics*: if a purely oscillatory feature of interest is present in the

* Given that such structures lie in the span of the POD modes

data, then DMD has the ability to represent it with only a single mode in a reduced-order model. Furthermore, such a feature can be argued to be clearly interpretable since, by definition, it represents the flow structure that was assumed to exist at (or near) a given frequency.

It should now be clear that the two classical methods of DMD and POD are both optimally efficient in an appropriately defined sense, and that DMD's version of 'optimal efficiency' can be of benefit to reduced-order modelling if the underlying flow structures are believed to be purely oscillatory. In this chapter, we attempt to address the following natural question. In scenarios where linear decomposition techniques are not optimally efficient and lead to a larger number of states than necessary, can nonlinear post-processing of their resulting states lead to improved efficiency? In other words, can the information captured by a large set of linear modes be represented using a smaller state?

In §3, it was seen that when applied to dynamically complex data ensembles, DMD can give a large number of distinct modes with both similar characteristic temporal frequencies and with similar spatial structures. Such behaviour may arise from noise in the underlying data set or, more interestingly, from the flow's non-exponential transient behaviour away from (and back to) its global attractor [10]. In §3 we considered an arguably extreme solution to ROM efficiency in this context, by post-processing the DMD modes to form clusters of dynamically similar modes. Subsequently, once the clusters of modes are found, each cluster was represented by one, or a small number of, spatial modes. Representing each cluster with a single, or a few, modes reduces the complexity of the decomposition and promotes a more intuitive picture of the degrees of freedom available in the underlying system.

Nonetheless, a cluster's representative mode spans a more limited subspace of the dataset than all of the modes in that cluster. In cases where dimensionality reduction is a priority, this may be a worthwhile sacrifice. Ideally, however, one would want to model the evolution of a larger set of, e.g. all of, DMD modes in the cluster, but with an efficient low-order model which contains a low number of degrees of freedom. Such a goal may

be possible if there is coherent nonlinearity in the underlying data ensemble. For example, excursions away from and relaxations to a global attractor are nonlinear processes. While it is known that this behaviour can be captured by a large (theoretically infinite) number of linear processes [10], nonlinear data compression has the potential to enable a more efficient representation of this phenomenon. Furthermore, a standard application of DMD (or the cluster analysis of §3) does not explicitly seek to exploit nonlinear temporal dependencies between modes, such as those exhibited by harmonics of oscillatory global modes. This provides further potential for nonlinear dimensional reduction of complex data ensembles.

We now present our methodology for nonlinearly-enabled dimensional reduction, based on interpretable data decomposition.

## 5.2 Methodology

As is standard, we assume that a matrix $X \in \mathbb{R}^{p\times m}$ of $m \in \mathbb{N}$ snapshots is available whose columns $x_i \in \mathbb{R}^p$ contain flow field information at sample times $t_1 < t_2 < \cdots < t_{m-1} < t_m$. We first apply a classical, linearly-motivated, modal decomposition methodology and extract a set of $n \leq m$ spatial mode shapes $\{\phi_i\}_{i=1}^n$. These modes may be created by a standard application of a linear modal decomposition methodology, such as POD, DMD or OMD. Alternatively, we also allow the possibility that these modes have been created by post-processing a standard modal ensemble using a method such as the cluster analysis of Chapter 3.

Our only prerequisite is that for the computed set of modes $\{\phi_i\}_{i=1}^n$, it is possible to create time series $(a_i(t_j))_{j=1}^m$ for each mode $i = 1, \ldots, n$. Again, such time series can be calculated using one of the methods described in §3. To begin, we use the time series $(a_i(\cdot))_{i=1}^n$ of the original modes to give an initial, and physically interpretable, projection of the the flow field. Noting that while $n < p$, where $p$ is the number of data points per snapshot, it is still the case that $n$ may be chosen to be relatively large—typically

$n = \mathcal{O}(10^2)$. Our aim, indicated schematically in Figure 5.2, is to attempt to find a lower-dimensional set of time series $(b_i(\cdot))_{i=1}^r$, with $r < n$, which can accurately approximate the time series $(a_i(\cdot))_{i=1}^n$ arising from the underlying projection of the flow field.

To achieve this, we will use the nonlinear methodology of *autoencoder neural networks*. An autoencoder neural network consists of two fundamental components. The first is a non-linear *encoder* function that maps the inputs (in this case the original time series $(a_i(\cdot))_{i=1}^n$) to produce a lower-dimensional state $(b_i(\cdot))_{i=1}^r$ applied to the input. These lower-dimensional states are typically referred to as the *latent states* or as the *encoded state*. Finally, the encoded states are then fed into a second nonlinear *decoder* function which, via the solution to an optimisation problem of the general form (5.1), in turn attempts to reconstruct the original higher dimensional state $a_i(\cdot)$.

Figure 5.2 shows a schematic view of this methodology. Our aim now is to define appropriate encoder and decoder architectures in which to search for the required nonlinear mappings. Since we are attempting to model time-series, we need to use encoder and decoder functions that consider the temporal nature of the input vector $(a_i(\cdot))_{i=1}^n$. Recurrent Neural Networks (RNNs) are explicitly designed to handle temporal signals, and we briefly recap (from §3.1), and then expand upon, some of their key properties.

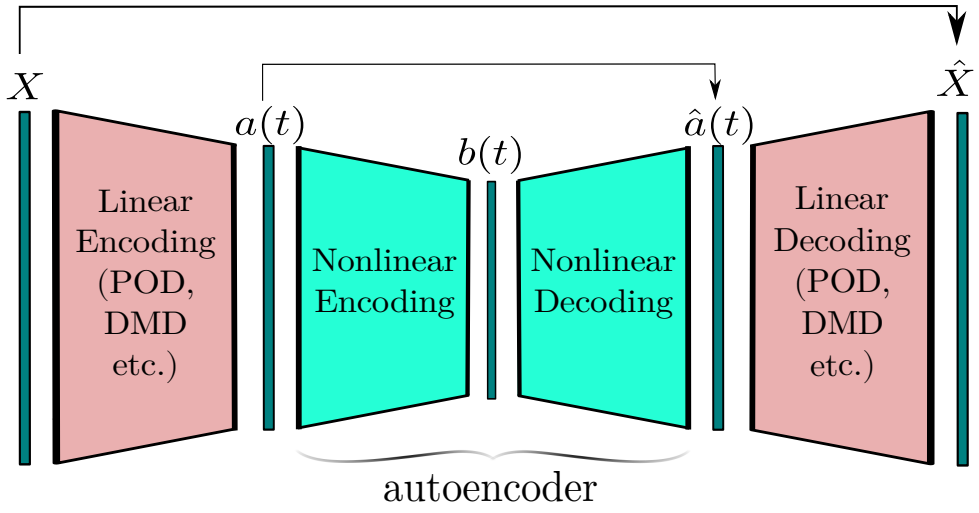



Figure 5.2: A schematic view of the proposed approach to interpretable dimensionality reduction.

### 5.2.1 A brief review of Recurrent Neural Networks

Recurrent neural networks (RNNs) are a class of neural networks widely used for processing time series data, due to their ability to contextualise the input vector $(a_i(\cdot))_{i=1}^n$ with information from previous time steps [85, 50, 80, 41]. The advantage of using recurrent neural networks, over other temporal modelling approaches such as Markov chains, is the automatic manner in which patterns and dependencies are found, and their capability to extract complex nonlinear dependencies and patterns [104]. The disadvantage of using RNNs, compared to other time series modelling techniques, is the difficulty associated with solving their corresponding optimisation problem [136](see §2.5.5 for a detailed discussion).

Elman recurrent neural networks are the simplest type of RNNs [50] and have been widely used for time series analysis [170, 102, 7, 38], especially in the context of natural language modelling and computational linguistics [121, 51]. To begin our introduction to RNNs, we first describe the simplest type of Elman RNN, which includes only one hidden layer. In the context of the current analysis, we will explain the network as an encoder function that maps a higher dimensional sequence $(a_i(\cdot))_{i=1}^n$ to a lower dimensional state $(b_i(\cdot))_{i=1}^r$.

This provides a relation between inputs $(a_i(t_j))$ sampled at times $t_1 < t_2 < \cdots < t_{m-1} < t_m$ and the latent states $(b_i(t_j))_{i=1}^r$. For each $j$, we let $a(t_j) = (a_i(t_j))_{i=1}^n \in \mathbb{R}^n$ be the vector of input weights at time $t_j$. The output $b(t_j) = (b_i(t_j))_{i=1}^r \in \mathbb{R}^r$, or latent state vector, at time $t_j$ is then given by

$$\begin{aligned} h(t_j) &= \tanh(W_{hi}a(t_j) + W_{hh}h(t_{j-1}) + d_h) \\ b(t_j) &= W_o h(t_j) + d_o. \end{aligned} \tag{5.5}$$

Here, $h(\cdot) \in \mathbb{R}^\ell$ is known as the hidden state, $W_o \in \mathbb{R}^{r\times\ell}$, $W_{hi} \in \mathbb{R}^{\ell\times n}$ and $W_{hh} \in \mathbb{R}^{\ell\times\ell}$ are weight matrices, $d_o$ and $d_h$ are constant vectors. The dimension $\ell$ is a modelling parameter and the $\tanh(\cdot)$ is applied to each component of its argument vector.

To explain the form of this model, first consider (5.5) but with the $\tanh(\cdot)$ term removed from the update equation for the hidden state $h(t_j)$. In this case, the mapping between $a(t_j)$ and $b(t_j)$ is simply that of a discrete-time linear system, forced by the input $a(t_j)$ and with output $b(t_j)$. As a consequence, information from previous time-steps can have a delayed influence on the input-output relation.

Furthermore, recalling the series expansion $\tanh(x) = x - \frac{1}{3}x^3 + \cdots$ about $x = 0$, it follows that the Elman RNN (5.5) both retains the ability to capture linear input-output relations (i.e. when the input parameters to the hyperbolic tangent are in the vicinity of zero) and extends to the possibility of capturing nonlinear input-output behaviour. Importantly, the fact that tanh saturates for inputs of large absolute values (at a first approximation via the effect of the $-\frac{1}{3}x^3$ term) implies that structure is imposed on the form of the nonlinear relation, which attempts to prevent the fitting of temporally unreasonable relations. The inclusion of temporal dynamics via the term $h(t_{j-1})$ in (5.5) implies that the Elman RNN has the ability to exploit any temporal coherence in the underlying data. Since we expect some level of temporal correlation to be present in the high-dimensional state $a(t_j)$, including time-dependency in the RNN model may be beneficial in terms of obtaining an efficient low-order representation of the data ensemble.

To capture complex nonlinear relations, it may be necessary to increase the number of hidden layers in an RNN. In a similar fashion to fully connected neural networks, increasing the number of hidden layers, with each hidden state being used as the input of the next hidden layer, increases the complexity of the nonlinear fitting function (see §2.5.1 for fully-connected NNs). Note that, depending on the specifics of the problem, alternative non-linearities can be used instead of the $\tanh(\cdot)$ function. Such non-linearities must have similar saturation characteristics.

The potential shortcoming of using an Elman network is that it is not not suitable to model long term dependencies. Suppose that in the example above we have a desired value of the latent state $(b_i(t_j))_{i=1}^r$, and we want to train the network so that the cost function $J = \|b(t_j) - (\tanh(W_o h(t_m) + d_o))\|_F^2$ is minimised. Suppose also that the in-

put sequence $(a_i(t_k)_{k=1}^m)_{i=1}^n$ includes a large number of samples, i.e. $m \gg 1$. Defining $y_j = \tanh(W_o h(t_m) + d_o)$, to update the weights $W_{hh}$ using a gradient-based algorithm, we use the chain rule to find the gradient

$$\frac{\partial J}{\partial W_{hh}} = \sum_{k=1}^{j} \frac{\partial J}{\partial y_j} \frac{\partial y_j}{\partial h(t_j)} \Big( \prod_{i=k+1}^{j} \frac{\partial h(t_i)}{\partial h(t_{i-1})} \Big) \frac{\partial h_k}{\partial W_{hh}}, \tag{5.6}$$

where $\prod$ signifies the product of the Jacobian terms

$$\frac{\partial h(t_i)}{\partial h(t_{i-1})} = W_{hh} \ \text{diag}(\tanh'[W_{hi} a(t_i) + W_{hh} h(t_{i-1}) + d_h]), \tag{5.7}$$

where $\tanh'(\cdot) = 1 - \tanh^2(\cdot)$. In a gradient-based solver, the gradient in (5.6) is used to update the weights $W_h h$ in the next iteration of the gradient descent optimisation. The overall update vector is determined as the sum of all time steps $k$'s contribution. However, the gradient of the $\tanh(\cdot)$ function has an upper bound of 1 and a lower band of 0 and if the largest eigenvalue of the matrix $W_{hh}^t$ is also stable, for a long sequence, the contribution from the initial time steps to the direction of movement of the solver will be negligible. As a result, the choice of weights prioritises to only model short-term dependencies. On the other hand, if the norm $\|W_{hh}^t\|_2 > 1$, or the gradient of the nonlinearity is not smaller than 1, the gradient terms will grow unstably, preventing the convergence of the optimisation algorithm. This problem is referred to as the *vanishing and exploding gradients* [80, 136].

Note that in cases where the input sequence does not have to be long, i.e. you can reasonably expect to have a good estimate of the output based on a few inputs, given good initialisation practice, see [64], and setting an upper bound for the gradients, an Elman RNN can still be used effectively.

For more complex datasets, where the use of longer input sequences may be necessary, often alternative mapping techniques, such as Long Short-Term Memory, LSTM, networks [80] or Gated Recurrent Unit, GRU, networks [41]. LSTMs utilise a more sophisticated mechanism of assigning priority to information from previous time steps. LSTMs utilise

a cell state, $c(t)$, and various *gate functions* that allow the network to store and forget about past states. An LSTM network with one hidden layer can be written as

$$\begin{aligned} i(t_j) &= \sigma(W_{ii}a(t_j) + W_{hi}h(t_{j-1}) + d_i) \\ f(t_j) &= \sigma(W_{if}a(t_j) + W_{hf}h(t_{j-1}) + d_f) \\ o(t_j) &= \sigma(W_{io}a(t_j) + W_{ho}h(t_{j-1}) + d_o) \\ g(t_j) &= \tanh(W_{ig}a(t_j) + W_{hg}h(t_{j-1}) + d_g) \\ c(t_j) &= f(t_j) \odot c(t_{j-1}) + i(t_j) \odot g(t_j) \\ h(t_j) &= o(t_j) \odot \tanh(c(t_j)) \\ b(t_j) &= W_o h(t_j) + d_o, \end{aligned} \tag{5.8}$$

where $W$ denotes weight matrices, $d$ denotes constant vectors, $\sigma$ denotes the sigmoid function with the range $\sigma \mapsto [0, 1]$, and $\odot$ denotes the elementwise multiplication operation. The functions $i(t), f(t), o(t)$ are known as gate functions, that allow or limit the passage of information from previous time steps to the current output and from the current time step to the future time steps. The function $g(t)$ contains the candidate state, which if the input gate $i(t) > 0$, can contribute to the cell state $c(t)$. The output gate $o(t)$ controls whether information from the cell state can pass to the current output.

As with the Elman neural networks, differentiating the recursive state will determine the flow of gradients from the output error to the weight matrices. However, here we have two recursive states $h(t)$ and $c(t)$. The flow of gradients to $h(t)$ is determined in a similar fashion to the Elman model. However, by differentiating the cell state $c(t_i)$ with respect to its previous value $c(t_{i-1})$ we will get the

$$\frac{\partial c(t_i)}{\partial c(t_{i-1})} = f(t_i),$$

which can be used to obtain

$$\frac{\partial c(t_m)}{\partial c(t_1)} = \prod_{i=2}^{m} f(t_i).$$

While this gradient can also diminish for longer sequences, there is no inherent reason for it to do so [17, 104]. In other words, unlike the Elman RNN where the convergence of the gradient-based algorithm dictates that long-term dependencies are ignored, the above gradient can in theory allow the passage of long-term dependencies, while the optimisation routine also converges to a solution.

An illustrative example is that of initiating the constant biases $d_f$ at a high value and only allowing small variations to it during the training process, through arbitrary clipping of its gradients. In such a case, the forget gate value, $f(t)$, will remain constant at $f(t) = 1$ and the derivative $\frac{\partial c(t_m)}{\partial c(t_1)} = 1$. Indeed the original LSTM algorithm was designed with the function $f(t_i) = 1$, and the forget gate was later designed as a way to allow the model to reset the cell state's value and ignore previous inputs. In deeper LSTM networks, the cell state $c(t)$ and the hidden state $h(t)$ are both passed on to another LSTM cell, with its own gate.

Another approach to addressing the problem of exploding/vanishing gradients in RNNs is that of Gated Recurrent Units, in (5.9). GRU models are more compact than LSTM networks, with only two gate functions. A reset gate, $r(t)$, that controls the flow of information from the previous time step to the current time step, and an update gate, $z(t)$, that controls the passage of information from the current time step to the future time steps. GRUs achieve the same goal of allowing information to pass through long recursive chains, with fewer parameters than LSTM networks.

$$\begin{aligned} z(t_j) &= \sigma(W_{zi}a(t_j) + W_{zh}h(t_{j-1}) + d_z) \\ r(t_j) &= \sigma(W_{ri}a(t_j) + W_{rh}h(t_{j-1}) + d_r) \\ n(t_j) &= \tanh(W_{ni}a(t_j) + W_{nh}(r(t_j) \odot h(t_{j-1})) + d_h) \\ h(t_j) &= (1 - z(t_j)) \odot n(t_j) + z(t_j) \odot h(t_{j-1}) \end{aligned} \tag{5.9}$$

Armed with the knowledge of RNNs as a class of sequence-to-sequence regression models, we now explain their use in the context of our order reduction approach for fluid mechanics data ensembles.

### 5.2.2 Our time-lagged autoencoder

Here we give a more detailed explanation of the method illustrated in Figure 5.2. Suppose we have applied a linearly-motivated modal decomposition method to a snapshot ensemble $X \in \mathbb{R}^{p\times m}$ and extracted a set of $n \in \mathbb{N}$ mode shapes $\Phi = \{\phi_i\}_{i=1}^{n}$. Depending on the decomposition technique used, the mode shapes may have different characteristics. Modes extracted using POD are by definition real-values and pairwise orthonormal, whereas modes extracted using DMD-like algorithms are complex-valued and are not in general spatially orthogonal.

For an approach based on DMD-like methodologies, after selecting an initial subset of the available modes and applying any predefined post-processing steps (e.g. mode clustering and representative modes), we obtain a set of complex modes $\Phi_{\text{DMD}} \in \mathbb{C}^{p\times n}$. For further analysis, we then consider the underlying mode ensemble as being formed of the real and imaginary parts of each mode, i.e. $\Phi = \{\text{Re}((\phi_i)_{\text{DMD}}), \text{Im}((\phi_i)_{\text{DMD}})\}_{i=1}^{n} \subset \mathbb{R}^{p\times 2n}$, allowing the real and imaginary parts of the mode to represent independent spatial mode shapes. As a result, the time series $a(t)$ associated with each mode will be real-valued.

Once the mode shapes have been computed, we find the mode shape time series $a(t)$ through the least square fit that minimises $\|X - \Phi a(t)\|_F^2$. The solution has the form

$$\begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ a(t_1) & a(t_2) & \cdots & a(t_m) \\ \downarrow & \downarrow & & \downarrow \end{pmatrix} = \left(\Phi^\top \Phi\right)^{-1} \Phi^\top X. \tag{5.10}$$

Note that for the case of orthonormal $\Phi$, the right-hand side of (5.10) can be simplified to $\Phi^\top X$.

Once the mode time series are found, we need to train an RNN autoencoder to find an efficient approximation $\hat{a}(t) \approx a(t)$ which depends upon a reduced number of latent states. The appropriate architecture for the autoencoder network depends on the number of data points and the quality of the data available. Here we explain the two architectures that we have used in the results section that follows.

Firstly, for practicality, we design the encoder function such that at each time $t_j$ it will only have access to a shorter sequence of snapshots than the full set of snapshots. This significantly reduces the computational cost of training the network as well as enabling the encoder to be used in for practical estimation purposes. We refer to the subset of the time series that the encoder function has access to as the input sequence and it can be expressed as $(a_i(t_{j-kd(\Delta t)})_{k=0}^{L})_{i=1}^{n}$, where $d(\Delta t)$ is the constant time difference between the snapshots in the input sequence, and $L+1$ is the number of snapshots in the input sequence. Note that $d$ is an integer and $d(\Delta t)$ is a multiple of the sampling time step $\Delta t$.

For data sets that are highly periodic and contain a low magnitude of noise, a shorter input sequence can be sufficient, while more complex datasets may require a longer input sequence to contextualise the true dynamics at time $t_j$. For the first case, we use an Elman neural network with two hidden layers as the encoder function. The encoded state $b(t)$ is defined by

$$\begin{aligned} h_1(t_j) &= \tanh\left(A_1 h_1(t_{j-1}) + B_1 a(t_j) + d_1\right) \\ h_2(t_j) &= \tanh\left(A_2 h_2(t_{j-1}) + B_2 h_1(t_j) + d_2\right) \\ b(t_j) &= C h_2(t_j) + d, \end{aligned} \tag{5.11}$$

where $A_i, B_i$ and $C_b$ are weighting matrices describing the underlying linear processes, and $d_i, d_b$ are constant bias vectors. As mentioned above, the state $a(t_j)$ is sampled such that, $a(t_j) \in (a_i(t_{j-kd\Delta t})_{k=0}^{L_i})_{i=1}^{n}$, and consequently the hidden state $h(t_j)$ and the encoded state $b(t_j)$, are evaluated at the same time steps as $a(t_j)$.

For most practical cases, the decoder function should be able to produce an approximation of the flow field at time $t_j$, given the value of the encoded state $b(t_j)$. We, therefore, design the model such that it has no temporal dependency. We use a fully-connected architecture with two hidden layers for the decoder function letting

$$\begin{aligned} h_3(t_j) &= \tanh(B_3 b(t_j) + d_3) \\ h_4(t_j) &= \tanh(B_4 h_3(t_j) + d_4) \\ \hat{a}(t_j) &= \hat{C} h_4(t_j) + \hat{d}. \end{aligned} \tag{5.12}$$

Note that the decoder in (5.12) has the same structure as an Elman network, without the temporal recursion.

The optimal weight matrices and bias vectors are found by solving the optimisation problem

$$\underset{\boldsymbol{W}}{\text{minimise}} \quad J := \sum_{j=1}^{n} \left\| a(t_j) - \hat{a}(t_j|\mathbf{W}) \right\|_F^2, \tag{5.13}$$

where

$$\boldsymbol{W} = \left\{ (A_i)_{i=1}^2,\, (B_i)_{i=1}^4,\, (d_i)_{i=1}^4, C, \hat{C}, d, \hat{d} \right\}$$

is the set of all weight matrices and bias vectors involved in the definitions of the encoder and decoder functions. It should be noted that $\hat{a}(t_j|\boldsymbol{W})$ implicitly depends on the original time series data $a(\cdot)$ via the input-output relations

$$a(\cdot) \xrightarrow{(5.11)} b(\cdot) \xrightarrow{(5.12)} \hat{a}(\cdot).$$

We note that similar architectures have been used extensively in natural language processing and sentiment analysis, fields that rely heavily on time-series analysis (see [42] and [16] for example). We also note that previous attempts at using RNNs in conjunction with the POD time series have been successfully used in creating a full dynamical system for forecasting and feedback control [156].

However, the distinct advantage of the above architecture is its suitability for model order reduction. Unlike the sequence-to-sequence autoencoder in [42], where the decoder reconstructs all of the input sequence, and the dimensions of $a(t)$ and $b(t)$ are the same, we prioritise the extraction of a low dimensional state corresponding to the current time step. The simultaneous training of a decoder function that requires no memory of the past allows us to more easily use the low-dimensional state $b(t)$ for control and estimation purposes. The reason for this is that in typical estimation algorithms, at each time-step, a correction to the state estimate is obtained as a function of the difference between the expected output of the estimated state and the true measured output. Using a decoder which requires no memory, implies that the expected output (as required by an estimation algorithm) can be easily computed via a static map involving the decoder function. Thus, implementing such an estimation algorithm (and any controller which depends upon it) is simpler with a memoryless decoder.

At the same time, the choice of the RNN structures here allows for a sequential encoding step that compresses multiple instances of the high-dimensional state $a(t)$ into one instance of a low-dimensional state $b(t)$, uses the recursive hidden states $h$ to enrich the encoded state. In contrast, multiple time-steps of the full state $a(t)$ are simultaneously used as the input to the forecasting model in [156], leading to a larger and more difficult model to train as the dimension of the state $a(t)$ increases.

As mentioned, the limitations of Elman RNNs prevent their application on more complex quasi-periodic time series. For such cases, LSTMs or GRU neural networks can be used as the encoder function. Similarly, the number of hidden layers in the decoder function can change to reduce the model's complexity. Fundamentally, however, the cost function of the autoencoder remains, as expressed in (5.13), the squared Frobenius norm of the residual between the reconstructed $\hat{a}(\cdot)$ and the input time series $a(\cdot)$.

In some cases, it is beneficial to apply a pre-processing step to normalise the original time series $a(\cdot)$. While this is not strictly necessary, if not performed the least-squares nature of the optimisation problem (5.13) implies that the autoencoder approach may

prioritise the reconstruction of only higher amplitude time series at the expense of less energetic time series. Whether this behaviour is desirable will depend upon a prior assessment of the dynamic significance of (and the desirability to model) the various time series extracted from the underlying ensemble of spatial modes. If it is desirable, normalising the underlying time series data can fully de-emphasise the priority of the high-energy modes.

For each time series $a_i(\cdot)$, we apply the transformation

$$\alpha_i(t) = \frac{a_i(t)}{|a_i|}, \tag{5.14}$$

where

$$|a_i| = \max(a(t)) - \min(a(t)).$$

For cases where $\min(a(t)) < 0$ and $\max(a(t)) > 0$, which is usually the case for DMD and POD time series, this normalisation enforces $-1 \leq \alpha_i(t) \leq 1$.

The normalised time series $\alpha(\cdot)$ can then be used in place of $a(\cdot)$ as the input to the neural network. An additional term can then augment the cost function to ensure a good fit between the resulting reconstructed velocity field. The resulting optimisation, for Elman encoder and decoder functions, is

$$\underset{\mathbf{W}}{\text{minimise}} \quad J := \sum_{j=1}^{m} \left\| \alpha(t_j) - \hat{\alpha}(t_j|\boldsymbol{W}) \right\|_F^2 + \eta \left\| X_{\text{train}} - \sum_{j=1}^{m} \sum_{i=1}^{n} \left[ |a_i| \hat{\alpha}_i \left( t_j | \boldsymbol{W} \right) \right] \cdot \phi_i \right\|_F^2, \tag{5.15}$$

where $\boldsymbol{W}$ is the set of weights matrices and constant bias vectors, $\hat{\alpha}$ is the reconstructed normalised time series, $\eta$ is a constant which determines the balance between how well all the scales are replicated regardless of their energetic contribution, and how well the velocity field is replicated.

We now present two illustrative examples which examine the potential of, and the challenges faced by, the aforementioned approach for ROM.

## 5.3 Results

We will now apply the above approach to two bluff body flows, one at a low Reynolds number and dynamically simple and one at a high Reynolds number and more dynamically complex. As motivated above, our aim, given an initial and moderately large set of dynamic modes, is to robustly replicate the observed flow dynamics that lie in the span of these modes with fewer degrees of freedom than are required by the original decomposition.

To use the terminology of RNNs, this will involve identifying a latent state $b \in \mathbb{R}^r$, a nonlinear encoder $f : \mathbb{R}^n \mapsto \mathbb{R}$ and a nonlinear decoder function $g : \mathbb{R}^r \mapsto \mathbb{R}^n$ such that the norm of the residuals

$$\|a - \hat{a}\|_F = \|a - (g \circ f)(a)\|_F = \|a - g(b)\|_F$$

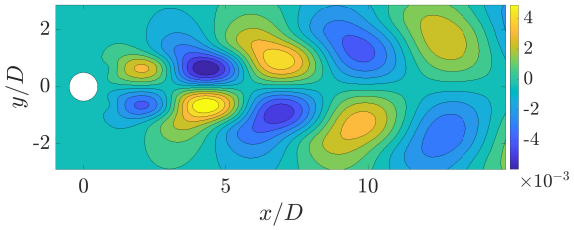
is small, when evaluated on the time series of temporal mode weights $a \in \mathbb{R}^n$. If a small residual can be achieved with $r \ll n$, this suggests that efficient data-compression is achievable. Furthermore, there may be potential to use the reduced state $b$ as the basis of a reduced order model. This will be explored subsequently, in §5.3.5.

### 5.3.1 Encoding POD coefficients for flow past a cylinder at low Reynolds number

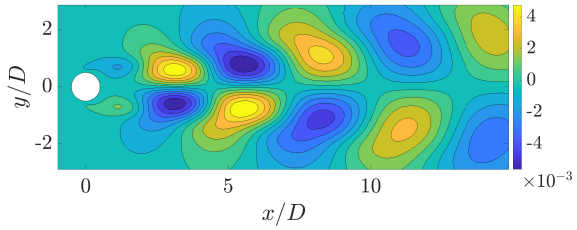
#### 5.3.1.1 Existing dynamical models

The supercritical laminar wake of a circular cylinder has been extensively used as a benchmark flow in existing literature [126, 56, 49, 110]. Noack et al. [128] applied POD to snapshots of a fully developed oscillatory wake, at Re = 100, and found the oscillatory POD modes that describe the dynamics of this flow field. They subsequently argue that a further, albeit less intuitive, mode of importance to the flow dynamics is the so-called *shift mode* which captures the difference between the time-averaged flow field evolving on

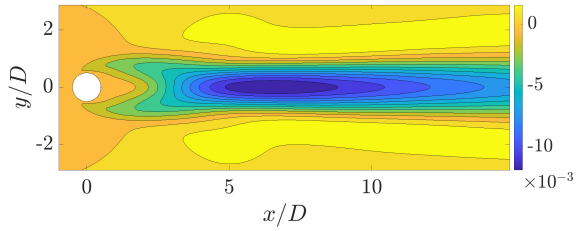
its global attractor (a stable limit cycle) and the unstable steady-state solution to the Navier-Stokes equations. For completeness, we have decomposed snapshots of a similar flow at Re= 60 using POD, and the three spatial mode shapes corresponding to these three features (one flow field for the shift mode, and a pair representing the dominant vortex shedding feature) are shown in Figure 5.3 and their corresponding time series are shown in Figure 5.4.




(a) Vortex shedding $\phi_1$



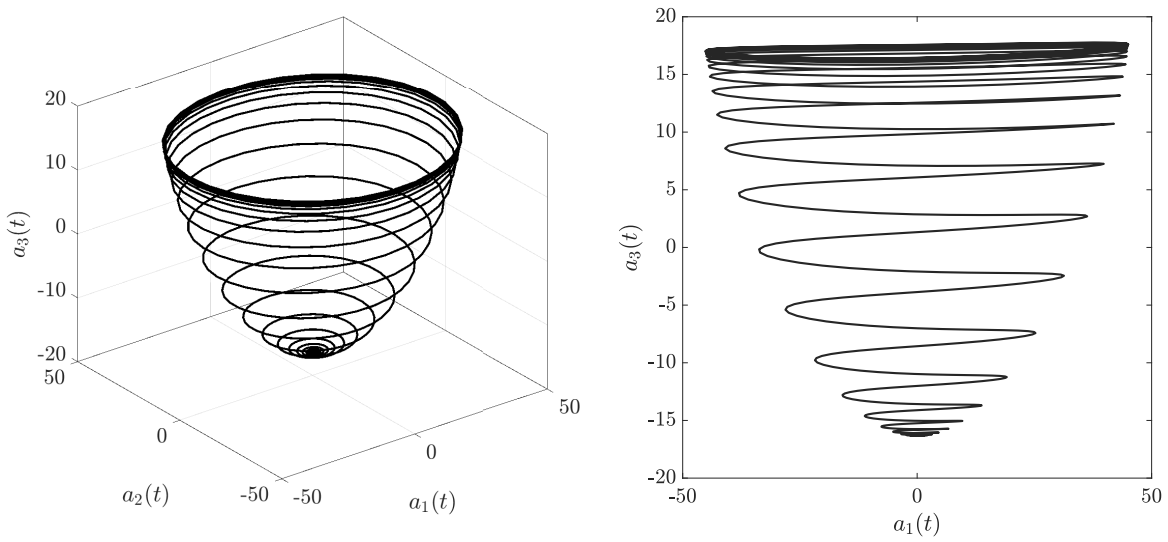

(b) Vortex shedding $\phi_2$




(c) Shift mode $\phi_3$

Figure 5.3: The three fundamental modes for cylinder wake at Re= 60: (a-b) vortex shedding modes; (c) the shift mode.

Interestingly, a simple canonical model for the evolution of these three modes is proposed in [128]. In particular, letting $a_1(t), a_2(t)$ be the time series corresponding to the two vortex shedding modes $\phi_1, \phi_2$, and $a_3(t)$ be the time series corresponding to the shift mode $\phi_3$—so that an approximation to the time-varying flow field is given by $\sum_{i=1}^{3} a_i(t)\phi_i$—the dynamic model has quadratic nonlinear form




(a) (b)

Figure 5.4: Vortex shedding time series, $a_1$ and $a_2$, and the shift mode time series $a_3$.

$$\frac{da}{dt} = \begin{pmatrix} \mu & -1 & 0 \\ 1 & \mu & 0 \\ 0 & 0 & -1 \end{pmatrix} a + \begin{pmatrix} 0 & 0 & -a_1 \\ 0 & 0 & -a_2 \\ a_1 & a_2 & 0 \end{pmatrix} a, \qquad a = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$$
$$:= \mathcal{A}a + N(a)a. \tag{5.16}$$

This minimal three-state model captures the unstable (about the equilibrium point $a = 0$) oscillatory structure of the vortex shedding mode pair $(a_1, a_2)$ and the transient transition of the flow onto its asymptotically stable limit cycle. The role of the shift mode is made clear by considering its evolution in the context of the oscillation amplitude

$$A = \sqrt{a_1^2 + a_2^2}.$$

Indeed, a simple calculation shows that

$$\frac{dA}{dt} = (\mu - a_3)A, \qquad \frac{da_3}{dt} = -a_3 + A^2,$$

suggesting that a limit cycle exits for $a_3 = A^2 = \mu$, and that close to this limit cycle (under the assumption that the shift mode is slaved via $a_3 = A^2$) the well-known Stuart-Landau equation for the oscillation amplitude is recovered:

$$\frac{dA}{dt} = \mu A - A^3.$$

It is shown in [128] that such simple nonlinear 3-state models, with coefficients obtained from fitting to DNS data, can capture well the general dynamics of the flow, including the nature (i.e. reaching a fixed amplitude and frequency over the limit cycle) of the limit cycle. However, due to unrealistic modelling of scale interactions, the position of the plane of oscillations in the three-dimensional phase space and the final amplitude of the oscillations, are not precisely captured by such a model.

Importantly, given our interest is in identifying a minimal basis with which to capture a flow's behaviour, it is further shown in [128] that it is possible to derive a low-order model with only two degrees of freedom that approximates not only the global mode oscillation, but also the shift mode and higher harmonics of the global mode. This is achieved by identifying an ODE for the evolution of the global mode coefficients $(a_1, a_2)$, via Galerkin projection of the Navier-Stokes equations, then subsequently quadratically slaving the behaviour of the shift mode and higher harmonics to the flow's evolution on this 2-dimensional invariant manifold. It is illustrated that this model performs better than a 3-mode model of the form (5.16) in predicting the amplitude and the plane of the oscillations.

Given the simple, albeit nonlinear, dynamics (5.16), it is perhaps not entirely unsurprising that the flow field of the cylinder wake at a low Reynolds number can be well approximated by a ROM with only two free states. However, to achieve this in [128] required an approach that appealed to both analytical insight and to detailed numerical analysis of the governing Navier-Stokes equations to extract resolved linear eigenfunction from which to extract the necessary Galerkin coefficients to underpin the model.

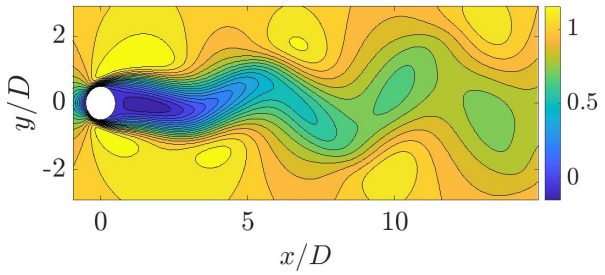



Figure 5.5: An indicative snapshot of the streamwise velocity for flow past a circular cylinder at Re=60.

Motivated by the fact that such a two-state model is achievable, in this section we investigate the question of whether similarly efficient models with equivalent accuracy can be identified automatically using snapshots of flow data and a RNN data-compression approach.

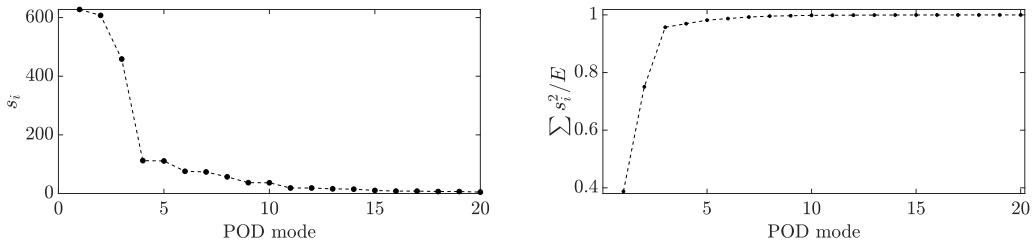
### Data Ensemble

Based on the aforementioned scale interactions the importance of the overall modelling of the flow, we set out to analyse 900 snapshots of the velocity field past a circular cylinder at Re = 60, one snapshot of which is shown in Figure 5.5. Data were obtained by direct numerical simulation of the NS equations using an in-house solver, as detailed in [97] Since we are interested in modelling the effect of the shift mode on the wake dynamics, data snapshots are chosen to be a time interval that includes the flow's transition from close to its unstable steady state to its saturation on a stable oscillatory limit cycle. The time step $\Delta t = 0.25s$ is constant for each consecutive pair of snapshots. For analysis, the time-averaged mean flow field was subtracted from each snapshot, to create an ensemble of velocity time-varying perturbations from the ensemble mean. For the remainder of this discussion, we will concentrate on this time-varying component of the velocity field.

For model training purposes, the first $m = 700$ of the available 900 snapshots were arranged in the data ensemble matrix $X \in \mathbb{R}^{p\times 700}$, where $p = 74094$. Each snapshot comprises the streamwise and the spanwise components of the velocity field, collected at 37864 spatial locations and interpolated onto a regular grid comprising of $p = 74094$

points in the flow domain. This segment of data used for training includes the transition to the limit cycle and two periods of steady oscillations of near-constant amplitude on the limit cycle. Singular value decomposition is then applied to $X$ to obtain a matrix of POD modes $\Phi \in \mathbb{R}^{p \times m}$, and these resulting modes are then used as the projection basis $\Phi$ with which to obtain time series coefficients using (5.10). The normalised time series can be seen in Figure 5.8 (in black), while the original time series can be seen in Figure 5.4.

Figure 5.6 (a) shows the singular values, $s_i$, of the 20 most dominant POD modes, and Figure 5.6 (b) shows the cumulative proportion of energy captured by a truncated set of POD modes against the number of the retained POD modes. Figure 5.6 (b) shows that the first 20 POD modes contain more than 99.9% of time-varying flow field's energy. For the remainder of this subsection, we use this truncated set of 20 POD modes and their associated time series as the projection bases and the high dimensional time-series vector $a(t)$. We note that if the same decomposition was performed using data drawn from the stable limit cycle only, then the cumulative energy captured by the global modes (i.e. the first two POD modes) would be significantly more dominant.




(a) (b)

Figure 5.6: (a) Individual and (b) cumulative energy of POD modes for flow past a circular cylinder at Re=60.

Having obtained $a(t)$, and applying the normalisation (5.14), we are now ready to apply the autoencoder to the normalised time series $\alpha(t)$. However, our choices of model parameters and hyperparameters should be further discussed.

**Construction of an autoencoder**

The optimisation problem (5.15) was solved using the ADAM variant of the gradient descent algorithm [91], and the learning rate was set at 0.025 (see §2.5.5 for the definition of learning rate).

Based on the results of the two-dimensional invariant manifold modelling approaches of [128] discussed in §5.3.1.1, we expect a two-dimensional encoded state $b(t)$ to be suitable for replicating the time series of POD modes. To minimise the number of parameters in the encoder function, the hidden layers $h_1$ and $h_2$ in (5.11) are chosen to have a dimension $\ell = 2$ as well. Since our aim is to reconstruct the POD time series of the first 20 modes, it follows that $\hat{\alpha}(t) \in \mathbb{R}^{20}$ and hence that the dimension of the hidden layers $h_3$ and $h_4$ in (5.12) are taken to be $\ell = 20$. The Lagrange multiplier $\eta$ was set to 0.004, the effects of varying $\eta$ will be presented later on in this section.

At any given time $t$, the encoder function requires the value of $a(t)$ at time steps $\{t, t - (25\Delta t), t - (50\Delta t), t - (75\Delta t), t - (100\Delta t)\}$. The choice of the separation between the input time-steps, comes from the period of the global modes' oscillations $T \approx 29\Delta t$ and the *look-back* time of $t - 100\Delta t$ was chosen to be of the same time scale as that of the dominant variations for the shift mode.

To illustrate how these parameters allow the input sequence to contain information about both long-term and short-term variations in the flow, Figure 5.7 shows the temporal coefficient $\alpha_1(t)$ and the input sequence corresponding to snapshots 673 (blue) and 525 (red). Choosing a slightly smaller separation than the period of oscillations, allows the the input sequence to span phase and amplitude values covering one period of oscillations over the limit cycle. Comparing the input sequence at $t = 673\Delta t$ with that at $t = 525\Delta t$, we notice a shift in the phase of the points (e.g. compare the phase at $425\Delta t$ and $573\Delta t$), and changes in amplitude ratios both due to low-frequency changes in the underlying data. This indicates that the look-back parameter is suitable for capturing information about the low-frequency events as well. Considering that the Elman model is not suitable for

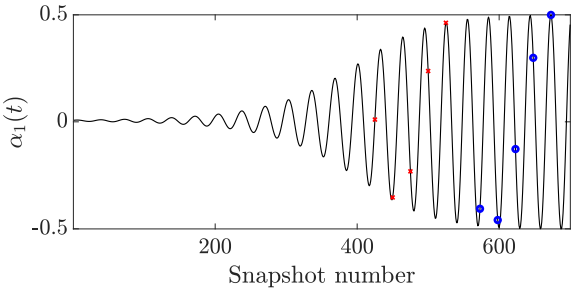


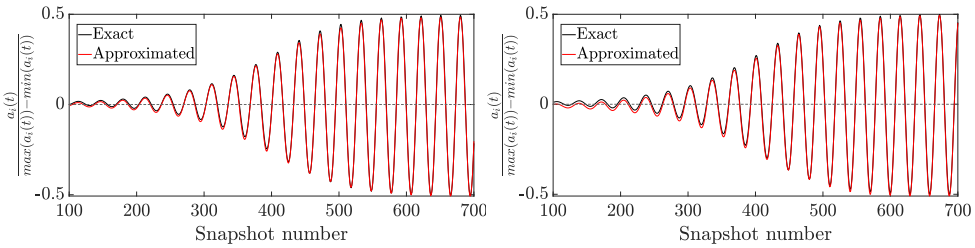

Figure 5.7: Two examples of input sequences (red crosses, blue markers) which can be passed to the constructed autoencoder.

longer input sequences, these parameters convey both low and high-frequency information through a short input sequence of length 5.

The decoder function, however, is required to find the immediate relationship between the encoded vector $b(t)$ and the normalised POD time series $\alpha(t)$. Therefore, the input to the decoder function is only $b(t)$ and does not include any previous time steps.

#### 5.3.1.2 Autoencoder Performance (Cylinder Wake)

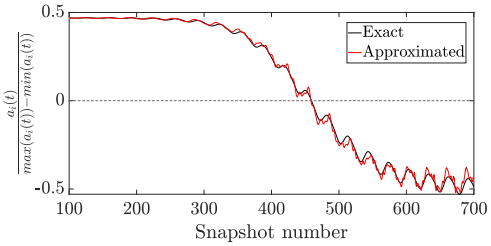
Optimising the model with the above parameters, and then applying the optimal model to the POD time series $\alpha(t)$ we find the approximated time series $\hat{\alpha}(t)$. Figure 5.8 shows $\hat{\alpha}(t)$ (in red) for the first ten POD time series in alphabetical order. The time series start at the $100^{\text{th}}$ snapshot as the encoder function was initialised with five equidistant snapshots in the interval $0 \leq t \leq 100\Delta t$. The first two time series, (a) and (b), are associated with the vortex shedding modes and the third, (c), is the mean-flow's shift mode. The subsequent figures show the time series for the modes representing the first harmonic of the global mode, (d) and (e), the *vortex shedding shift mode* [156], (f) and (g), and the second harmonic of the global mode, (i) and (j). We consider the mode in (h) to be related to the dynamics of the flow's transition to its stable limit cycle.



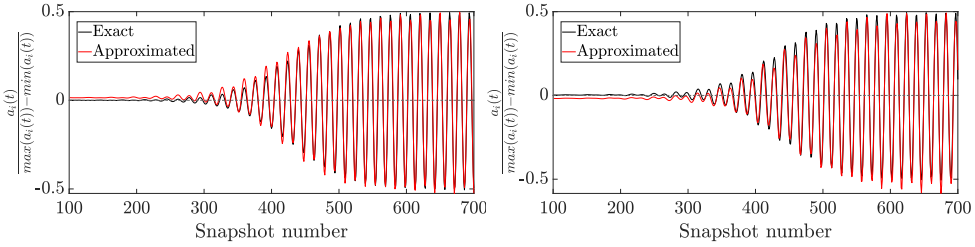

(a) error = 4.4%




(b) error = 5.6%

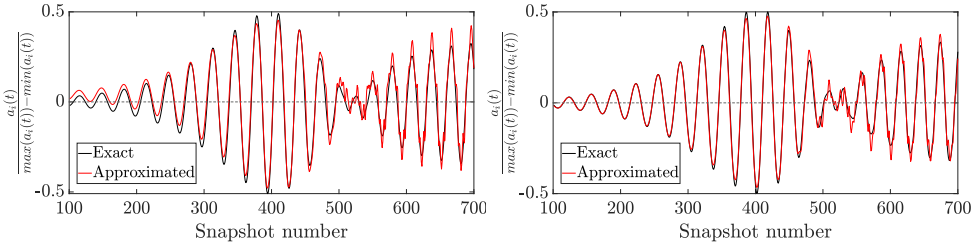



(c) error = 4.2%



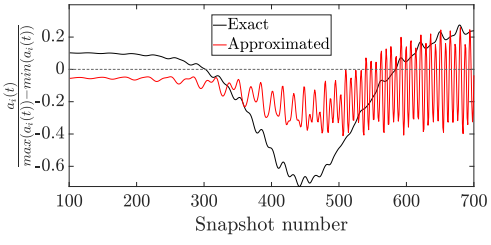

(d) error = 12.1%

(e) error = 14.2%

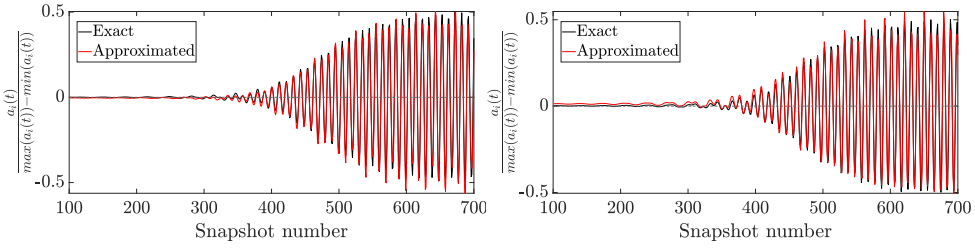



(f) error = 20.6%

(g) error = 12.9%




(h) error = 78%

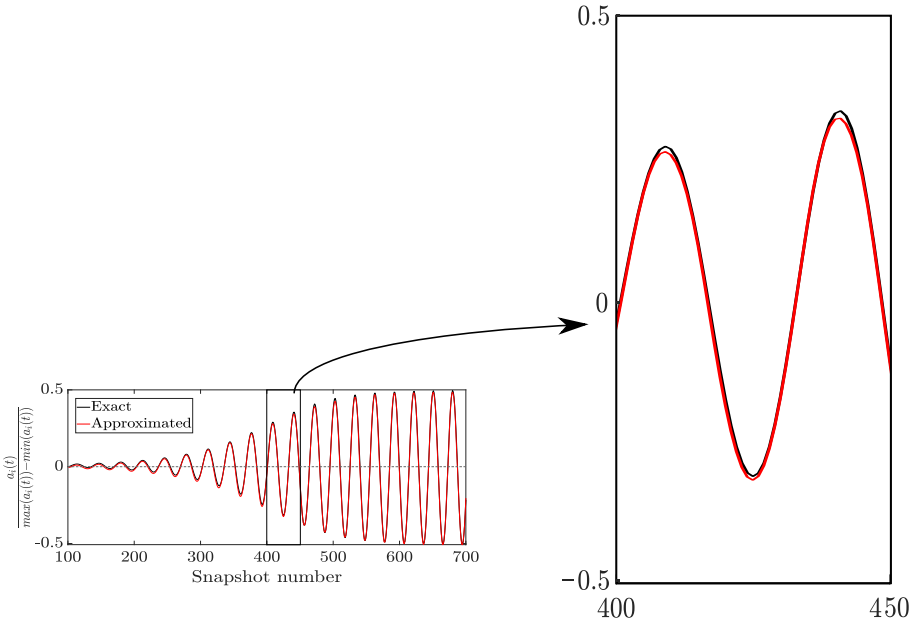



(i) error = 20.1%

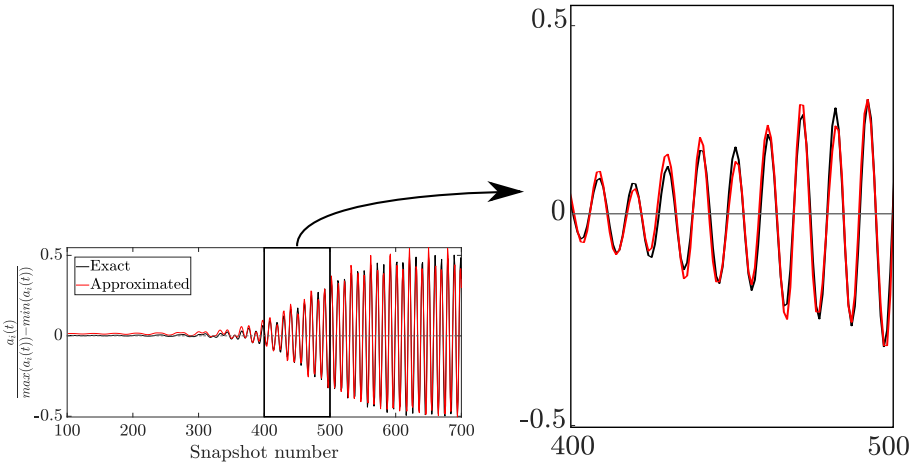
(j) error = 18.6%

Figure 5.8: Exact (black) and approximated (red) normalised POD time series.

For legibility, Figure 5.9 shows a closer look at a few snapshots of the two time series $\alpha_1$ and $\alpha_{10}$.




(a) Vortex shedding mode $\alpha_1$




(b) Second harmonic of the vortex shedding mode $\alpha_{10}$

Figure 5.9: The reconstructed time series closely follow the input time series

In order to provide a comparison with a more standard model reduction approach, we simply make the trivial observation that truncation of the POD basis can also be expressed as an autoencoder. In particular, if the flow field is simply projected onto the first two POD modes, this corresponds to an encoder mapping

$$b(t_j) = \begin{pmatrix} \mathbf{I}_2 & 0 \\ 0 & 0 \end{pmatrix} a(t_j), \qquad j = 1, \ldots, m,$$

where $\mathbf{I}_2 \in \mathbb{R}^{2\times 2}$ is the identity matrix mapping the full vector of POD weights onto only its first two components. The decoding step is then simply

$$\hat{a}(t_j) = \mathbf{I}_{20} b(t_j), \qquad j = 1, \ldots, m,$$

where $I_{20} \in \mathbb{R}^{20\times 20}$ is also an identity matrix. By construction, such an encoding is the energy-optimal representation of the underlying data in terms of the *linear* span of two spatial mode shapes.

Figure 5.10 shows the percentage error

$$E(\alpha_i) := \frac{\sum_{j=1}^{m} \|\alpha_i(t_j) - \hat{\alpha}_i(t_j)\|_2^2}{\sum_{j=1}^{m} \|\alpha_i(t_j)\|_2^2} \times 100\%$$

for all $n = 20$ time series $\alpha_i(t)$ (in blue). Also included are the error bars for the trivial two degree of freedom truncation autoencoder, which captures the exact solution for the first two time series but has error $E(\hat{\alpha}) = 100\%$ for the remaining 18 POD weights.

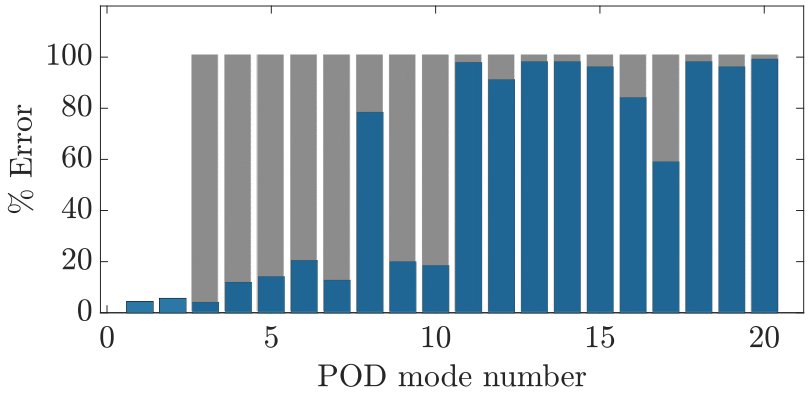



Figure 5.10: Reconstruction error of normalised POD time series using truncation (grey), and using encoding (blue)

We now discuss the performance of the nonlinear autoencoder. Similar to the truncation case, the nonlinear autoencoder also prioritises representing the two global modes

($\phi_1$ and $\phi_2$), perhaps unsurprisingly since these modes have the largest energetic contribution to the underlying ensemble of velocity fluctuations. However, recalling that (5.15) balances prioritising the reconstruction error of the velocity fluctuations with the error associated to reconstructing the *normalised* time series, it is interesting to note a clear deviation in performance from the trivial truncation autoencoder.

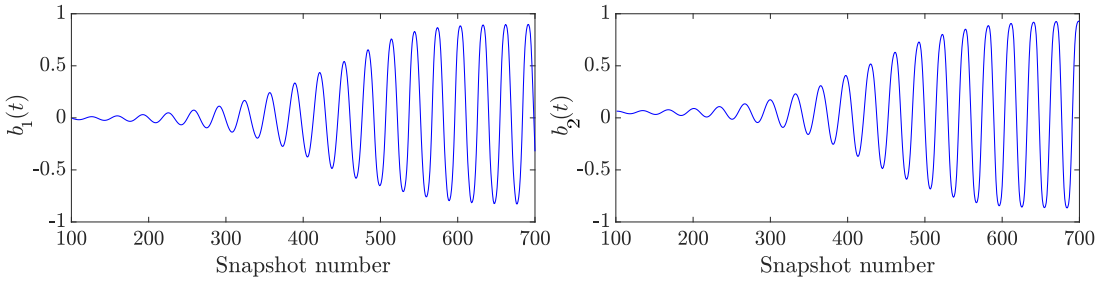
In particular, the nonlinear autoencoder (with two underlying degrees of freedom) is also able to capture both the shift mode $\phi_3$ and the first harmonic of the global mode $\phi_4, \phi_5$ with reasonably low reconstruction error (of $4.2\%, 12.1\%$ and $14.2\%$, respectively). Although these errors are non-trivial, an inspection of the three time series shown in Figure 5.8 (c), (d) and (e) indicates that the time series are in fact very well captured, with the error accumulating merely due to small phase and amplitude variations over the chosen simulation interval.

In addition, the nonlinear autoencoder is also able to capture the second harmonic of the global modes $\phi_9$ and $\phi_{10}$ with less than $21\%$ reconstruction error. Interestingly, this is achieved at a lower reconstruction error than the more energetic mode $\phi_8$ which we consider to be related to the dynamics of the transition of the flow from an unstable steady solution to its stable limit cycle (again, such a mode would not be expected to have such a high singular value if data were sampled from the limit cycle regime only). The fact that modes $\phi_9, \phi_{10}$ are captured at lower reconstruction error than $\phi_8$ suggests that a more coherent, even if nonlinear, mapping can be found between them and the identified latent states $b$, which are shown in Figure 5.11 and discussed below. As discussed further in §5.3.2, this is not entirely surprising: due to energetic significance the nonlinear autoencoder still prioritises capturing the global modes $\phi_1, \phi_2$, suggesting that the latent state will be transparently related to their time series $a_1, a_2$. As a consequence, modes which have a direct dynamic relation to these time series—in particular, their harmonics—have a greater chance of being reconstructed, with smaller errors, by the nonlinear autoencoder.

Finally, we note that the mode shape and time series of the modes $\phi_6$ and $\phi_7$, whose time series are captured with an error of $E(\hat{\alpha}) = 21\%$, strongly resemble those of the

modes extracted in [156] using the DPOD algorithm, that model a spatial shape change in the vortex shedding feature. In particular, the modes model the movement of higher vortex-shedding amplitudes downstream of the flow after transitioning to the limit cycle.

In the discussion at the start of this subsection we noted that in the existing literature, the temporal evolution of the shift mode and global mode are known to be related by a simple nonlinear ODE. Here, we find that the encoded time series $b_1(t_j)$ and $b_2(t_j)$ strongly resemble the evolution of the first two POD modes, allowing for phase shifts and scaling. Figure 5.11 shows $b_1$ and $b_2$. Similar to the vortex shedding time series, the encoded signals exhibit an oscillatory behaviour with a dominant frequency of $St = 0.14$ over the limit cycle, and include two full periods of oscillations on the limit cycle for the training set. Unlike the more traditional modelling approaches however, such as truncation, the nonlinear dependencies among different physical features, as well as the corresponding mapping functions were obtained automatically through the process of model discovery.




(a) (b)

Figure 5.11: Time series of the encoded states $b_1, b_2$, indicating similarity vortex shedding mode amplitude times series.

To ensure that the good performance of the autoencoder is not due to overfitting to the training dataset, we apply the autoencoder on a test dataset. The test dataset includes 100 snapshots of the velocity field, snapshots $801 \leq t_j \leq 900$, with snapshots $701 \leq t_j \leq 801$ used to initialise the encoder. Table 5.1 shows the error

$$E(X) = \frac{\|X - \sum_{i=1}^{20} \phi_i \hat{a}_i(t)\|_F}{\|X\|_F}$$

in reconstructing the time-varying velocity snapshots, $X$, over both training and testing subsets. The autoencoder method allows for a considerably higher percentage of the flow to be replicated, through a decoding process, than a truncated state of the same dimension.

We note again that the underlying dataset mostly contains snapshots where the data is transitioning to its oscillatory limit cycle and transient features are significantly more energetic than they are over the limit cycle. In particular, the shift mode contains 20.6% of the flow's energy.

| | Truncation $r = 2$ | Autoencoder (train) | Autoencoder (test) |
|---|---|---|---|
| $E(X)$ | 25% | 8.0% | 8.5% |

Table 5.1: Reconstruction error of time varying velocity.

#### 5.3.1.3 The effects of varying the parameter $\eta$ in (5.15)

By choosing an appropriate value of $\eta$, we encourage the autoencoder to replicate as many time series as possible while also considering the energetic contribution. As a result, we allow the optimiser to note the strong dynamical relationship between the second harmonic modes and the most energetic modes, by noting that a common encoded state $b(t)$ can be mapped well to both dynamical features. A low value of $\eta$ results in assigning the same importance to the reconstruction of each time series. To illustrate this point, Figure 5.12 shows the values of $E(\alpha_i)$ for an autoencoder optimised at $\eta = 0$ (in blue). In comparison to the case when $\eta = 0.004$, not only do the highly energetic vortex shedding modes have a high $E(\alpha_i)$, but also fewer modes have a reconstruction error $E(\alpha_i) < 25\%$. This is due to fewer dynamical relationships between the low energy modes leading to an encoded state that can not map well to many states at the same time.

Figure 5.12 also shows the values of $E(\alpha_i)$ for a model optimised at $\eta = 100$ (in red). In this case the modes with the highest energetic contribution are even slightly better captured than the model at $\eta = 0.004$ (see Figure 5.10), with $E(\alpha_1) = 3.9\%$ and

$E(\alpha_2) = 2.4\%$. However, the high emphasis on capturing the more energetic features, leads to ignoring many low-energy features that the model at $\eta = 0.004$ captures.

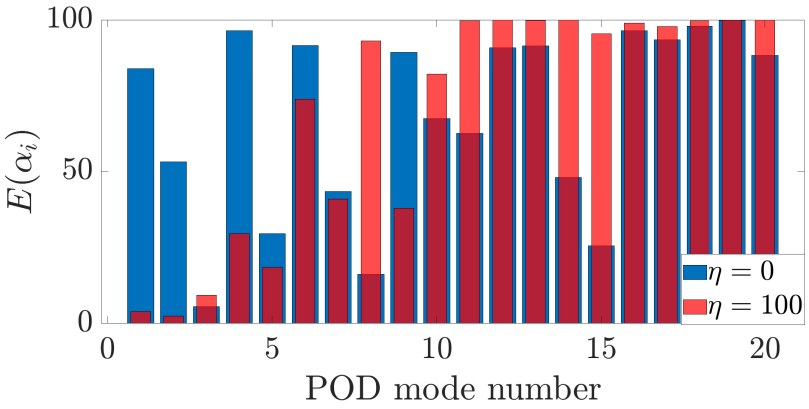



Figure 5.12: Reconstruction error $E(\alpha_i)$ against POD mode coefficient $\alpha_i$ of the normalised POD time series at $\eta = 0$ and $\eta = 100$.

In comparison to the existing machine learning based feature identification methods in the literature, while the weights and constants inside the encoder and decoder functions remain uninterpreted, the encoded state is by design interpreted through a known projection step. As a result, the autoencoder does not solely concentrate on replicating the dataset, but also on replicating interpretable features of the dataset. These interpretable features, were themselves obtained directly from the dataset and although they are in agreement with the known significant features of this flow [157], they were not designed through expert knowledge.

### 5.3.2 Autoencoder performance and observability

We now briefly discuss a possible explanation for why *nonlinearity* of the underlying mappings in the autoencoder leads to success in capturing the behaviour of multiple modal time series using only two underlying degrees of freedom. To aid the discussion, we recall the classical notion of *observability* of linear systems. Consider a linear system

$$\dot{a}(t) = \mathcal{A}a(t), \qquad y(t) = \mathcal{C}a(t)$$

with state $a(t) \in \mathbb{R}^n$, system matrix $\mathcal{A} \in \mathbb{R}^{n\times n}$ and measured output $y \in \mathbb{R}^r$ where $1 \leq r \leq n$ is the dimension of the measurement signal. The system is said to be (linearly) *observable* if the observability matrix $\mathcal{O}$ has full rank, that is if

$$\operatorname{rank}(\mathcal{O}) = \operatorname{rank}\begin{pmatrix} \mathcal{C} \\ \mathcal{C}\mathcal{A} \\ \vdots \\ \mathcal{C}\mathcal{A}^{n-1} \end{pmatrix} = n.$$

Observable systems have the desirable property that given knowledge of only the output $y(t)$, the full state can be asymptotically recovered using a linear observer [108].

For nonlinear systems, there is a natural extension of the concept of observability. In this case, consider a system of the form

$$\dot{a}(t) = f(a(t)), \qquad y(t) = g(a(t)) \tag{5.17}$$

where the state and measurement processes are now given by arbitrary nonlinear mappings $f : \mathbb{R}^n \mapsto \mathbb{R}^n$ and $g : \mathbb{R}^n \mapsto \mathbb{R}^r$. Extending the definition of observability to the nonlinear setting requires the Lie derivative notation, specifically for multivariable functions $f, g$ as above the Lie derivative of $g$ with respect to $f$ is defined by $L_f g := (\nabla g)f$. We discuss the case with output dimension $r = 1$ only for simplicity. The nonlinear system (5.17) is said to be locally observable at a point $x \in \mathbb{R}^n$ if the following rank condition on the nonlinear observability matrix $\mathcal{O}_{\mathrm{NL}}(x)$ holds

$$\operatorname{rank}\left(\mathcal{O}_{\mathrm{NL}}(x)\right) := \operatorname{rank}\begin{pmatrix} (\nabla g(x))^\top \\ (\nabla L_f g(x))^\top \\ \vdots \\ \left(\nabla L_f^{n-1} g(x)\right)^\top \end{pmatrix} = n$$

It is not difficult to show that $\mathcal{O}_{\mathrm{NL}}(x) = \mathcal{O}$ for any $x \in \mathbb{R}^n$ when $f = \mathcal{A}, g = \mathcal{C}$, meaning that the two definitions coincide in the case of linear systems. In the nonlinear setting, local observability implies that the full state $a(t)$ can be asymptotically identified by an appropriate dynamic observer [77]

We now discuss the performance of the nonlinear autoencoder for the cylinder wake modes in the context of observability. Although it is not clear that autoencoders and observers are directly equivalent, they do have some notable similarities. First, the fact that the encoded state can depend, through recursion of the multiple layers in an RNN (see (5.5)), on past history of the latent states endows the encoder function with a structure reminiscent of the dynamical system underpinning a linear or nonlinear observer. Second, the dimension $r$ of the latent states $b \in \mathbb{R}^r$ can be viewed as analogous to the dimension of the output mapping $y(t) \in \mathbb{R}^r$ in so far as it reflects the degree of information available with which to reconstruct the full state of the system.

Returning to the cylinder wake specifically, recall from §5.3.1.1 the simple 3-state model proposed by [128] for the dynamic interaction of the shift mode and the global modes, $\dot{a}(t) = \mathcal{A}a + N(a)a$ where

$$\mathcal{A} = \begin{pmatrix} \mu & -1 & 0 \\ 1 & \mu & 0 \\ 0 & 0 & -1 \end{pmatrix}, \qquad N(a) = \begin{pmatrix} 0 & 0 & -a_1 \\ 0 & 0 & -a_2 \\ a_1 & a_2 & 0 \end{pmatrix}.$$

Consider first the case of the linearised dynamics $\dot{a} = \mathcal{A}a$, in which $N(a) \equiv 0$. Suppose that we attempt to recreate the state of the system from a measurement of the form

$$y(t) = \mathcal{C}a = \begin{pmatrix} c_1 & c_2 & 0 \end{pmatrix} a,$$

reflecting the situation of an autoencoder with two latent states $b_1, b_2$ which are given as a linear combination of the global mode's POD time series coefficients $a_1$ and $a_2$. It is easy to compute the linear observability matrix

$$\mathcal{O} = \begin{pmatrix} c_1 & c_2 & 0 \\ \mu c_1 + c_2 & \mu c_2 - c_1 & 0 \\ c_1(\mu^2 - 1) + 2\mu c_2 & c_2(\mu^2 - 1) - 2\mu c_1 & 0 \end{pmatrix}.$$

It is clear that $\text{rank}(\mathcal{O}) = 2 < 3 = n$ meaning that the system is not linearly observable from any measurement which is a linear combination of the two global mode amplitudes. Conversely, and for ease of computation, we now calculate the nonlinear observability matrix in the case when only one of the global mode amplitudes can be measured. Suppose that $g(a(t)) = a_1(t) = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} a(t)$. Then letting

$$f(a) = \mathcal{A}a + N(a)a = \begin{pmatrix} f_1(a) \\ f_2(a) \\ f_3(a) \end{pmatrix} = \begin{pmatrix} \mu a_1 - a_2 - a_1 a_3 \\ a_1 + \mu a_2 - a_2 a_3 \\ -a_3 + a_1^2 + a_2^2 \end{pmatrix}$$

it follows that $L_f g(a) = \mu a_1 - a_2 - a_1 a_3$ and $L_f^2 g(a) = (\mu - a_3) f_1 - f_2 - a_1 f_3$. Hence,

$$\nabla L_f g(a) = \begin{pmatrix} \mu - a_3 \\ -1 \\ -a_1 \end{pmatrix}$$

and

$$\nabla L_f^2 g(a) = \begin{pmatrix} (\mu - a_3)\frac{\partial f_1}{\partial a_3} - \frac{\partial f_2}{\partial a_1} - f_3 - a_1 \frac{\partial f_3}{\partial a_1} \\ (\mu - a_3)\frac{\partial f_1}{\partial a_2} - \frac{\partial f_2}{\partial a_2} - a_1 \frac{\partial f_3}{\partial a_2} \\ (\mu - a_3)\frac{\partial f_1}{\partial a_3} - f_1 - \frac{\partial f_2}{\partial a_3} - a_1 \frac{\partial f_3}{\partial a_3} \end{pmatrix} = \begin{pmatrix} (\mu - a_3)^2 - (1 + 2a_1^2) - (a_1^2 + a_2^2 - a_3) \\ -2(\mu - a_3) - 2a_1 a_2 \\ 2a_1(\mu - a_3) + a_1 + 2a_2 \end{pmatrix}.$$

and, consequently,

$$\mathcal{O}_{\mathrm{NL}}(a)$$
$$= \begin{pmatrix} 1 & 0 & 0 \\ \mu - a_3 & -1 & -a_1 \\ (\mu - a_3)^2 - (1 + 2a_1^2) - (a_1^2 + a_2^2 - a_3) & -2(\mu - a_3) - 2a_1 a_2 & 2a_1(\mu - a_3) + a_1 + 2a_2 \end{pmatrix}.$$

Note that $\mathcal{O}_{NL}(0)$ coincides with the rank-deficient observability matrix computed in the linear case. However, for $a \neq 0$, the extra terms introduced by the nonlinear couplings in $N(a)a$ may imply that the nonlinear observability condition $\mathrm{rank}(\mathcal{O})_{\mathrm{NL}} = 3$ holds. We do not verify this for the general case, but note that on the limit cycle when $a_3 = \mu$ and, up to phase shifts, $a_1 = \sqrt{\mu}\sin t, a_2 = \sqrt{\mu}\cos t$, the nonlinear observability matrix reduces to

$$\mathcal{O}_{\mathrm{NL}}(a(t)) = \begin{pmatrix} (\nabla g(a))^\top \\ (\nabla L_f g(a))^\top \\ (\nabla L_f^2 g(a))^\top \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & -a_1 \\ -(1 + 2a_1^2) & -2a_1 a_2 & a_1 + 2a_2 \end{pmatrix}$$
$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & -\sqrt{\mu}\sin t \\ -(1 + \mu - \mu\cos(2t)) & -\mu\sin(2t) & \sqrt{\mu}(\sin(t) + 2\cos(t)) \end{pmatrix}$$

It follows that for any point on the limit cycle trajectory, the nonlinear observability condition $\mathrm{rank}(\mathcal{O}_{\mathrm{NL}}) = 3$ holds. Therefore, somewhat surprisingly given that the shift mode does not vary on the limit cycle, it is theoretically possible to asymptotically identify the full state of the nonlinear 3-mode system, given only knowledge of the first state $a_1$.

A consequence of this small observation from the theory of nonlinear observers hints at why the nonlinear autoencoder constructed in §5.3.1.2 may succeed: if the internal structure of the autoencoder can reflect that of the theoretically possible nonlinear observer, then information about the shift mode $\phi_3$ should be reconstructible from global mode information only.

A similar argument may be made with respect to the reconstruction of higher harmonics of the global mode. In a linear framework, the sequence of harmonics of the global mode can be represented as an infinite superposition of independent 2-dimensional oscillators. A similar analysis to that conducted above using linear observability theory would then imply that higher harmonics should not be observable, given measurements which contain only amplitude information of lower harmonics. Translating this to a hypothesis about autoencoder performance, would imply that a linear autoencoder whose latent states resemble those of the global mode amplitudes should not be able to accurately reconstruct the amplitude time series of higher harmonics. However, due to nonlinear interactions between the global mode, its harmonics and the shift mode, we expect that nonlinear observation theory implies (by a similar argument to that presented above) the identification of higher harmonic amplitudes should be possible within a nonlinear observer framework. The fact that the nonlinear autoenoder of §5.3.1.2 is able to accurately represent some higher harmonic information partially supports this arguement. That the reconstruction error is not zero, however, suggests that the 2-layer RNN framework chosen cannot match the theoretical optimal performance of potential nonlinear observers. This may either be due to the lack of complexity available by the given architecture of the relatively simple RNNs chosen, or due to the fact that there is no guarantee that globally optimal parameter values have been identified due to the local, gradient-based, nature of the autoencoder training.

The application of autoencoders to the cylinder wake highlights the potential of the presented approach for finding interpretable encodings of a fluid flow, as well as the sense in which the encoding can be interpreted through a mode-shape layer. As in §3 and §4, we test the applicability of this method on the more challenging dataset from a highly turbulent wake, where due to the problem of frequency mixing, the POD modes are not interpretable as being associated with a single physical phenomenon. Furthermore, due to the broadband and non-linear nature of the underlying flow, representing a physical phenomenon with a single mode is challenging.

### 5.3.3 Dimensionality reduction for the wake of an axisymmetric bluff body

In §3 we studied a diametric cross-section of the 3-dimensional 'bullet shaped' bluff body at $Re = 1.88 \times 10^5$. For this flow, there are three coherent structures which have been observed experimentally. The bubble pumping mode, associated with the low-frequency elongation of the recirculation region at $St \approx 0.05$ [20], the vortex shedding mode at $St \approx 0.2$, and a chaotic change to the angle of rotation at a much lower frequency $St \approx 0.002$ [142].

In §3 we applied OMD to an ensemble of 2732 PIV flow field snapshots and the resulting OMD modes were post-processed using Algorithm 1, as shown in Figure 3.22. The sequential snapshots were chosen in a way that the azimuthal orientation of the plane of symmetry, as estimated using the method in [25], is parallel to the plane of view with a variance of $\pi/5$ rad.

For the current discussion, we assume the direction of the flow field is constant, and as a result, the effects of the azimuthal rotations on the current dataset are considered negligible. However, the out-of-plane movement of the wake introduces a degree of epistemic uncertainty to this dataset.

The clusters of similar modes aid our understanding of the coherent structures available in the system. However, by construction, the modes in each cluster also exhibit high multicollinearity. That is, by definition the spatial modes in a cluster span a similar subspace, as defined in (3.6). After applying the clustering method in §3, with spatial similarity cut-off $\epsilon = 0.5$, and spectral similarity parameters $s_0 = 0.75$ and $p = 2$, we obtain two significant clusters of modes with cluster-averaged Strouhal numbers $\langle St \rangle = 0.06$ and $\langle St \rangle = 0.23$, respectively.

Suppose now that we want to find a reduced order state to model the dynamical behaviour of a mode. For reconstruction purposes, the existence of collinear features in the basis matrix $\Phi$ is not important. However, the time series of individual modes will not

be interpretable, as a linear least square fit does not consider the evolution of individual mode-shapes and only minimises the overall reconstruction error. To address this, in §3 we obtained a representative mode for each cluster by applying Singular Value Decomposition to the rescaled modes in the cluster, and using the first singular vector (POD mode) as the representative mode. Such representative modes are therefore appropriately weighted averages of the modes inside a given cluster. While these modes are useful for flow analysis and visualisation, for a broadband and nonlinear system, a small number of representative modes may span only a 'small' proportion of the underlying data. That can be the case even if the modes represent dominant clusters of modes.

Figure 5.13 shows the representative modes for the two significant clusters of this flow. Using these two modes as the projection bases we find the projection error

$$\frac{\left\|(X-\bar{X})-\Phi\left(\Phi^{\top}\Phi\right)^{-1}\Phi^{\top}(X-\bar{X})\right\|_F^2}{\left\|(X-\bar{X})\right\|_F^2}=0.09027,$$

where $\bar{X}$ is the time-averaged velocity field. In other words, we find that the four representative mode-shapes only span 9.7% of the time-varying snapshots.

The two clusters contain a total of 32 complex modes, whose real and imaginary parts provide 64 real-valued velocity fields. Using this full set of mode shapes, we can recreate 29.3% of the time-varying velocity field. Although still small, this should be viewed in the context of an original decomposition comprised of 2000 complex-valued modes. Figure 5.14 shows a snapshot of the streamwise time-varying flow field in panel (a), its projection on the full set of mode shapes in the two dominant clusters in panel (b), and the same snapshot projected onto only the two representative modes in (c). The prevalence of high amplitude, incoherent small-scale features in the full snapshot (a), explains why the projection of the snapshot on these relatively large-scale coherent structures does not capture a majority of the flow field's energy.

Comparing the projected snapshots in (b) and (c) with the full snapshot (a), note that a convective structure of negative streamwise fluctuations centred at approximately

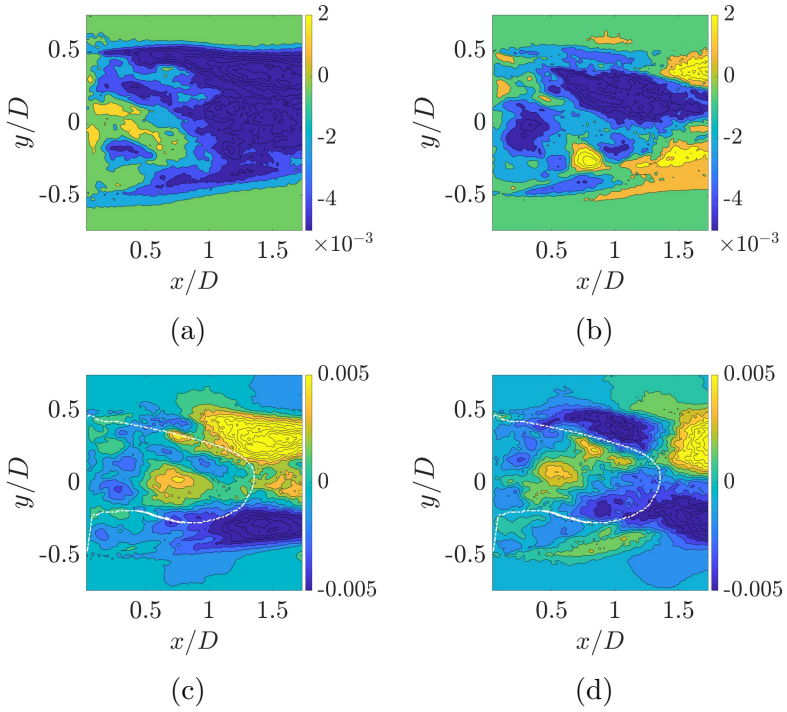
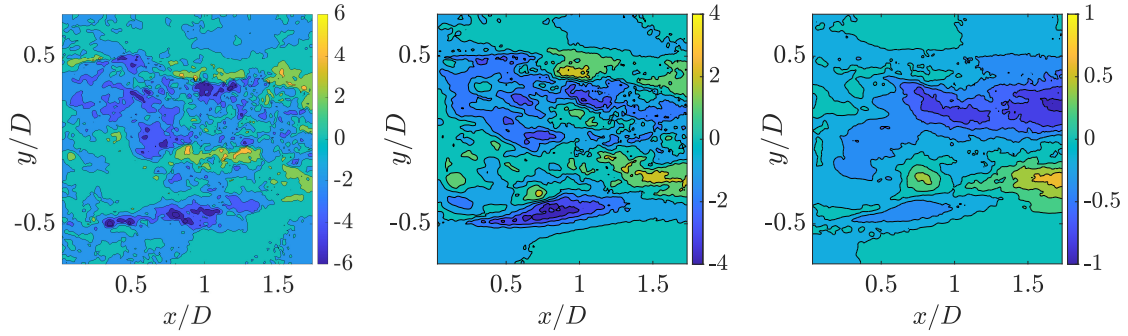



Figure 5.13: Real and imaginary components of representative modes: (a-b) bubble-pumping mode at $\langle St \rangle \approx 0.06$; (c-d) shedding mode at $\langle St \rangle = 0.23$.

$(x/D, y/D) = (0.75, -0.4)$ is captured in both projections (b) and (c). We find the same feature in snapshot (a), albeit masked significantly by smaller-scale features that are present in the vicinity of this location. Similarly, both projections in (b) and (c) show convective features for $x/D > 1.5$. The existence of multiple convective features in projection (b) allows it to accurately capture two convective features of opposite signs in the region $x/D > 1$, $y/D > 0$ (i.e. both positive and negative velocity perturbations are present in this region). However, if only two vortex-shedding mode-shapes are used in the projection bases, only a single large-scale structure with negative fluctuations (i.e. the one centred at $(x/D = 1.75, y/D = 0.25)$ in (c)) is present in that region, meaning that this nuance of the wake's behaviour cannot be captured. Overall, projection (b) resembles the near-wake's shape much more closely, with a correlation of 0.61, compared to projection (c), with a correlation of 0.16.

We would like to reiterate that this does not discredit the importance of the representative modes in illustrating the spatial and spectral features included in the cluster.




(a) (b) (c)

Figure 5.14: (a) a snapshot of the time varying velocity field; (b) its projection using all the modes in the clusters; and (c) its projection using only the representative modes.

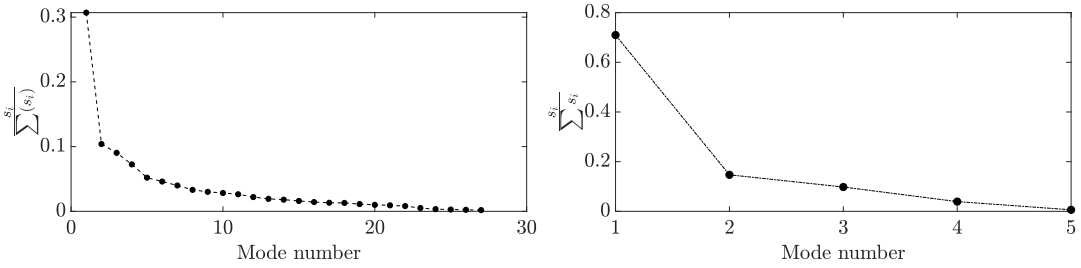
However, the example above shows that we can not expect one mode shape to fully capture a quasi-periodic physical phenomenon in a turbulent flow.

The discrepancy between the two projections can be better explained by the relatively broad range of spatial features present in each cluster. To illustrate this point we need to revisit how representative modes were obtained. Recall that in §3.2.3, the modes in a cluster were rescaled using the significance statistic $\sigma$, and the representative mode was found by applying an SVD to the weighted mode ensemble

$$\mathcal{M}_{\text{shedding}} := \begin{pmatrix} & \uparrow & \\ \cdots & \phi_i\sigma_i & \cdots \\ & \downarrow & \end{pmatrix}, \quad \text{for } i \text{ such that } \phi_i \in C_{\text{shedding}}.$$

In §3 the first SVD mode of $\mathcal{M}$ was chosen as the representative mode of the cluster. Figure 5.15 shows the singular values of the matrix $\mathcal{M}_{\text{shedding}}$ associated with the vortex shedding feature. While the representative mode is clearly dominant and spans 30% of the subspace covered by all the features in this cluster, to span 90% of the mode ensemble $\mathcal{M}_{\text{shedding}}$, 15 representative POD modes would have to be used. A similar analysis shows that to represent 90% of the bubble pumping mode ensemble, requires 3 representative modes. The 90% threshold is within the range of values used as arbitrary cut-offs for

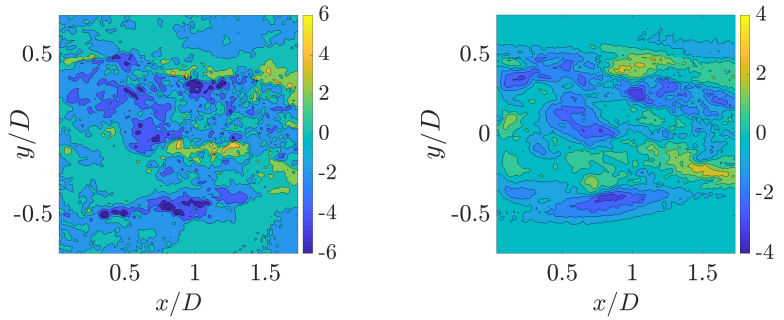
POD-based dimensionality reduction although alternative methods have been developed in recent years (see [26] and the references therein). This is a generalisation of the analysis in §3, and for the remainder of this chapter, the set of SVD modes that is retained from each cluster is referred to as the representative modes or the representative set.




(a) (b)

Figure 5.15: Singular values corresponding to an SVD of all modes associated with the vortex shedding cluster.

Using the 15 vortex shedding and 3 bubble-pumping representative modes, we find the projection in Figure 5.16 (b), which shows a significant correlation with the time-varying snapshot in Figure 5.16 (a), at 0.53. Note that due to the orthonormality of the POD modes, using them as the projection bases prevents multicollinearity. However, an ensemble of 18 representative modes implies there will be 36 real-valued time series $a_i(t)$. For practical use in control and estimation applications, further dimensionality reduction will be necessary. We will use the method introduced in §5.2.2 to further reduce the dimension of the full projected state $a(t)$.




(a) Snapshot (b) Projection

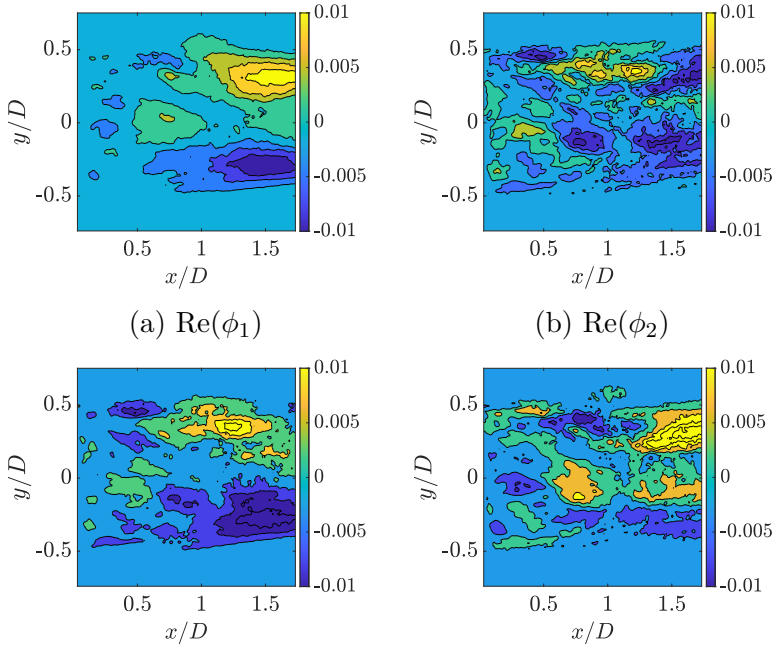
Figure 5.16: (a) Snapshot of the time varying velocity field; (b) its projection using 18 representative modes.

First, we must present some of the representative modes and interpret the spatial features in each mode. We have already shown the first representative modes of each cluster in Figure 5.13. Expectedly, the first representative mode of the vortex shedding cluster includes large-scale, antisymmetric, advective structures in the region outside of the average position of the recirculation bubble at $x/D > 1.5$, see Figure 5.13 (c) and (d). In contrast, in the second representative shedding mode, Figure 5.17 (b), the flow field is approximately symmetric about the centreline $y = 0$ for $x/D > 1.5$.

To illustrate the effect of the inclusion of the second POD mode from the vortex shedding cluster, we superimpose the real parts of the first and second representative modes and find the resulting flow field, shown in Figure 5.17 (c). Comparing Figure 5.17 (c) and Figure 5.17 (a), the location of the peak of the positive fluctuations has shifted upstream to $x/D \approx 1.25$, creating a spatial 'lag' effect between the asymmetric fluctuation features. Moreover, the negative structural perturbation at $(0.75, 0.4)$ has shifted to $(0.5, 0.5)$, closer to the separation point $(0, 0.5)$. Furthermore, the outline of the negative fluctuations structure in the region $x/D > 1.25$ is also shifted towards the centreline $y/D = 0$. In contrast, Figure 5.17 (d) shows that subtracting the two mode shapes results in a downstream shift of the peak, compared to (a), as well as moving the outline of the negative structure further away from the centreline.

The example above illustrates that linear combinations of the representative mode-shapes allow the model to create advective structures of different shapes, at varying spanwise locations and with varying levels of asymmetry. The average length scale and shape of the feature are, however, captured by the first representative mode. We note that while such variations are observed in this high Reynolds number wake, they are not observed for the low Reynolds number cylinder wake.

The mean bubble pumping mode seen in Figure 5.13 (a) and (b) captures the large-scale elongation and contraction of the recirculation region at $St \approx 0.06$. Figure 5.18 shows the real and imaginary part of the second representative mode of the bubble pumping cluster. The evolution of the large-scale diagonal structure of negative fluctuations centred

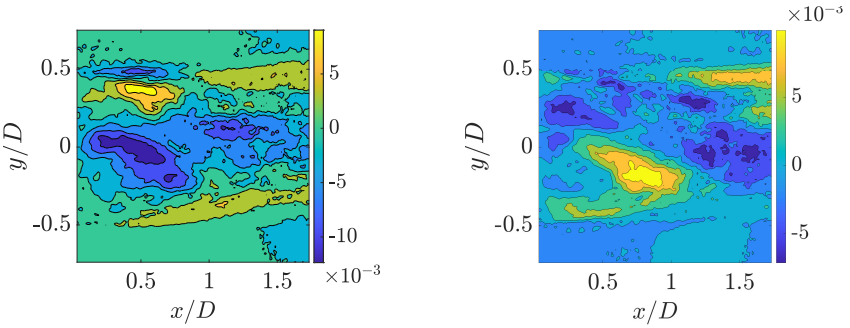



(a) Re($\phi_1$) (b) Re($\phi_2$)

(c) Adding mode shapes (d) Subtracting mode shapes

Figure 5.17: The effect of combining mode shapes from a cluster: (a-b) real parts of the first and second representative modes from the vortex shedding cluster; (c) the sum of the mode shapes; (d) the difference of the mode shapes.

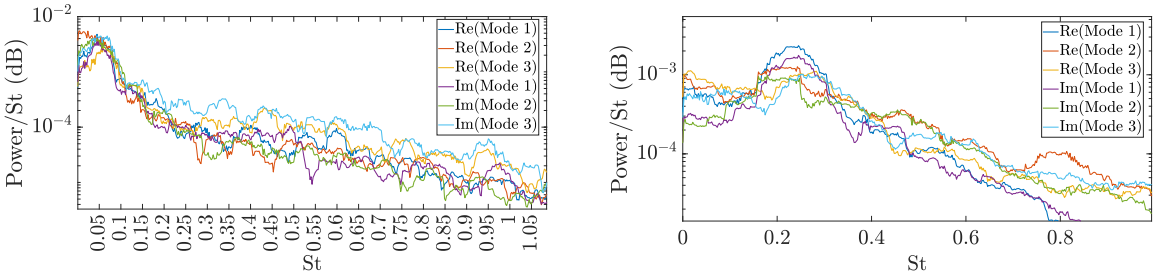
at $(0.75, -0.25)$ in Figure 5.18 (a), to that in (b) at $(0.75, 0.25)$, captures the movement of the recirculation region in the $y$ direction. This is consistent with the findings of [34] at lower Reynolds numbers that the shape of the wake pulsates in alternating spanwise directions.




(a) Re($\phi_2$) (b) Im($\phi_2$)

Figure 5.18: The pulsation mode: (a) its real component; (b) its imaginary component.

Having interpreted the spatial mode shapes in each representative set, we can form the projection bases $\Phi \in \mathbb{R}^{p \times 64}$ using the real and imaginary parts of the modes. We then




(a) Bubble pumping (b) Vortex shedding

Figure 5.19: Power spectral densities of amplitudes $\alpha_i(t)$ for the real and imaginary parts of the top three ranked modes in each cluster: (a) bubble pumping cluster; (b) vortex shedding cluster.

obtain the time series $a_i(t)$, and normalise each time series using (5.14) to obtain the time series $\alpha_i(t)$. The power spectral density of the time series, $\alpha_i(t)$, associated with the real and imaginary parts of each cluster's top three representative modes are presented in Figure 5.19. Despite the variation in the spatial features of each representative mode, the spectral characteristics of the representative modes are largely consistent with the original OMD modes in each cluster. The vortex shedding modes have a dominant frequency at $St \approx 0.2$, and the modes in the bubble pumping cluster show a peak at $St \approx 0.06$. However, in both clusters, the higher frequencies are more energetic in the higher mode numbers (i.e. for the lower-ranked POD modes of each individual cluster). The power spectral densities were found using a multi-taper estimate, with five Slepian taper functions [162].

To find a lower dimension encoding of these features, we can now train an autoencoder, as presented in §5.2, to map the normalised time series vector $\alpha(t)$ to a lower dimensional state $b(t)$. Unlike the dataset in §5.3.1, the time series show aperiodic and quasi-periodic behaviour, and therefore it is expected that a larger set of training data would be required to extract coherent patterns from the signal. LSTM neural networks are suitable for noisy datasets and can model more complex and long-term temporal patterns. However, the number of trained parameters in an LSTM network requires stringent verification through the use of a large training set and a large test-set, to ensure overfitting is avoided. How-

ever, our dataset is not large enough to avoid overfitting using a deep encoder-decoder model. Based on this limitation, and the fact that for our application the decoder function only requires the current snapshot of the encoded state, we use one hidden layer for the fully connected decoder function. Meanwhile, the encoder function will be an LSTM network. Both networks are chosen to have only one hidden state, in order to minimise the number of optimisation parameters.

Note that the design of our autoencoder is such that the last step of projecting from the time series $\hat{a}(t) \in \mathbb{R}^n$ to the velocity snapshot $x(t) \in \mathbb{R}^p$, where $p > n$, is performed through a linear projection. As a result, an optimal autoencoder can, at best, fully reconstruct the linear projection of the time-varying snapshot onto the projection bases. As a result, we can replace the time-varying snapshots $X$ with the projected snapshots

$$\mathfrak{X} = \sum_i \phi_i a_i(t)$$

where $\Phi = \{\phi_i\}$ is the matrix of representative mode-shapes. Note that due to the broadband nature of this flow, $\mathfrak{X}$ spans only 25% of the turbulent fluctuations, it is nonetheless the case that, as seen in Figure 5.16, two of the recognised and dominant dynamical features of the wake (vortex shedding, recirculation) are captured by the modes in these clusters. The omission of smaller scales in the projected snapshots curbs the problem of excessive gradients (of the neural network parameters), which, as discussed in §5.2, can lead to poor convergence results for optimisation-based model training.

We set the first 2000 snapshots of $\mathfrak{X}$ as the training set $\mathfrak{X}_{\text{train}}$ and the last 731 snapshots as the test set $\mathfrak{X}_{\text{test}}$. As mentioned, the presence of noise and the size of the network, call for ensuring that the training process is robust to overfitting. We, therefore, add an $\ell_2$-norm penalty of all optimisation parameters as a regularisation term to the cost function (5.15), a method also known as Tikhonov or ridge regularization. The resulting optimisation problem is

$$\min_{\boldsymbol{W}} \sum_{t=1}^{m}\sum_{i=1}^{n} \|\alpha_i(t_j) - \hat{\alpha}_i(t_j|\boldsymbol{W})\|_2^2 + \eta \left\| \mathfrak{X}_{\text{train}} - \sum_{j=1}^{m}\sum_{i=1}^{n} \phi_i \cdot |a_i| \hat{\alpha}_i(t_j|\boldsymbol{W})) \right\|_F^2 + \omega \|\boldsymbol{W}\|_F^2 , \tag{5.18}$$

where $\boldsymbol{W}$ is the generalised set of parameters, weights and constants, of the network, $\eta$ is the parameter that determines the emphasis on the accuracy of the reconstructed snapshots, and $\omega$ determines the emphasis on avoiding overfitting (see §5.2 for a detailed definition of the learnable weights).

At a given time step $t$, the encoder function requires the vector $\alpha(t)$ at $\{t, t-(10\Delta t), t-(20\Delta t), \cdots, t-(200\Delta t)\}$. The decoder function, on the other hand, only requires the current value of $b(t)$ to find the estimated vector $\hat{\alpha}(t)$. To solve (5.18), with $\eta = 4 \times 10^{-5}$ and $\omega = 0.9$, we use the ADAM variant of the gradient descent algorithm and update the weights at each iteration with the learning rate of 0.005.

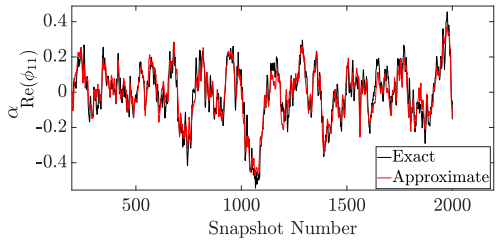
We choose the dimension of the vector $b(t)$ to be $r = 15$. Bear in mind that the original clusters included 64 OMD time series, meaning that we are attempting to reduce the underlying degrees of freedom in the time-series by over 75%.

The reconstructed training set time series $\alpha_i(t)$ corresponding to two representative modes from each cluster are presented in 5.20. (a) and (b) show the estimated temporal evolution of the real and imaginary part of the bubble pumping mode in Figure 5.13, while (c) and (d) show the time series corresponding to the pulsation mode in Figure 5.18. The time series $\alpha_i(t)$ for the first and second representative modes from the vortex shedding cluster are shown in (e)-(h). The autoencoder captures the dominant time scales of each time series well and the majority of the error

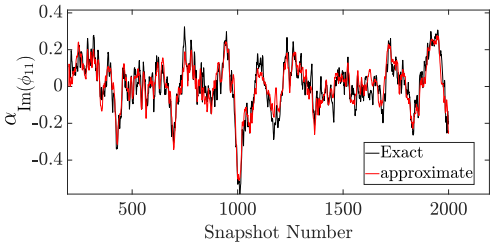
$$E(\alpha) = \frac{\left\|\alpha_i(t) - \hat{\alpha}_i(t)\right\|_F}{\left\|\alpha_i(t)\right\|_F} \times 100\%$$

comes from the smaller scales in each sequence. The Pearson's correlation $C$ between each time series $\alpha_i$ and its reconstructed counterpart $\hat{\alpha}_i$ is also shown to indicate the degree of directional similarity between the estimated and the exact time series. Mode $\phi_{1j}$ refers

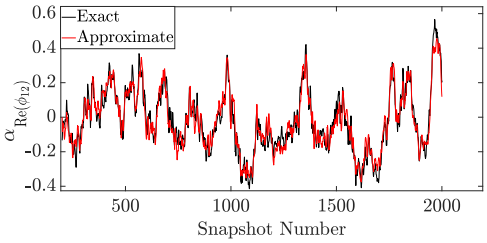
to the $j$th mode of the bubble pumping cluster, and mode $\phi_{2j}$ refers to the $j$th mode of the vortex shedding cluster.



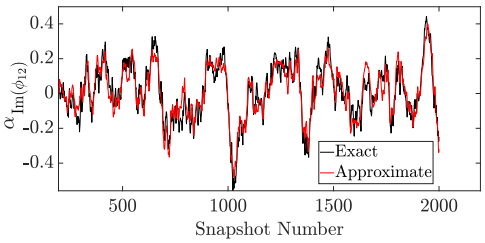

(a) Bubble pumping, $E(\alpha) = 33\%$, $C = 0.95$



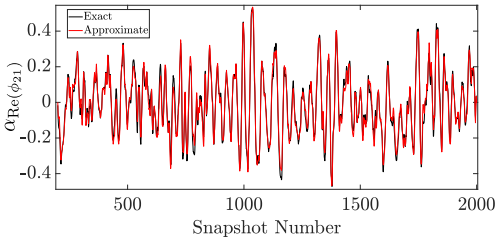

(b) Bubble pumping, $E(\alpha) = 39\%$, $C = 0.92$



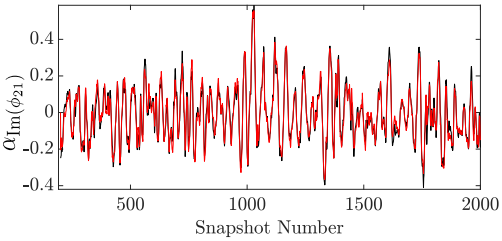

(c) Pulsation mode, $E(\alpha) = 28\%$, $C = 0.96$



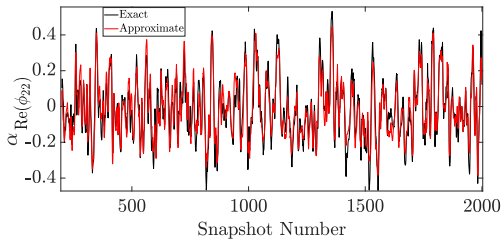

(d) Pulsation mode, $E(\alpha) = 36\%$, $C = 0.93$



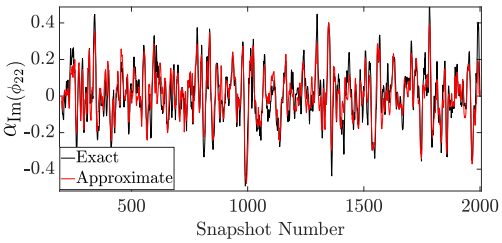

(e) Vortex shedding 1, $E(\alpha) = 16\%$, $C = 0.99$




(f) Vortex shedding 1, $E(\alpha) = 20\%$, $C = 0.98$



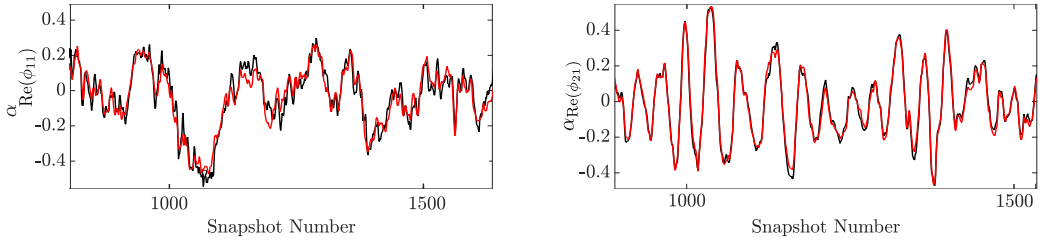

(g) Vortex shedding 2, $E(\alpha) = 46\%$, $C = 0.93$




(h) Vortex shedding 2, $E(\alpha) = 36\%$, $C = 0.88$

Figure 5.20: Exact normalised time series $\alpha(t)$ (in black) and the reconstructed time series $\hat{\alpha}(t)$ (in red). Relative error $E(\alpha)$ and correlation statistics $C$ are reported in subfigure labels.

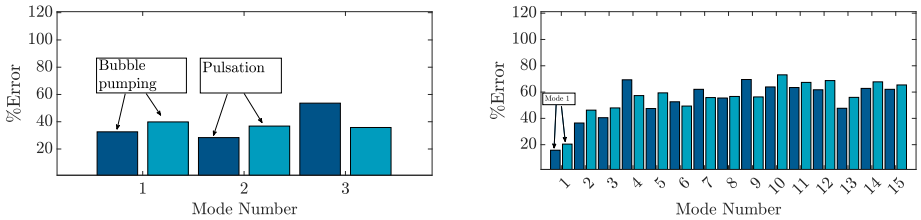
For legibility, Figure 5.21 provides a smaller temporal window of the first time series from each cluster, corresponding to figures (a) and (e) in Figure 5.20.




(a) Bubble pumping (b) Vortex shedding

Figure 5.21: A restricted time window of exact normalised time series $\alpha(t)$ (in black) and the reconstructed time series $\hat{\alpha}(t)$ (in red).

The errors associated with the normalised time series $\alpha_i(t)$, for the real and imaginary parts of the modes, are shown in Figure 5.22. The first column at each mode number refers to the real part of the representative mode and the second column refers to the imaginary part.




(a) Bubble pumping (b) Vortex shedding

Figure 5.22: The relative reconstruction error $E$ for time series $\alpha$ in bubble pumping and vortex shedding clusters.

The overall mismatch between the normalised vectors $\alpha$ and $\hat{\alpha}$ is 53.2%. Note that, as Figure 5.22 shows, the autoencoder performs better at lower mode numbers, i.e. the most energetic features. Rescaling the time series using their respective amplitudes and multiplying each time series with its respective mode we find the reconstruction error

$$E(\mathfrak{X}) = \frac{\left\| \mathfrak{X}_{\text{train}} - \hat{\mathfrak{X}}_{\text{train}} \right\|_F}{\left\| \mathfrak{X}_{\text{train}} \right\|_F} \times 100\%$$

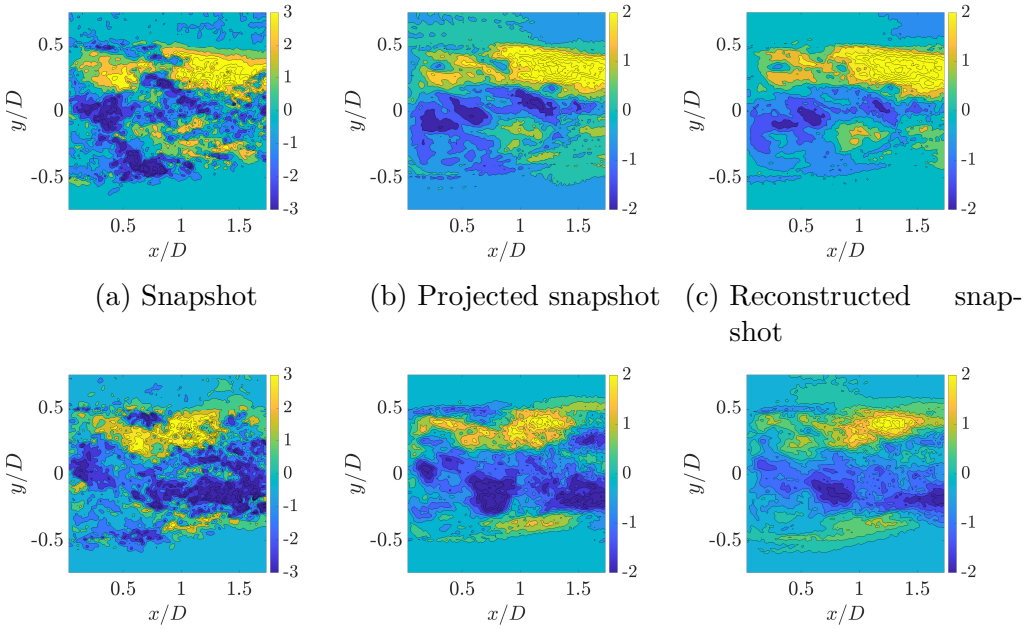
between the estimated snapshots $\hat{\mathfrak{X}}$ and the projected snapshots $\mathfrak{X}$ to be 37.3%. In other words, the autoencoder finds a reduced order state of size 15 that captures 62.7% of the energy of the flow features captured by the modes belonging to the two dominant clusters.

For comparison with the nonlinear approach discussed above, we now consider representing the flow field using only a truncated set of representative modes that retains the highest ranked modes across both clusters, one from the bubble pumping cluster and seven from the vortex shedding cluster, to form a reduced order state had a size of 16. The reconstruction error was 54%, meaning that 46% of the energy was captured. In other words, the autoencoder captures a significantly higher proportion of the flow field, 62.7% compared to 46%, using a slightly smaller state dimension, 15 as opposed to 16.

To determine whether the autoencoder can capture appropriate spatial features, we will analyse the reconstructed snapshots of the flow field, comparing them with the exact time-varying snapshots. The reconstructed snapshots $\hat{\mathfrak{X}}_{\text{train}}$ and the original time-varying snapshots $X_{train}$ have a Pearson's correlation of 0.60. In comparison, the projected snapshots $\mathfrak{X}_{\text{train}}$, using 36 representative mode shapes, and the snapshots $X_{train}$ have a Pearson's correlation of 0.64. The significant correlation shows that the reconstructed snapshots do indeed capture the general shape of the original snapshots. Figure 5.23 shows two snapshots of the time-varying flow, the projected snapshot, and the reconstructed snapshot.

To ensure the performance of the autoencoder is not affected by overfitting, after initialising the encoder function we apply the autoencoder to a set of 531 projected snapshots $\mathfrak{X}_{\text{test}}$ to find the reconstruction error $E(\mathfrak{X}) = 42.1\%$. There is also significant spatial correlation between the estimated snapshots $\hat{\mathfrak{X}}_{\text{test}}$ and the original snapshots $X_{\text{test}}$ at 0.59.

Table 5.2 presents a summary of the autoencoder's performance, as compared to the best-truncated set of representative modes. The results show that the autoencoder performs better than truncating the representative modes, in reconstructing the projected snapshots and capturing the overall shape of the near wake.




(a) Snapshot (b) Projected snapshot (c) Reconstructed snapshot

(d) Snapshot (e) Projected snapshot (f) Reconstructed snapshot

Figure 5.23: A comparison between: (a),(d) the original snapshots; (b),(e) the projected snapshots; and (c), (f) the reconstructed snapshot, at $t = 1100\Delta t$ (top row) and $t = 1700\Delta t$ (bottom row).

| | Truncation | Autoencoder (train) | Autoencoder (test) |
|---|---|---|---|
| State dimension | 16 | 15 | 15 |
| $E(\mathfrak{X})$ | 54% | 37.3% | 42.1% |
| Correlation $\langle X, \hat{\mathfrak{X}} \rangle$ | 0.55 | 0.60 | 0.59 |

Table 5.2: Summary of the autoencoder's performance

### 5.3.4 The effect of the encoded state dimension

Given the performance improvement, over a projection-based approach, achieved using an autoencoder with latent space dimension $r = 15$ in §5.3.3, we now investigate the influence of $r$ on autoencoder performance. Table 5.3 reports the relative error $E(\mathfrak{X})$ and correlation $\langle X, \hat{\mathfrak{X}} \rangle$ for three different values $r = 5, 10$ and 15 of the latent space dimen-

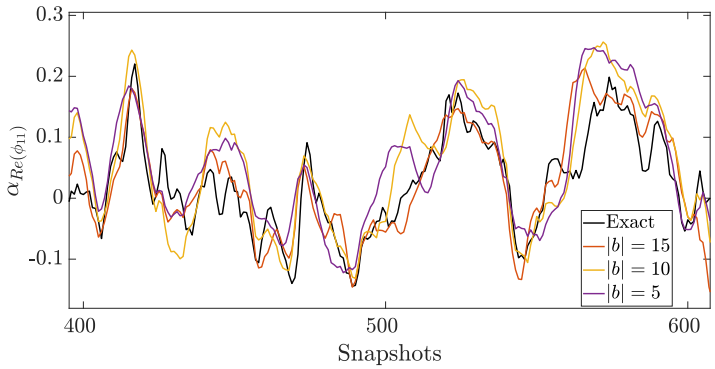
sion $r$, which describe the autoencoder's performance when reconstructing the velocity snapshots. The same training and testing data ensembles as in §5.3.3 are used here.

| | Train | Test | Train | Test | Train | Test |
|---|---|---|---|---|---|---|
| State dimension $r$ | 15 | 15 | 10 | 10 | 5 | 5 |
| $E(\mathfrak{X})$ | 37.3% | 42.1% | 47.7% | 51.5% | 62.5% | 64.6% |
| Correlation $\langle X, \hat{\mathfrak{X}} \rangle$ | 0.6 | 0.59 | 0.53 | 0.51 | 0.45 | 0.45 |

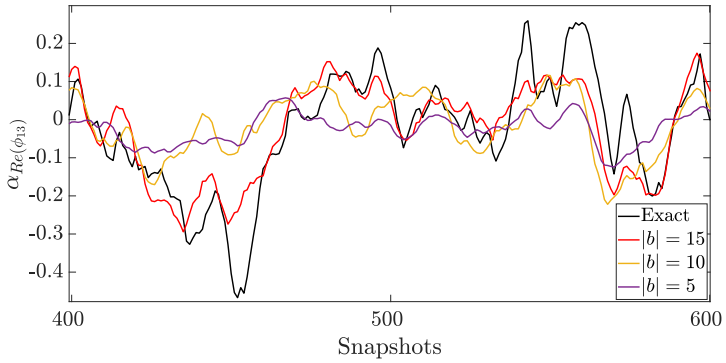
Table 5.3: Relative error and correlation statistics for varying encoded state dimension $r$, both for training and testing data ensembles.

As expected, it can be observed from Table 5.3 that decreasing $r$ corresponds to a decrease in the accuracy of the reconstructed state. For example, at $r = 5$, the reconstruction error increases to nearly 65%, indicating a significantly worst performance than a projection-based approach detailed in §5.3.3. Furthermore, a change from $r = 15$ to $r = 5$ causes the correlation between the original and reconstructed snapshots to decrease by 25% to a value of 0.45.

To give more detail about the influence of $r$ on autoencoder performance, we now look at a selection of indicative reconstructed state time series for the three considered values of $r$. Figure 5.24 shows the reconstructed time-series $\alpha$ corresponding to the real part of the bubble pumping mode, in (a), and the real part of the third representative mode of the bubble pumping cluster, in (b).




(a) Bubble-pumping mode




(b) Low-energy pulsation mode

Figure 5.24: Comparing the reconstructed DMD time series, for three different dimensions, $|b| = 5, 10$ and $|b| = 15$, of the encoded state.

Overall, using a lower value of the encoded state dimension $r$ causes a decrease, for both highlighted cases, in the accuracy of the reconstructed time series $\alpha(t_i)$. However, this effect is less pronounced for the more energetic bubble-pumping mode, $\alpha_{Re(\phi_{11})}$ in (a), than it is for the lower energy mode, $\alpha_{Re(\phi_{13})}$, in (b). An explanation for this behaviour is that the loss function for the autoencoder model includes a term that directly penalises the discrepancy between reconstructed snapshots and the original snapshots. This results in the model prioritising the reconstruction of higher energy modes. Furthermore, it can be seen in Figure 5.24 (b) that, for the lower energy mode, using an encoded state dimension $r = |b| = 5$ gives a state approximation with only low-amplitude oscillations which do not capture the large excursions from the mean present in the original data (the black "exact" series). In other words, the autoencoder model learns that since the

dimension $|b| = r$ is not sufficiently large to capture all aspects of the data, then it must concentrate on reconstructing the high-energy features. Further, it avoids accumulating errors associated with the prediction of low energy features by approximating them by time series with amplitudes close to zero.

Finally, it is interesting to note that the prioritisation of capturing high-energy features observed when reducing $r$ reflects the behaviour observed previously when increasing the value of the balance parameter $\eta$ in (5.18): a higher value of $\eta$ results in a higher emphasis on capturing of high-energy features, while lowering $\eta$ results in de-emphasising the influence of mode energy.

### 5.3.5 Modelling the dynamics using RNNs

The above discussion indicates that the use of an autoencoder in conjunction with the appropriate interpretable modal decomposition method allows us to significantly reduce the dimension of the state, whilst still capturing a significant portion of the energy of the desired coherent structures as well as the overall shape of the wake. In other words, we have shown that knowing the value of the reduced order state $b$ at time $t$, we can reconstruct the two dominant structures of the velocity field to a satisfactory extent.

The next step in designing a ROM for the flow is to model the evolution of the encoded time series. In particular, we are interested in forecasting how the flow field behaves, through forecasting the encoded state $b(t)$.

We formulate the discrete-time forecasting problem as

$$\tilde{b}(t) = f(b(t - j_1\Delta t), b(t - j_2\Delta t), \cdots, b(t - j_k\Delta t)), \tag{5.19}$$

where $\tilde{b}(t)$ is the predicted state, $j_i \in \mathbb{N}$ are constant positive integers and function $f(.)$ is a function of a chosen form. Recent studies have used RNNs as forecasting models, in conjunction with nonlinear order reduction. In particular, order-reduction using convolu-

tional autoencoders has been used in conjunction with LSTMs for modelling the dynamics of a variety of unsteady wakes [72, 49].

To avoid overfitting to the training snapshots, we minimise the number of optimisable parameters by using a GRU neural network with one hidden layer as function $f(.)$, due to its ability to extract complex temporal patterns with a smaller number of parameters than an LSTM network.

In formulating (5.15) and (5.18), we noted that the parameter $\eta$ allows us to specify the desired balance between capturing as many features as possible and concentrating on the more energetic features in the dataset. Here, we would like to use the same idea to ensure that the predicted state $\tilde{b}(t)$ prioritises the reconstruction of the dynamically significant features.

Unlike the time series $a_i(t)$, where the amplitude of the time series determines its energetic contribution to the dataset, the time series $b_i(t)$ do not follow an intuitive order of significance. This is also true for the features extracted using convolutional autoencoders in the literature [126, 57, 49]. However, an advantage of the current approach is that the pre-trained decoder function can provide a mapping between the encoded state and the physically intuitive state $a(t)$. We can therefore use an augmented cost function, where the prediction error in forecasting the state $b(t)$ is balanced with the ability of the predicted state to replicate the temporal coefficients $\hat{a}_i(t) = |a_i|\hat{\alpha}(t)$. The weights and biases of the GRU forecasting model are then found by solving the optimisation problem

$$\underset{\mathbf{W}}{\text{minimise}} \quad J := \sum_t \left( \left\| b(t) - \tilde{b}(t, \mathbf{W}) \right\|_F^2 \right) + \xi \sum_t \left( \left\| \hat{a}(t) - \tilde{a}(t, \mathbf{W}) \right\|_F^2 \right) + \zeta \left\| \mathbf{W} \right\|_F^2, \tag{5.20}$$

where $\tilde{b}(t)$ is the predicted value of the encoded state, $\xi$ is a user-defined parameter signifying the balance between minimising the reconstruction error and the energetic significance of the predicted features in replicating the velocity field, $\xi$ determines the emphasis on avoiding overfitting to the training set. The predicted time series $\tilde{a}(t)$ is found by decod-

ing the predicted state $\tilde{b}(t)$ using the pre-trained decoder function, and rescaling using the normalisation factors $|a_i|$ defined in (5.14).

For the discrete-time forecasting problem in (5.19), the quality of the predicted values can vary significantly as a function of the input time steps $b(t - j_i\Delta t)$. We, therefore, train a range of forecasting models with varying input time steps and compare their performance. We define the *visible window* as the set of time steps used as the input to the forecasting model. For each model, the visible window is parameterised, as in Figure 5.25, using the *"look-back"* parameter, $L$, the distance between each two input snapshot, $d$, and the *"look-forward"* parameter $q$. The sequential nature of the GRU model implies that the look forward parameter, $q$, is of paramount importance since it determines the difference between the temporal information provided by the hidden state (i.e. what the signal in the visible window looks like) and the predicted time step. Therefore, we will analyse the effects of varying $q$ by training and verifying the performance of neural networks at various values of $q$.

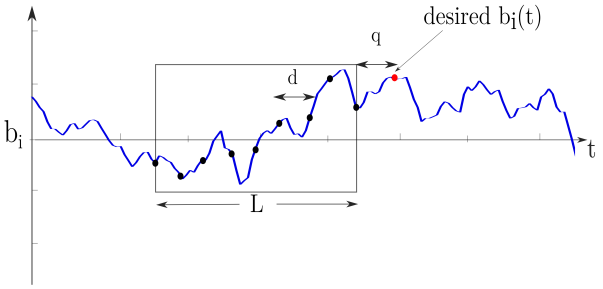



Figure 5.25: A schematic overview of the visible window in auto-regressive forecasting

We illustrate the potential of this approach by training a forecasting model for the unsteady wake of a cylinder at $Re = 60$. We will then use the same structure to model the dynamics of the turbulent flow past an axisymmetric bluff body at $Re = 5.8 \times 10^5$.

**Cylinder wake**

We solve the optimisation problem (5.20) and train a GRU forecasting model, as function $f(\cdot)$ in (5.19), to predict the value of the encoded state $b(t)$, given a finite set of values of $b$ from the previous time steps, $b(t - j\Delta t)$.

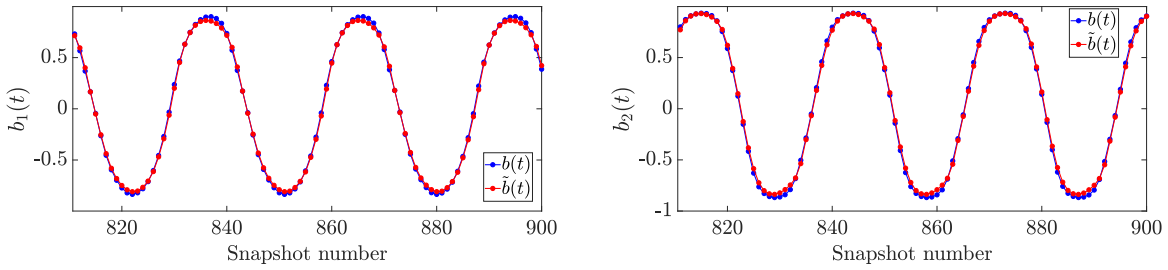
The GRU model has one hidden layer, and the dimension of the hidden layer is the same as that of the hidden state $b$, in this case 2. To train the NN we use the sequence of 600 encoded states that were obtained by encoding the training velocity snapshots. Similarly, the model is validated using the 100 snapshots of $b(t)$ found by encoding the test set.

We first train a model that predicts $b$ one step forward in time, i.e. $q = \Delta t$. The visible window is defined using the parameters $d = \Delta t$ and $L = 10\Delta t$, and we set the coefficients $\xi$ and $\zeta$ from (5.20) to 1 and 0, respectively. We use the ADAM variant of the gradient-descent algorithm with a learning rate of 0.01 to solve the optimisation problem. Note the $\ell_2$ regularisation factor $\zeta$ was set to 0 since this dataset is numerical and noise free.

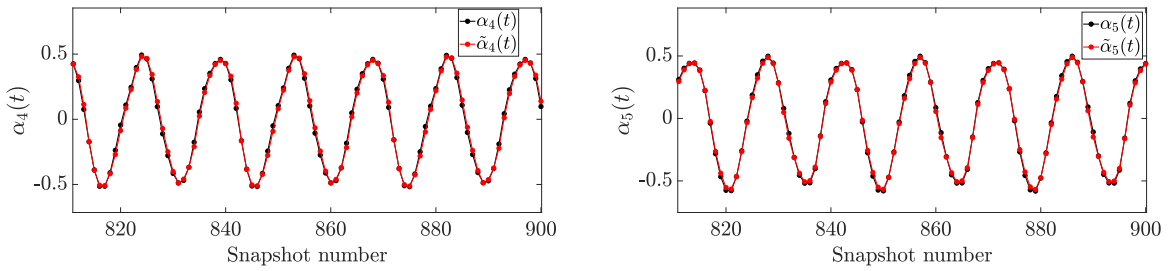
Defining the reconstruction error as

$$E(b) = \frac{\sqrt{\sum_t \left( \|\tilde{b}(t) - b(t)\|_2^2 \right)}}{\sqrt{\sum_t \|b(t)\|_2^2}} \times 100\%,$$

the resulting model predicts the encoded states with $E(b) = 3.1\%$ for the training set, and $E(b) = 3.8\%$ for the test set. The plots in Figure 5.26 (a) and (b) show the reference sequences $b_i(t)$ (in blue) and predicted time series $\tilde{b}_i(t)$ (in red). The decoded mode coefficients $\tilde{\alpha}_4(t)$ and $\tilde{\alpha}_5(t)$ are shown in (c) and (d), along with the original normalised POD time series $\alpha_4(t)$ and $\alpha_5(t)$, corresponding with the first harmonic mode. The predicted mode coefficients $\tilde{\alpha}_i(t)$ are in good agreement with the original mode coefficients $\alpha_i(t)$, with the residual errors being within $\pm 3\%$ of the residual errors of the autoencoder, in Figure 5.10.




(a) $\tilde{b}_1(t)$ (b) $\tilde{b}_2(t)$




(c) $\tilde{\alpha}_4(t)$ (d) $\tilde{\alpha}_5(t)$

Figure 5.26: Actual and predicted time series of the encoded states and selected POD amplitudes.

Importantly we can evaluate the performance of the forecasting model in predicting the velocity field, by decoding the time series $\tilde{b}(t)$. The resulting ensemble of predicted snapshots, $\tilde{X}$, can be compared to the reference snapshots $X_{train}$ and $X_{test}$ using the reconstruction error

$$E(\tilde{X}) = \frac{\|\tilde{X} - X\|_F}{\|X\|_F} \times 100\%.$$

For the case of the model with $q = \Delta t$, the predicted snapshots show good agreement with the reference snapshots with $E(\tilde{X}) = 8.7\%$ for the training set and $E(\tilde{X}) = 9.7$ for the test set. For context, the forecasting method cannot theoretically achieve smaller residuals than those achieved by the autoencoder. Comparing the prediction residuals and the reconstruction errors of the autoencoder, in Table 5.1, the forecasting model performs within 2% of the best results it can theoretically achieve.

Finally, carrying out the above analysis for five values of the look-forward parameter,

$$q = i\Delta t \quad \text{for } i = 1, \ldots, 5$$

we find that, as expected, the overall effect of increasing the parameter $q$ is a gradual increase in the residual error of the predicted flow field. However, the periodicity of the signals, with a dominant shedding period of $T_{\text{sh}} \approx 28.6\Delta t$, leads to a dip in residual errors when the furthest point in the visible window, i.e. $L + q = (10 + q)\Delta t$, is approximately half a shedding cycle out of phase with the snapshot being predicted. That is, when $q = 4\Delta t$, so that $L + q = 14\Delta t \approx T_{\text{sh}}/2$.

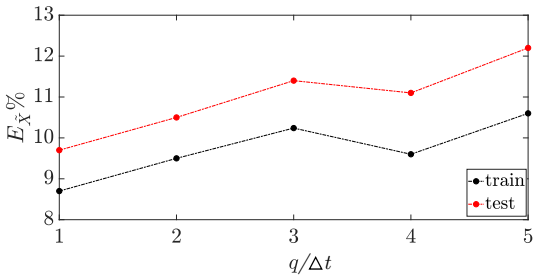



Figure 5.27: The effect of look-forward parameter $q$ on the reconstruction error $E$. Here, $\Delta t \approx \frac{1}{28.6} T_{\text{sh}}$.

### Axisymmetric bluff body

Solving the optimisation problem in (5.20) in a similar analysis to the above, we train a GRU with one hidden layer to forecast the encoded state $b(t)$ of the velocity snapshots from the turbulent wake. The GRU's hidden layer has the same dimension as the encoded state $b$, in this case 15. We use the 1800 snapshots of the encoded state, $b(t)$, corresponding to the snapshots 200 to 2000 of the velocity field, as the training set. For the test-set, we use the state $b(t)$, for the snapshots 2200 to 2731. These correspond to the same training and testing datasets as the ones used for training the autoencoder. Parameters $\xi$ and $\zeta$ in (5.20) were set to 1 and 10, respectively. The optimisation is carried out using the ADAM algorithm with the learning rate 0.01.

We then train five models, corresponding to five values of the look-forward parameter

$$q = i\Delta t \quad \text{for } i = 1, \ldots, 5$$

where $\Delta t$ is the sampling time-step of the PIV snapshots. The parameters $d = \Delta t$ and $L = 10$ remain unchanged for all trained models.

To provide some context, based on the frequency of the vortex shedding feature, the duration $5\Delta t$ corresponds to the time taken for a vortex shedding structure with the spatial length scale of $0.5D$, where $D$ is the diameter of the base of the bluff body, to move a distance of $0.3D$ downstream of the flow. In other words, this is enough time for the overall shape of the wake to vary significantly, with the average correlation between each two snapshots of the flow field that are five time-steps apart being 0.35.

The first model predicts the encoded state one time step forward in time, i.e. $q = \Delta t$. Figure 5.28 shows the first five predicted time series $\tilde{b}_i(t)$ and the corresponding $b_i(t)$. The full set of time series is available in §A.3.

The error associated with the forecasting of each encoded time series is presented in Figure 5.29.

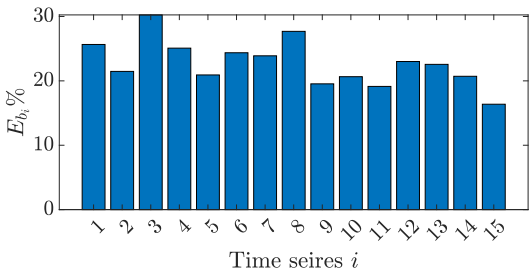
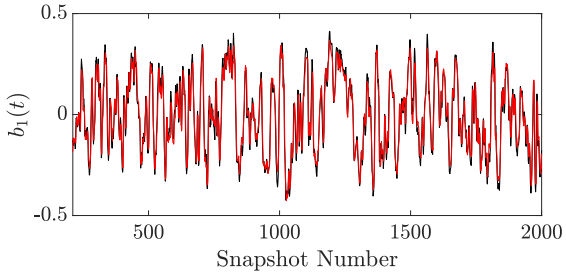
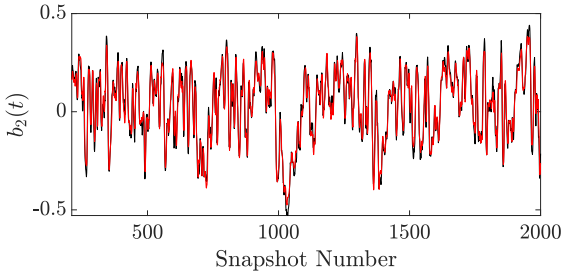



Figure 5.29: Errors associated with the encoded time series

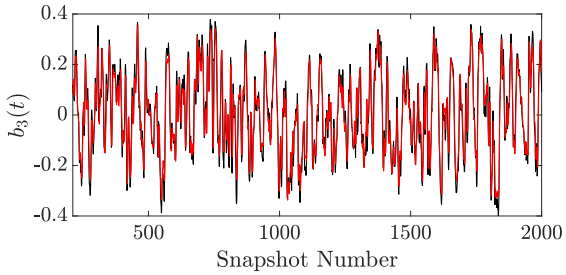
For legibility, a closer look at a few snapshots of the time series $\tilde{b}_3$ and $\tilde{b}_5$ are also provided in Figure 5.30. The overall error in forecasting the state $b(t)$ was $E(b) = 23.6\%$.



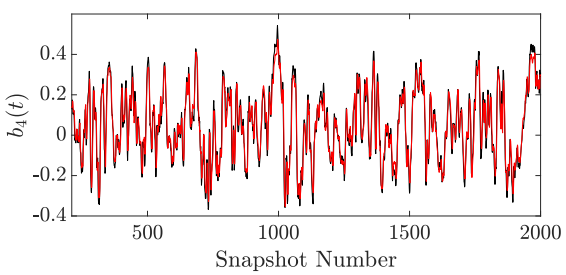

(a) $\tilde{b}_1(t)$, $E(b) = 26\%$, $C = 0.97$



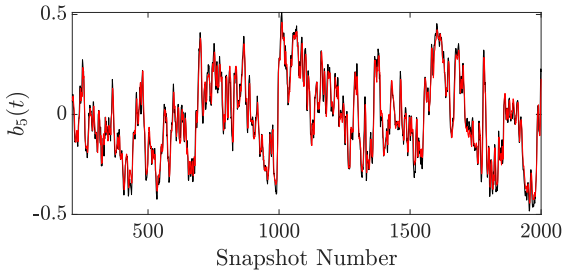

(b) $\tilde{b}_2(t)$, $E(b) = 21\%$, $C = 0.98$




(c) $\tilde{b}_3(t)$,$E(b) = 30\%$, $C = 0.95$

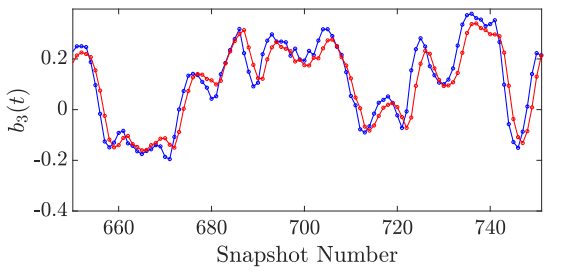



(d) $\tilde{b}_4(t)$,$E(b) = 25\%$, $C = 0.97$

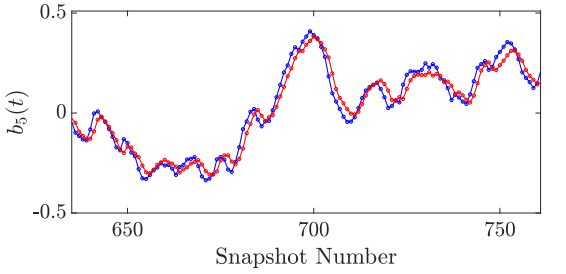



(e) $\tilde{b}_5(t)$,$E(b) = 21\%$, $C = 0.98$

Figure 5.28: Extracted time series $b_i(t)$ (in black) and forecast $\tilde{b}_i(t)$ (in red)




(a) $\tilde{b}_3(t)$

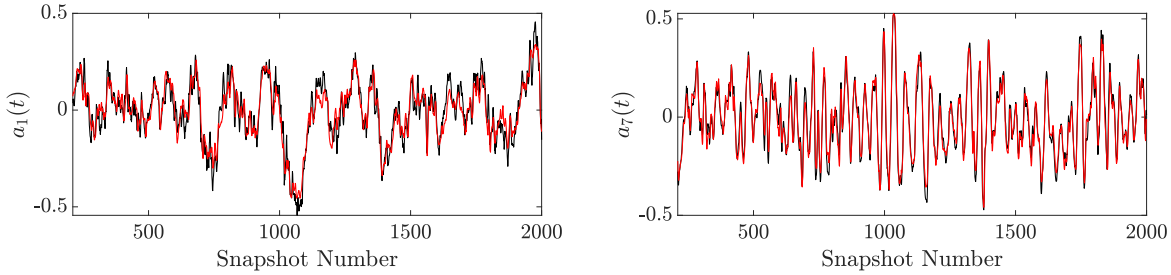



(b) $\tilde{b}_5(t)$

Figure 5.30: A few snapshots of $b_i(t)$ (in blue) and forecast $\tilde{b}_i(t)$ (in red)

By decoding the predicted state $\tilde{b}$ we can quantify the resulting mode coefficients and predicted velocity fields, leading to a more intuitive understanding of the model's performance. Examples of the forecast mode coefficients $\tilde{\alpha}_i(t)$ can be seen in figure Figure 5.31.

The time series in Figure 5.31 (a) is the predicted mode coefficient of the first bubble pumping mode, and the sequence in (b) corresponds with the first vortex shedding mode. The original time series $\alpha_i$ are also shown (in black). Comparing the original mode coefficients with the predicted sequence, we note that the predicted time series in (a) captures the large-scale variations in $\alpha_1$, and the residual is mostly due to the prevalence of shorter time scales in $\alpha_1$.




(a) $\tilde{\alpha}_1(t)$, $E(\alpha) = 35\%$ (b) $\tilde{\alpha}_7(t)$,$E(\alpha) = 22\%$

Figure 5.31: The exact time series $\alpha_i$ and forecast $\tilde{\alpha}_i(t)$ (in red)

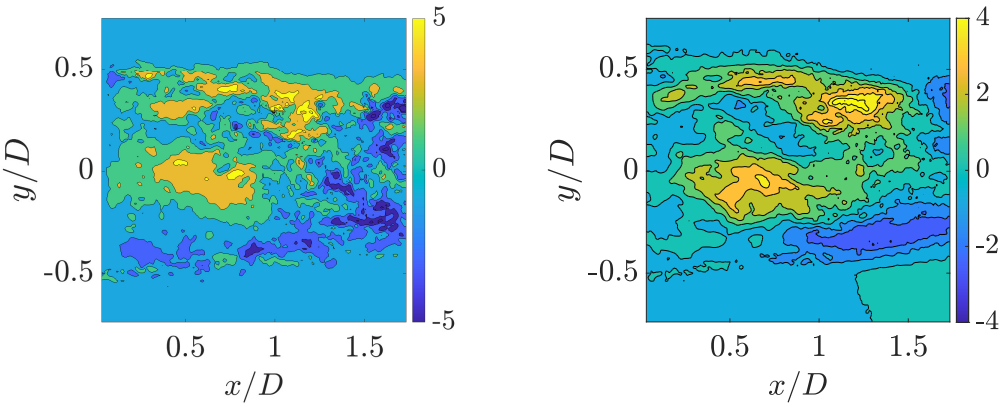
The predicted velocity snapshots, $\tilde{X}$, resemble the overall shape of the original velocity snapshots $X$, with the Pearson's correlation $\langle X, \tilde{X} \rangle = 0.59$ for the training set and $\langle X, \tilde{X} \rangle = 0.57$ for the test set. Table 5.4 summarises the results of the forecasting method for the turbulent wake of the axisymmetric bluff body.

| | Train | Test |
|---|---|---|
| $E(b)$ | 23.6% | 27.8% |
| $E(\tilde{X})$ | 41.2% | 45.8% |
| Correlation $\langle X, \tilde{X} \rangle$ | 0.59 | 0.57 |

Table 5.4: Summary of forecasting performance

An example of the predicted snapshots can be seen in Figure 5.32, alongside the original snapshot of the time-varying velocity field. As one can see, the overall shape of the velocity field is predicted well. The residual between the two snapshots is mostly due to the presence of highly energetic small-scale structures in the original snapshots. These small

structures are of two types; the first group are not temporally coherent and often spatially coincide with the large-scale vortex shedding and bubble pumping modes. An example of such structures can be seen at $(x/D, y/D) = (1.2, 0.25)$ in (a). The second group correspond with the dynamics of the flow near the separation points $(0, 0.5)$ and $(0, -0.5)$; although these structures are not as energetically significant as the two significant features studied here, they are indeed coherent in time (see, the snapshots in Figure 5.23 (a) and (d), for example). This suggests that the present model would benefit from the inclusion of higher order modes, or perhaps the use of the OMDor method, developed in §4, for extracting features that emphasise this region of the flow field.




(a) Snapshot 1840 (b) Predicted snapshot 1840

Figure 5.32: A snapshots of the stream-wise flow field and its predicted counterpart

Repeating the same analysis for five visible windows, characterised by the look-forward parameter $q$ in Figure 5.25, we find that the prediction error increases as the parameter $q$ increases. Figure 5.33 shows the variation of the prediction error with the look-forward parameter $q$. Considering the nature of the flow, it is not entirely surprising that the rate of this increase is steeper than that of the cylinder flow. However, it is important to note that the method remains capable of finding the overall shape of the flow field, as the correlation $\langle X, \tilde{X} \rangle$ remains above 0.5 even for the case where $q = 5\Delta t$. Recall that the shape of the wake varies significantly in five time steps. The fact that the predictions capture the overall shape of the wake, even in cases where the velocity snapshots are not

predicted with high accuracy, indicates that the model is effective at predicting useful statistics about the flow field.

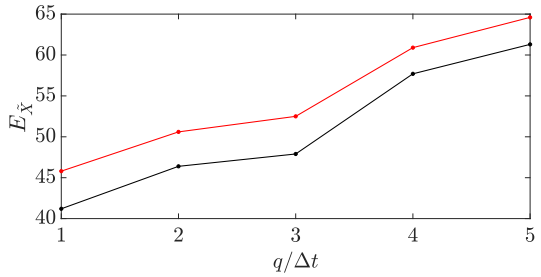
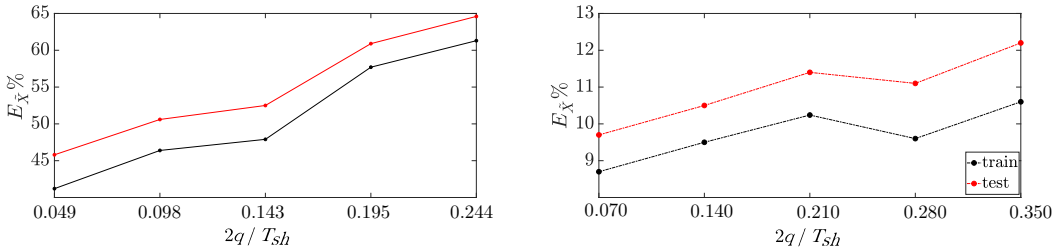



Figure 5.33: Variation of the prediction error with the look-forward parameter $q$. Here $\Delta t \approx T_{sh}/40.9$, where $T_{sh}$ is the time period of the flow's global vortex shedding mode.

### A comparison between the predictions for the two flows

Finally, we will briefly compare the performance of the dynamics models in predicting the above flows. This will also provide a more intuitive view of the quality of the predictions made for each flow, by contextualising the look-forward parameter $q$ in terms of each flow's shedding frequency. For the flow past a cylinder at Re= 60, using the shedding Strouhal number of $St \approx 0.14$ and the sampling frequency of 4 Hz, we find that there are 28.6 samples collected for each vortex shedding cycle, i.e. $T_{sh} = 28.6\Delta t$. Similarly, given the sampling rate of 720 Hz and the vortex shedding Strouhal number of $St \approx 0.23$, there are 40.9 samples collected for each vortex shedding cycle of the turbulent flow past the axisymmetric bluff body, i.e. $T_{sh} = 40.9\Delta t$. Figure 5.34 shows the reconstruction error as a function of the look-forward parameter, normalised by half the respective shedding period (i.e., in units of $q/(T_{sh}/2)$).

The reason for normalising by the half-shedding period $T_{sh}/2$ is as follows. Suppose that a flow's global vortex shedding mode has a spatial length scale $L_0$, defined to be the streamwise distance between the centres of two adjacent coherent regions of velocity fluctuations of opposite signs. The time unit $T_{sh}/2$ then corresponds to the time taken for the velocity fluctuations due to vortex shedding at a given point $(x, y)$ in space to go




(a) Axisymmetric bluff body (b) Circular cylinder

Figure 5.34: Comparison between the predictions made for the turbulent flow past an axisymmetric bluff body, (a), and the laminar flow past a circular cylinder, (b). The x-axis shows the look-forward parameter $q$ as a fraction of the shedding half-period.

from their maximum positive value to their minimum negative values. In other words, this is the time needed for a contiguous region of positive velocity fluctuations to move one length scale $L_0$ downstream.

Comparing the two flows in Figure 5.34, the RNN-based model manages to reconstruct the laminar cylinder wake in (b) with significantly lower reconstruction errors than the model for the turbulent flow past the axisymmetric bluff body. This is despite the fact that, compared to the turbulent flow in (a), predictions are made about temporal points further ahead in the shedding cycle. For example, the reconstruction error approximately at $q/(T_{sh}/2) \approx 0.25$ (i.e., corresponding to the global mode moving a distance of approximately $L_0/4$) is approximately 11% for the cylinder wake compared with 65% for turbulent flow past the axisymmetric bluff body.

Furthermore, the increasing trend observed in Figure 5.34 (a) shows that, as expected, prediction accuracy decreases if one attempts to predict further ahead in time. However, as mentioned previously, for $q = 5\Delta t$ (which corresponds to $q/(T_{sh}/2) \approx 0.25$) the predicted snapshots $\tilde{X}$ and exact snapshots $X$ still have a correlation satisfying $\langle X, \tilde{X} \rangle > 0.5$. In view of the fact that the average correlation between pairs of snapshots sampled $5\Delta t$ apart is 0.35, it is arguable that the trained model is able to make non-trivial predictions of the flow's future behaviour.

## 5.4 Conclusion

A method was developed for nonlinear extraction of low dimensional states, based on time series that correspond to interpretable and linearly extracted coherent structures. The method ensures that the extracted features are inherently interpretable by means of a decoder function that can relate the extracted state to linearly extracted time-series. The method was applied on modes and time series extracted by POD, from high dimensional numerical snapshots of a flow past a cylinder. In line with the theoretical findings of [128], the method automatically extracted a two-dimensional state that corresponds well with the first two POD time series of the flow, and found a nonlinear mapping between the encoded state and other, orthonormal, mode time series. The resulting decoded flow field captures a higher percentage of the transitioning flow, 92%, than a truncated set of POD modes with the same dimension, 75%.

The method was then applied to clusters of modes from the turbulent wake of an axisymmetric bluff body, which were found using the method described in §3. A low-dimensional temporal state was extracted to encode the subset of the dynamics of the flow captured by the modes in the clusters. The encoded state, of dimension 15, performs better than selecting 16 mode time-series in capturing the snapshot ensemble. The encoded state captures 62.7% of the underlying data and the 16 mode time-series capture 46% of the flow field.

The suitability of the encoded time series for forecasting the shape of the flow field was then analysed using a GRU network. The results show that using the encoded state, the cylinder flow can be predicted with high accuracy, with a residual error of less than 10%, and the overall shape of the turbulent flow is predicted well, with a residual error of 41% and Pearson's correlation of 0.59.

Chapter 6

# Conclusions

The increase in the quality and quantity of experimental studies using particle image velocimetry, PIV, and the decreasing cost of numerical simulations, have led to a proliferation of data relating to fluid flows. Simultaneously, there has been a steep increase in the use of data-driven techniques for the analysis and modelling of fluid flows. The methods proposed in this thesis are particularly focused on helping the practitioner extract useful information from ensembles of, usually high-dimensional, snapshots of the velocity field.

In this thesis, a number of data-driven techniques were proposed for the analysis and extraction of reduced-order models of fluid flows. Throughout the thesis, there has been an emphasis on the practicality and interpretability of the final results, for the use of a practitioner in flow-control and estimation. All techniques have been tested on a range of relevant datasets, including those which are synthetic, numerical and experimental. In particular, we have developed methods that address the following challenges in data-driven feature extraction and modelling of fluid flows:

1. In **Chapter 3** we developed a method for finding clusters of spatially, and temporally, similar modes. A number of factors, such as non-exponential behaviour and the presence of noise, can lead DMD-like algorithms to extract clusters of modes with similar mode-shapes and eigenvalues. Defining and extracting these clusters is not a trivial task, especially in broadband turbulent flows. Our method allows the

practitioner to have an intuitive view of the degrees of freedom in the underlying system. The method analyses the similarity of the extracted modes and can narrow down a large set of extracted modes, to a small set of significant clusters.

2. In **Chapter 4** we developed a method of modal decomposition that can balance the ability of the extracted modes to model the dynamics of a high dimensional dataset, with their ability to map to a small number of measurable outputs of the system. In most data-driven techniques, features are extracted that can be used for the reconstruction of an ensemble of high-dimensional snapshots. However, for control and estimation purposes, the practitioner will often have access to a limited set of measurements. However, there is no guarantee that the features necessary for the reconstruction of the high-dimensional dataset can also be mapped to the low-dimensional outputs. The coherent structures extracted using our method are by definition suitable for mapping to a low dimensional measurement of choice.

3. In **Chapter 5** we used neural networks to extract an efficient low-order representation of the flow field. By construction, these reduced order states map to physically interpretable flow features extracted using linear methods. There have been a number of developments in data-driven fluid mechanics that use the potential of machine learning to find reduced order models of the flow. These methods have been successfully employed to model the temporal evolution of benchmark velocity fields. However, a drawback of using these methods is that the emphasis on the reconstruction of snapshots can lead to reduced order states that are not human-interpretable. Our method exploits the nonlinear relationships amongst a large set of linearly extracted time series to find compact representations of the flow. The suitability of the reduced order state for modelling the flow-dynamics was also investigated.

# Future Developments

There are a number of ways to expand upon the presented work. Here we briefly mention a few possible directions that could help with the adoption of the presented methods for flow control and estimation.

1. The method presented in **Chapter 3** casts the problem of finding clusters of similar modes as one of finding communities within a graph. Our method finds these communities based on the notion of maximal cliques since it ensures a more stringent definition. However, there are a number of more computationally efficient methods for detecting such communities. The main advantage of the current method is that it does not impose arbitrary limits such as the number of modes in each cluster or strict boundaries partitioning the modes into separate graphs. Based on our experimental results, after the elimination of weak connections, the resulting graph from a typical DMD-like decomposition is sparse, and the algorithm converges in reasonable time even for challenging datasets.

   However, for a generic graph, the problem of finding all of the maximal cliques is NP-complete, and therefore future work can concentrate on ensuring that the resulting graph from a decomposition is of suitable form for the detection of maximal cliques. In particular, since finding the maximal cliques of chordal graphs is much more computationally efficient, one can ensure that the resulting graph at each iteration is chordal by imposing a suitable similarity criterion.

2. The method proposed in **Chapter 4** concentrates on ensuring that an imprint of the extracted dynamical features can be found using the sparse set of measurements that can be utilised for estimating the state of the system in real-time. There are three directions for expanding upon this work, that can be beneficial for real-life application in state estimation and control.

Firstly, the existing algorithm can be applied to a wider set of datasets that include snapshots of high-dimensional flow-fields, as well as corresponding snapshots of low-dimensional measured output. Future experimental studies that collect such datasets can be of great benefit to the current application. Also, the theoretical work can find the effect of output regulation, as in (4.8), on the observability Gramian of the resulting ROM.

Secondly, the problem that we concentrated on in **Chapter 4** relates to state estimation of autonomous systems. Another practical constraint on the application of modal decomposition in flow control is that the dynamics of the flow under control can be significantly different than that of the unforced flow. A natural extension of the existing method is to find a subspace of the high-dimensional flow field that is responsive to a given control input.

We note that existing methods such as DMD with control have attempted to consider the effect of control input on the dynamics. However, DMD with control assumes that the full state is available in real-time. For application in practical flow control and estimation problems, a balance between observability and controllability should also be considered. A preliminary optimisation problem that could potentially be used for extracting such features is

$$\begin{aligned} \underset{L,M,C}{\text{minimize}} \quad & J := \|X' - LML^\top X - LB_1\mathfrak{U}\|_F^2 + \alpha\|P - CL^\top X\|_F^2 \\ \text{s.t.} \quad & L^\top L = I, \\ & M \in \mathbb{R}^{r\times r},\ L \in \mathbb{R}^{p\times r},\ C \in \mathbb{R}^{\ell\times r},\ B_1 \in \mathbb{R}^{r\times m} \end{aligned} \tag{6.1}$$

where $B_1 \in \mathbb{R}^{r\times m}$ maps the effect of the control input on the projected state $L^\top X$, and $\mathfrak{U} \in \mathbb{R}^{m\times N}$ contains collected snapshots of control input.

Thirdly, the proposed OMDor algorithm assumes a linear relationship between the state and the output, as well as a set of linear projection bases $L$. However, the algorithm can be adjusted to include nonlinear relations, e.g. quadratic, between the

state and the output. Moreover, the projection method used can also be replaced by nonlinear projection methods such as neural networks.

3. In **Chapter 5** we looked at feature extraction methods that maintain a connection with modal decomposition. There are two directions that the current study can be expanded in. Firstly, in the current study, the reconstruction of velocity snapshots is limited by the ability of the extracted modes to reconstruct the velocity field. The motivation for this is that in most applications, a large enough set of linearly extracted modes can be used for accurate reconstruction of the velocity field. However, an improvement to the current method is to allow the model to both maintain a connection to the modal decomposition and also not be limited to the subspace spanned by the modal decomposition. To that end, a new architecture can be implemented, where convolutional layers can map the original snapshots to the portion of the velocity field not spanned by the modes.

   Finally, the connection between the encoded state and external measurements can also be used, in a similar fashion to OMDor, to ensure that the encoded state can be used for estimation of the velocity field using a limited number of measured outputs. Here we give an example of how this approach can be explored in future work.

   In the case of the flow past axisymmetric bluff bodies, there are established links between the velocity field and the pressure at the base of the flow. The length of the re-circulation region has been shown to be correlated to the area-averaged pressure disturbance at the base of the bluff body [25]. In a similar preliminary analysis, we have observed, as shown in Figure 6.1, that the time series of the representative bubble pumping mode and the length of the recirculation region are strongly correlated. It is therefore reasonable to expect that the reduced order state of the flow field, which can be mapped to this modal time series, can also be mapped to the area-averaged pressure disturbance. In a similar spirit to the analysis in **Chap-**

**ter 5**, one can find a mapping between the encoded velocity state and a series of interpretable features that can describe the shape of the pressure field.

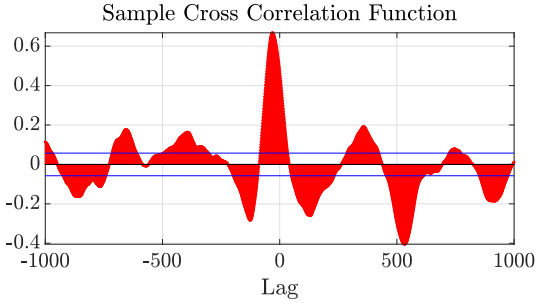



Figure 6.1: Cross correlation between the temporal coefficient of the bubble pummping mode and the length of the recirculation region.

# Bibliography


[1] Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PLOS ONE*, 11(2):1–19, 02 2016.

[2] R. J. Adrian. Hairpin vortex organization in wall turbulence. *Physics of Fluids*, 19(4):041301, 2007.

[3] B. Alexander, D. Sipp, and P. J. Schmid. Closed-loop control of an open cavity flow using reduced-order models. *Journal of Fluid Mechanics*, 641:1–50, 2009.

[4] S. Anantharamu and K. Mahesh. A parallel and streaming dynamic mode decomposition algorithm with finite precision error analysis for large data. *Journal of Computational Physics*, 380:355–377, 2019.

[5] J. D. Anderson Jr. *Fundamentals of aerodynamics*. Tata McGraw-Hill Education, 2010.

[6] H. Arbabi and I. Mezic. Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the koopman operator. *SIAM Journal on Applied Dynamical Systems*, 16(4):2096–2126, 2017.

[7] M. Ardalani-Farsa and S. Zolfaghari. Chaotic time series prediction with residual analysis method using hybrid elman–narx neural networks. *Neurocomputing*, 73(13):2540–2553, 2010.

[8] P. Artymiuk, R. Spriggs, and P. Willett. Graph theoretic methods for the analysis of structural relationships in biological macromolecules. *Journal of the American Society for Information Science and Technology*, 56(5):518 – 28, 2005.

[9] N. Aubry, P. Holmes, J. L. Lumley, and E. Stone. The dynamics of coherent structures in the wall region of a turbulent boundary layer. *Journal of Fluid Mechanics*, 192:115–173, 1988.

[10] S. Bagheri. Koopman-mode decomposition of the cylinder wake. *Journal of Fluid Mechanics*, 726:596–623, 2013.

[11] S. Bagheri. Effects of weak noise on oscillating flows: Linking quality factor, floquet modes, and koopman spectrum. *Physics of Fluids*, 26(9):094104, 2014.

[12] S. BAGHERI, L. BRANDT, and D. S. HENNINGSON. Input–output analysis, model reduction and control of the flat-plate boundary layer. *Journal of Fluid Mechanics*, 620:263–298, 2009.

[13] P. Baj, P. J. K. Bruce, and O. R. H. Buxton. The triple decomposition of a fluctuating velocity field in a multiscale flow. *Physics of Fluids*, 27(7), 2015.

[14] P. Baldi and K. Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, 1989.

[15] D. Barkley. Linear analysis of the cylinder wake mean flow. *EPL (Europhysics Letters)*, 75(5):750, 2006.

[16] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria, and U. R. Acharya. Abcdm: An attention-based bidirectional cnn-rnn deep model for sentiment analysis. *Future Generation Computer Systems*, 115:279–294, 2021.

[17] J. S. Bayer. *Learning sequence representations*. PhD thesis, Technische Universität München, 2015.

[18] M. Beit-Sadi, J. Krol, and A. Wynn. Data-driven feature identification and sparse representation of turbulent flows. *International Journal of Heat and Fluid Flow*, 88:108766, 2021.

[19] E. Berger, M. Sastuba, D. Vogt, B. Jung, and H. Ben Amor. Estimation of perturbations in robotic behavior using dynamic mode decomposition. *Advanced Robotics*, 29(5):331–343, 2015.

[20] E. Berger, D. Scholz, and M. Schumm. Coherent vortex structures in the wake of a sphere and a circular disk at rest and under forced vibrations. *Journal of Fluids and Structures*, 4(3):231 – 257, 1990.

[21] M. Bergmann and L. Cordier. Optimal control of the cylinder wake in the laminar regime by trust-region methods and pod reduced-order models. *Journal of Computational Physics*, 227(16):7813–7840, 2008.

[22] M. Bergmann, L. Cordier, and J.-P. Brancher. Optimal rotary control of the cylinder wake using proper orthogonal decomposition reduced-order model. *Physics of Fluids*, 17(9):097101, 2005.

[23] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

[24] A. Bjorck and G. H. Golub. Numerical methods for computing angles between linear subspaces. *Mathematics of Computation*, 27(123):579–594, 1973.

[25] R. Brackston. *Feedback control of three-dimensional bluff body wakes for efficient drag reduction*. PhD thesis, Imperial College London, 2017.

[26] M. C. Brindise and P. P. Vlachos. Proper orthogonal decomposition truncation method for data denoising and order reduction. *Experiments in Fluids*, 58(4), 2017.

[27] C. Bron and J. Kerbosch. Algorithm 457: Finding all cliques of an undirected graph. *Commun. ACM*, 16(9):575–577, 1973.

[28] G. L. Brown and A. Roshko. On density effects and large structure in turbulent mixing layers. *Journal of Fluid Mechanics*, 64(4):775–816, 1974.

[29] B. W. Brunton, L. A. Johnson, J. G. Ojemann, and J. N. Kutz. Extracting spatial–temporal coherent patterns in large-scale neural recordings using dynamic mode decomposition. *Journal of Neuroscience Methods*, 258:1–15, 2016.

[30] S. L. Brunton, B. W. Brunton, J. L. Proctor, E. Kaiser, and J. N. Kutz. Chaos as an intermittently forced linear system. *Nature Communications*, 8(1), May 2017.

[31] S. L. Brunton, B. R. Noack, and P. Koumoutsakos. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52(1):477–508, 2020.

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

[33] M. Budišić, R. Mohr, and I. Mezić. Applied koopmanism. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(4):047510, 2012.

[34] Y. Bury and T. Jardin. Transitions to chaos in the wake of an axisymmetric bluff body. *Physics Letters A*, 376(45):3219 – 3222, 2012.

[35] A. Cassinelli, M. de Giovanetti, and Y. Hwang. Streak instability in near-wall turbulence revisited. *Journal of Turbulence*, 18(5):443–464, 2017.

[36] D. Cavar and K. E. Meyer. Large eddy simulation of turbulent jet in cross flow: Part 2–pod analysis and identification of coherent structures. *International Journal of Heat and Fluid Flow*, 36:35–46, 2012.

[37] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019.

[38] S. K. Chandar. Grey wolf optimization-elman neural network model for stock price prediction. *Soft Computing*, 25(1):649–658, 2021.

[39] K. K. Chen, J. H. Tu, and C. W. Rowley. Variants of dynamic mode decomposition: Boundary condition, Koopman, and Fourier analyses. *Journal of Nonlinear Science*, 22(6):887 – 915, December 2012.

[40] A. Chiarini, D. Gatti, A. Cimarelli, and M. Quadrio. Structure of turbulence in the flow around a rectangular cylinder. *Journal of Fluid Mechanics*, 946:A35, 2022.

[41] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Oct. 2014.

[42] A. M. Dai and Q. V. Le. Semi-supervised sequence learning. *Advances in neural information processing systems*, 28:3079–3087, 2015.

[43] A. d'Aspremont, L. El Ghaoui, M. I. Jordan, and G. R. G. Lanckriet. A direct formulation for sparse pca using semidefinite programming. *SIAM Review*, 49(3):434–448, 2007.

[44] Y. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization, 2014.

[45] S. T. M. Dawson, M. S. Hemati, M. O. Williams, and C. W. Rowley. Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition. *Experiments in Fluids*, 57(3):42, 2016.

[46] G. Dergham, D. Sipp, J.-C. Robinet, and A. Barbagallo. Model reduction for fluids using frequential snapshots. *Physics of Fluids*, 23(6):064101, 2011.

[47] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, Sep 1936.

[48] B. Eisenhower, T. Maile, M. Fischer, and I. Mezic. Decomposing building system data for model validation and analysis using the koopman operator. *SimBuild 2010*, 01 2010.

[49] H. Eivazi, H. Veisi, M. H. Naderi, and V. Esfahanian. Deep neural networks for nonlinear model order reduction of unsteady flows. *Physics of Fluids*, 32(10):105104, 2020.

[50] J. L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990.

[51] J. L. Elman. Language as a dynamical system. *Mind as motion: Explorations in the dynamics of cognition*, pages 195–223, 1995.

[52] B. P. Epps and A. H. Techet. An error threshold criterion for singular value decomposition modes extracted from piv data. *Experiments in Fluids*, 48(2):355–367, February 2010.

[53] N. B. Erichson and C. Donovan. Randomized low-rank dynamic mode decomposition for motion detection. *Computer Vision and Image Understanding*, 146:40–50, 2016.

[54] L.-H. Feng, J.-J. Wang, and C. Pan. Proper orthogonal decomposition analysis of vortex dynamics of a circular cylinder under synthetic jet control. *Physics of Fluids*, 23(1):014106, 2011.

[55] P. Freymuth. On transition in a separated laminar boundary layer. *Journal of Fluid Mechanics*, 25(4):683–704, 1966.

[56] K. Fukami, K. Hasegawa, T. Nakamura, M. Morimoto, and K. Fukagata. Model order reduction with neural networks: Application to laminar and turbulent flows. *SN Computer Science*, 2(6):467, 2021.

[57] K. Fukami, T. Nakamura, and K. Fukagata. Convolutional neural network based hierarchical autoencoder for nonlinear mode decomposition of fluid field data. *Physics of Fluids*, 32(9):095110, 2020.

[58] M. Gaster, E. Kit, and I. Wygnanski. Large-scale structures in a forced turbulent mixing layer. *Journal of Fluid Mechanics*, 150:23–39, 1985.

[59] M. Gavish and D. L. Donoho. The optimal hard threshold for singular values is $4/\sqrt{3}$. *IEEE Transactions on Information Theory*, 60(8):5040–5053, 2014.

[60] E. Gillies. Low-dimensional control of the circular cylinder wake. *Journal of Fluid Mechanics*, 371:157–178, 1998.

[61] C. Gin, B. Lusch, S. L. Brunton, and J. N. Kutz. Deep learning models for global coordinate transformations that linearise pdes. *European Journal of Applied Mathematics*, 32(3):515–539, 2021.

[62] F. J. Gonzalez and M. Balajewicz. Deep convolutional recurrent autoencoders for learning low-dimensional feature dynamics of fluid systems. *arXiv preprint arXiv:1808.01346*, 2018.

[63] I. Goodfellow, O. Vinyals, and A. Saxe. Qualitatively characterizing neural network optimization problems. In *International Conference on Learning Representations*, 2015.

[64] I. J. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, Cambridge, MA, USA, 2016.

[65] P. J. Goulart, A. Wynn, and D. Pearson. Optimal mode decomposition for high dimensional systems. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 4965–4970, 2012.

[66] J. Graff, M. Ringuette, T. Singh, and F. Lagor. Reduced-order modeling for dynamic mode decomposition without an arbitrary sparsity parameter. *AIAA Journal*, 07 2020.

[67] W. R. Graham, J. Peraire, and K. Y. Tang. Optimal control of vortex shedding using low-order models. part ii—model-based control. *International Journal for Numerical Methods in Engineering*, 44(7):973–990, 1999.

[68] M. Grilli, P. J. Schmid, S. Hickel, and N. A. Adams. Analysis of unsteady behaviour in shockwave turbulent boundary layer interaction. *Journal of Fluid Mechanics*, 700:16 – 28, 2012.

[69] J. Grosek and J. N. Kutz. Dynamic mode decomposition for real-time background/foreground separation in video. *arXiv preprint arXiv:1404.7592*, 2014.

[70] F. Gueniat, L. Pastur, and F. Lusseyran. Investigating mode competition and three-dimensional features from two-dimensional velocity fields in an open cavity flow by modal decompositions. *Physics of Fluids*, 26(8), August 2014.

[71] M. Habibi, S. Dawson, and A. Arzani. Data-driven pulsatile blood flow physics with dynamic mode decomposition. *Fluids*, 5(3):111, 2020.

[72] K. Hasegawa, K. Fukami, T. Murata, and K. Fukagata. Machine-learning-based reduced-order modeling for unsteady flows around bluff bodies of various shapes. *Theoretical and Computational Fluid Dynamics*, 34(4):367–383, May 2020.

[73] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. 2015.

[74] L. H. O. Hellström, A. Sinha, and A. J. Smits. Visualizing the very-large-scale motions in turbulent pipe flow. *Physics of Fluids*, 23(1):011703, 2011.

[75] M. S. Hemati, C. W. Rowley, E. A. Deem, and L. N. Cattafesta. De-biasing the dynamic mode decomposition for applied koopman spectral analysis of noisy

datasets. *Theoretical and Computational Fluid Dynamics*, 31(4):349–368, Aug 2017.

[76] M. S. Hemati, M. O. Williams, and C. W. Rowley. Dynamic mode decomposition for large and streaming datasets. *Physics of Fluids*, 26, 2014.

[77] R. Hermann and A. Krener. Nonlinear controllability and observability. *IEEE Transactions on Automatic Control*, 22(5):728–740, 1977.

[78] B. Herrmann, P. J. Baddoo, R. Semaan, S. L. Brunton, and B. J. McKeon. Data-driven resolvent analysis. *Journal of Fluid Mechanics*, 918:A10, 2021.

[79] C. Ho and P. Huerre. Perturbed free shear layers. *Annual Review of Fluid Mechanics*, 16(1):365–422, 1984.

[80] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[81] R. Horaud and T. Skordas. Stereo correspondence through feature grouping and maximal cliques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(11):1168–1180, 1989.

[82] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[83] N. Hutchins and I. Marusic. Evidence of very long meandering features in the logarithmic region of turbulent boundary layers. *Journal of Fluid Mechanics*, 579:1–28, 2007.

[84] M. Ilak and C. Rowley. Reduced-order modeling of channel flow using traveling pod and balanced pod. In *3rd AIAA Flow Control Conference*, page 3194, 2006.

[85] M. Jordan. Serial order: a parallel distributed processing approach. technical report, june 1985-march 1986. Technical report, California Univ., San Diego, La Jolla (USA). Inst. for Cognitive Science, 1986.

[86] M. R. Jovanovic, P. J. Schmid, and J. W. Nichols. Sparsity-promoting dynamic mode decomposition. *Physics of Fluids*, 26(2), 2014.

[87] J.-N. Juang and R. S. Pappa. An eigensystem realization algorithm for modal parameter identification and model reduction. *Journal of Guidance, Control, and Dynamics*, 8(5):620–627, 1985.

[88] K. Kandasamy, W. Neiswanger, J. Schneider, B. Póczos, and E. P. Xing. Neural architecture search with bayesian optimisation and optimal transport. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, page 2020–2029. Curran Associates Inc., 2018.

[89] K. Kandasamy, K. R. Vysyaraju, W. Neiswanger, B. Paria, C. R. Collins, J. Schneider, B. Poczos, and E. P. Xing. Tuning hyperparameters without grad students: Scalable and robust bayesian optimisation with dragonfly. *Journal of Machine Learning Research*, 21(81):1–27, 2020.

[90] K. Kankanwadi. Effects of free-stream turbulence on entrainment into a cylinder wake. Master's thesis, Department of Aeronautics, 2017.

[91] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[92] S. Koichi, M. Arisaka, H. Koshino, A. Aoki, S. Iwata, T. Uno, and H. Satoh. Chemical structure elucidation from 13c nmr chemical shifts: Efficient data processing using bipartite matching and maximal clique algorithms. *Journal of Chemical Information and Modeling*, 54(4):1027 – 1035, 2014.

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

[94] M. Korda and I. Mezić. On convergence of extended dynamic mode decomposition to the koopman operator. *Journal of Nonlinear Science*, 28(2):687–710, 2018.

[95] J. Kou and W. Zhang. An improved criterion to select dominant modes from dynamic mode decomposition. *European Journal of Mechanics - B/Fluids*, 62:109 – 129, 2017.

[96] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[97] J. Krol and A. Wynn. Dynamic reconstruction and data reconstruction for subsampled or irregularly sampled data. *Journal of Fluid Mechanics*, 825:133 – 66, 2017.

[98] J. N. Kutz, X. Fu, and S. L. Brunton. Multiresolution dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 15(2):713–735, 2016.

[99] S. Le Clainche and J. M. Vega. Higher order dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 16(2):882–925, 2017.

[100] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015.

[101] C. Y. Li, Z. Chen, T. K. T. Tse, A. U. Weerasuriya, X. Zhang, Y. Fu, and X. Lin. A parametric and feasibility study for data sampling of the dynamic mode decomposition: range, resolution, and universal convergence states. *Nonlinear Dynamics*, Jan 2022.

[102] P. Li, Y. Li, Q. Xiong, Y. Chai, and Y. Zhang. Application of a hybrid quantized elman neural network in short-term load forecasting. *International Journal of Electrical Power & Energy Systems*, 55:749–759, 2014.

[103] Q. Li, F. Dietrich, E. M. Bollt, and I. G. Kevrekidis. Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the koopman operator. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(10):103111, 2017.

[104] Z. C. Lipton. A critical review of recurrent neural networks for sequence learning. *ArXiv*, abs/1506.00019, 2015.

[105] A. Lozano-Durán and J. Jiménez. Time-resolved evolution of coherent structures in turbulent channels: characterization of eddies and cascades. *Journal of Fluid Mechanics*, 759:432–471, 2014.

[106] L. Lu and G. Papadakis. An iterative method for the computation of the response of linearised navier-stokes equations to harmonic forcing and application to forced cylinder wakes. *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS*, 74:794–817, 2014.

[107] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang. The expressive power of neural networks: A view from the width. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[108] D. G. Luenberger. Observing the state of a linear system. *IEEE transactions on military electronics*, 8(2):74–80, 1964.

[109] J. Lumley. Stochastic tools in turbulence. 1970.

[110] B. Lusch, J. N. Kutz, and S. L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9(1):4950, 2018.

[111] O. A. Mahfoze, A. Moody, A. Wynn, R. D. Whalley, and S. Laizet. Reducing the skin-friction drag of a turbulent boundary-layer flow with low-amplitude wall-normal blowing within a bayesian optimization framework. *Phys. Rev. Fluids*, 4, Sep 2019.

[112] J. Mann and J. N. Kutz. Dynamic mode decomposition for financial trading strategies. *Quantitative Finance*, 16(11):1643–1655, 2016.

[113] S. Mariappan, A. D. Gardner, K. Richter, and M. Raffel. Analysis of dynamic stall using dynamic mode decomposition technique. *AIAA journal*, 52(11):2427–2439, 2014.

[114] I. Marusic. On the role of large-scale structures in wall turbulence. *Physics of Fluids*, 13(3):735–743, 2001.

[115] L. Massa, R. Kumar, and P. Ravindran. Dynamic mode decomposition analysis of detonation waves. *Physics of Fluids*, 24(6):066101, 2012.

[116] R. Maulik, B. Lusch, and P. Balaprakash. Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders. *Physics of Fluids*, 33(3):037106, 2021.

[117] K. Meyer, J. Pedersen, and O. Özcan. A turbulent jet in crossflow analysed with proper orthogonal decomposition. *Journal of Fluid Mechanics*, 583:199–227, 2007.

[118] I. Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1-3):309–325, 2005.

[119] I. Mezić. Analysis of fluid flows via spectral properties of the koopman operator. *Annual Review of Fluid Mechanics*, 45(1):357–378, 2013.

[120] A. Michalke. On spatially growing disturbances in an inviscid shear layer. *Journal of Fluid Mechanics*, 23(3):521–544, 1965.

[121] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, pages 1045–1048, 2010.

[122] N. Mohan, K. Soman, and S. Sachin Kumar. A data-driven strategy for short-term electric load forecasting using dynamic mode decomposition model. *Applied Energy*, 232:229–244, 2018.

[123] B. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, 26(1):17–32, February 1981.

[124] J. Morton, A. Jameson, M. J. Kochenderfer, and F. Witherden. Deep dynamical modeling and control of unsteady fluid flows. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[125] T. W. Muld, G. Efraimsson, and D. S. Henningson. Flow structures around a high-speed train extracted using proper orthogonal decomposition and dynamic mode decomposition. *Computers & Fluids*, 57:87–97, 2012.

[126] T. Murata, K. Fukami, and K. Fukagata. Nonlinear mode decomposition with convolutional neural networks for fluid dynamics. *Journal of Fluid Mechanics*, 882:A13, 2020.

[127] B. Noack, P. PAPAS, and P. Monkewitz. The need for a pressure-term representation in empirical Galerkin models of incompressible shear flows. *Journal of Fluid Mechanics*, 523, 01 2005.

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

[129] B. R. Noack, W. Stankiewicz, M. Morzyński, and P. J. Schmid. Recursive dynamic mode decomposition of transient and post-transient wake flows. *Journal of Fluid Mechanics*, 809:843–872, 2016.

[130] T. Nonomura, H. Shibata, and R. Takaki. Dynamic mode decomposition using a kalman filter for parameter estimation. *AIP Advances*, 8(10):105106, 2018.

[131] K. Oberleithner, M. Sieber, C. N. Nayeri, C. O. Paschereit, C. Petz, H.-C. Hege, B. R. Noack, and I. Wygnanski. Three-dimensional coherent structures in a swirling jet undergoing vortex breakdown: stability analysis and empirical mode construction. *Journal of fluid mechanics*, 679:383–414, 2011.

[132] A. Okajima. Strouhal numbers of rectangular cylinders. *Journal of Fluid Mechanics*, 123:379–398, 1982.

[133] N. Omata and S. Shirayama. A novel method of low-dimensional representation for temporal behavior of flow fields using deep autoencoder. *AIP Advances*, 9(1):015006, 2019.

[134] S. E. Otto and C. W. Rowley. Linearly recurrent autoencoder networks for learning dynamics. *SIAM Journal on Applied Dynamical Systems*, 18(1):558–593, 2019.

[135] A. R. Oxlade, J. F. Morrison, A. Qubain, and G. Rigas. High-frequency forcing of a turbulent axisymmetric wake. *Journal of Fluid Mechanics*, 770:305–318, 2015.

[136] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1310–1318, 2013.

[137] J. L. Proctor, S. L. Brunton, and J. N. Kutz. Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, 15(1):142–161, 2016.

[138] J. L. Proctor and P. A. Eckhoff. Discovering dynamic patterns from infectious disease data using dynamic mode decomposition. *International health*, 7(2):139–145, 2015.

[139] D. Rempfer. On low-dimensional galerkin models for fluid flow. *Theoretical and Computational Fluid Dynamics*, 14(2):75–88, Jun 2000.

[140] D. Rempfer and H. F. Fasel. Evolution of three-dimensional coherent structures in a flat-plate boundary layer. *Journal of Fluid Mechanics*, 260:351–375, 1994.

[141] W. C. Reynolds, D. E. Parekh, P. J. D. Juvet, and M. J. D. Lee. Bifurcating and blooming jets. *Annual Review of Fluid Mechanics*, 35(1):295–315, 2003.

[142] G. Rigas, A. Oxlade, A. Morgans, and J. Morrison. Low-dimensional dynamics of a turbulent axisymmetric wake. *Journal of Fluid Mechanics*, 755, 2014.

[143] D. Rockwell and E. Naudascher. Review—Self-Sustaining Oscillations of Flow Past Cavities. *Journal of Fluids Engineering*, 100:152–165, 06 1978.

[144] A. Roshko. On the wake and drag of bluff bodies. *Journal of the Aeronautical Sciences*, 22(2):124–132, 1955.

[145] C. W. Rowley. Model reduction for fluids, using balanced proper orthogonal decomposition. *International Journal of Bifurcation and Chaos*, 15(03):997–1013, 2005.

[146] C. W. Rowley, I. Mezic, S. Bagheri, P. Schlatter, and D. S. Henningson. Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics*, 641:115 – 127, 2009.

[147] A. Ruhe. Rational krylov sequence methods for eigenvalue computation. *Linear Algebra and its Applications*, 58:391–405, 1984.

[148] D. Sashidhar and J. N. Kutz. Bagging, optimized dynamic mode decomposition (bop-dmd) for robust, stable forecasting with spatial and temporal uncertainty-quantification, 2021.

[149] A. M. Saxe, J. L. McClelland, and S. Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[150] H. Schaeffer and S. G. McCalla. Sparse model selection via integral terms. *Phys. Rev. E*, 96:023302, Aug 2017.

[151] P. J. Schmid. Nonmodal stability theory. *Annu. Rev. Fluid Mech.*, 39:129–162, 2007.

[152] P. J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5 – 28, 2010.

[153] P. J. Schmid. Application of the dynamic mode decomposition to experimental data. *Experiments in Fluids*, 50(4):1123 – 1130, 2011.

[154] P. J. Schmid, L. Li, M. P. Juniper, and O. Pust. Applications of the dynamic mode decomposition. *Theoretical and Computational Fluid Dynamics*, 25(1):249–259, Jun 2011.

[155] A. Seena and H. J. Sung. Dynamic mode decomposition of turbulent cavity flows for self-sustained oscillations. *International Journal of Heat and Fluid Flow*, 32(6):1098–1110, 2011.

[156] S. G. Siegel, J. Seidel, C. Fagley, D. M. Luchtenburg, K. Cohen, and T. Mclaughlin. Low-dimensional modelling of a transient cylinder wake using double proper orthogonal decomposition. *Journal of Fluid Mechanics*, 610:1–42, 2008.

[157] D. Sipp and A. Lebedev. Global stability of base and mean flows: A general approach and its applications to cylinder and open cavity flows. *Journal of Fluid Mechanics*, 593:333 – 358, 2007.

[158] J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[159] K. Taira, S. L. Brunton, S. T. M. Dawson, C. W. Rowley, T. Colonius, B. J. McKeon, O. T. Schmidt, S. Gordeyev, V. Theofilis, and L. S. Ukeiley. Modal analysis of fluid flows: An overview. *AIAA Journal*, 55:4013–4041, 2017.

[160] N. Takeishi, Y. Kawahara, and T. Yairi. Learning koopman invariant subspaces for dynamic mode decomposition. *arXiv preprint arXiv:1710.04340*, 2017.

[161] J. Taylor and M. N. Glauser. Towards practical flow sensing and control via pod and lse based low-dimensional tools. *J. Fluids Eng.*, 126(3):337–345, 2004.

[162] D. Thomson. Spectrum estimation and harmonic analysis. *Proceedings of the IEEE*, 70(9):1055–1096, 1982.

[163] G. Tissot, L. Cordier, N. Benard, and B. R. Noack. Model reduction using dynamic mode decomposition. *Comptes Rendus Mécanique*, 342(6-7):410–416, 2014.

[164] E. Tomita, A. Tanaka, and H. Takahashi. The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science*, 363(1):28 – 42, 2006.

[165] A. Towne, O. T. Schmidt, and T. Colonius. Spectral proper orthogonal decomposition and its relationship to dynamic mode decomposition and resolvent analysis. *Journal of Fluid Mechanics*, 847:821–867, 2018.

[166] J. Tu, C. W. Rowley, D. Luchtenburg, S. Brunton, and N. Kutz. On dynamic mode decomposition: Theory and applications. *Journal of Computational Dynamics*, 1(2):391–421, 2014.

[167] M. Udell. *Generalized low rank models*. Stanford University, 2015.

[168] Z.-H. Wan, L. Zhou, B.-F. Wang, and D.-J. Sun. Dynamic mode decomposition of forced spatially developed transitional jets. *European Journal of Mechanics - B/Fluids*, 51:16–26, 2015.

[169] H. F. Wang, H. L. Cao, and Y. Zhou. Pod analysis of a finite-length cylinder near wake. *Experiments in Fluids*, 55(8):1790, Jul 2014.

[170] J. Wang, W. Zhang, Y. Li, J. Wang, and Z. Dang. Forecasting wind speed using empirical mode decomposition and elman neural network. *Applied Soft Computing*, 23:452–459, 2014.

[171] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley. A data–driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.

[172] M. O. Williams, C. W. Rowley, and I. G. Kevrekidis. A kernel-based approach to data-driven koopman spectral analysis. *arXiv preprint arXiv:1411.2260*, 2014.

[173] C. H. K. Williamson. Vortex dynamics in the cylinder wake. *Annual Review of Fluid Mechanics*, 28(1):477–539, 1996.

[174] A. Wynn, D. Pearson, B. Ganapathisubramani, and P. Goulart. Optimal mode decomposition for unsteady flows. *Journal of Fluid Mechanics*, 733:473 – 503, 2013.

[175] S. Yagmur, S. Dogan, M. H. Aksoy, E. Canli, and M. Ozgoren. Experimental and numerical investigation of flow structures around cylindrical bluff bodies. In *EPJ Web of Conferences*, volume 92, page 02113. EDP Sciences, 2015.

[176] E. Yeung, S. Kundu, and N. Hodas. Learning deep neural network representations for koopman operators of nonlinear dynamical systems. In *2019 American Control Conference (ACC)*, pages 4832–4839, 2019.

[177] C. Zhang and S. I. Chernyshenko. Quasisteady quasihomogeneous description of the scale interactions in near-wall turbulence. *Phys. Rev. Fluids*, 1:014401, May 2016.

[178] H. Zhang, C. W. Rowley, E. A. Deem, and L. N. Cattafesta. Online dynamic mode decomposition for time-varying systems. *SIAM Journal on Applied Dynamical Systems*, 18(3):1586–1609, 2019.

[179] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006.

[180] S. Zuccher, A. Bottaro, and P. Luchini. Algebraic growth in a blasius boundary layer: Nonlinear optimal disturbances. *European Journal of Mechanics-B/Fluids*, 25(1):1–17, 2006.

[181] P. Zwintzscher, F. Goméz, and H. Blackburn. Data-driven control of the turbulent flow past a cylinder. *Journal of Fluids and Structures*, 89:232–243, 2019.

# Appendices

## A.1 Bron-Kerbosch algorithm

Here, we give a short and general overview of the classic Bron-Kerbosch algorithm. There are three sets that will be refined at each call of the algorithm:

1. The set $R$ of the nodes that are shown to form a clique, initialised as the empty set. At each call of the algorithm a node is added to it.

2. The set $P$ of the nodes that are candidates for being in a clique, i.e. all the nodes that could become members of $R$. This set is initialised as all nodes in the graph and in subsequent calls of the function contains the set of nodes that are connected to all the nodes in set $R$.

3. The set $X$ of nodes that the algorithm has already considered in a previously terminated call of the function. The idea of this set is that once all the candidate nodes have been exhausted and the clique in $R$ cannot admit more nodes, the clique will only be returned if the set $X$ is empty. This means that there is no other node that is connected to all nodes in $R$ which is not already a member of $R$.

After each call of the algorithm, the prospect set $P$ and the excluded $X$ sets are updated as in lines 10 and 11 of Algorithm 3. For completeness, Algorithm 3 presents a detailed psuedocode of the classic Bron-Kerbosch algorithm.

**Algorithm 3** Bron-Kerbosch Algorithm

1: $P \leftarrow \mathcal{V}$ ▷ Initialise the prospect set as the set of all vertices
2: $R \leftarrow \emptyset$ ▷ Initialise the connected set as the empty set
3: $X \leftarrow \emptyset$ ▷ Initialise the excluded set as the empty set
4: **function** BronKerbosch(P,R,X)
5: **if** $P = \emptyset \quad \& \quad X = \emptyset$ **then**
6: **Return** $R$
7: **else**
8: **for** $v \in P$ **do**
9: BronKerbosch($R \cup v$,$P \cap N(v)$,$X \cap N(v)$) ▷ $N(v)$ is the set of nodes that are connected to $v$
10: $P \leftarrow P \backslash \{v\}$ ▷ $v$ is eliminated from $P$
11: $X \leftarrow X \cup \{v\}$ ▷ $v$ is added to $X$
12: **end for**
13: **end if**
14: **end function**

Other variants of the Bron-Kerbosch algorithm employ more sophisticated heuristics for choosing which node to consider at each iteration, instead of simply looping through all the modes connected to a specific mode. This limits the number of calls that do not return a maximal clique. In this paper we employ a more computationally-efficient variant of [164], which bounds the number of calculations by $O(3^{n/3})$, where $n$ is the number of graph vertices. However, in many practical applications where the adjacency matrix is sparse, the algorithm converges faster than the theoretical upper bound.

## A.2 An example of a convolutional layer

Here we present an illustrative example that can be useful in giving an intuitive understanding of the operations involved in a convolutional layer. For the input vector

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{pmatrix}, \quad \text{and a cross correlation filter } W = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix},$$

an activation function $\sigma(\cdot)$ is applied to the output of the cross correlation to get

$$f_1(x) = \sigma \begin{pmatrix} w_1x_1 + w_2x_2 + w_3x_3 \\ w_2x_2 + w_3x_3 + w_4x_4 \\ \vdots \\ w_1x_{p-2} + w_2x_{p-1} + w_3x_p \end{pmatrix} + b_1 = y.$$

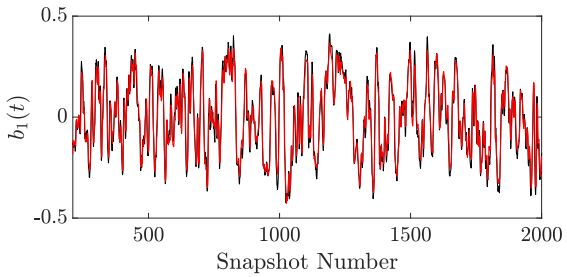
The cross correlation operation is often accompanied by a so called *pooling* operation. An example of a pooling operation is

$$h_1(f_1(x)) = \begin{pmatrix} \max(y_1, y_2) \\ \max(y_3, y_4) \\ \vdots \\ \max(y_{p-n}, y_{p-n+1}) \end{pmatrix}$$

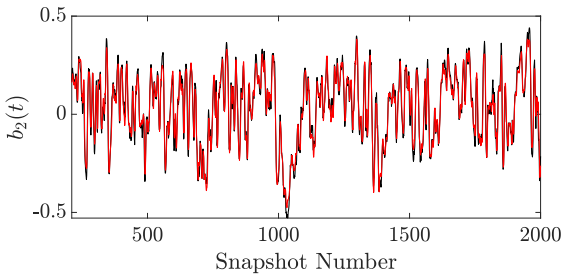
where the output vector reduces the input's dimension by a factor of 2.

## A.3 Encoded states

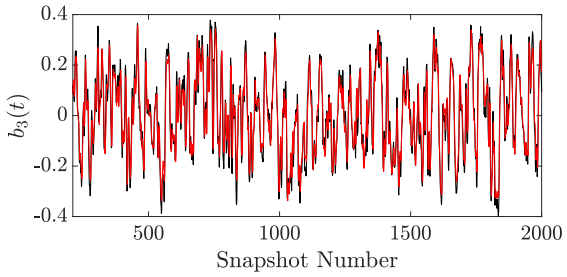
The encoded time series for the flow past an axisymmetric bluff body can be seen in Figure 6.3.



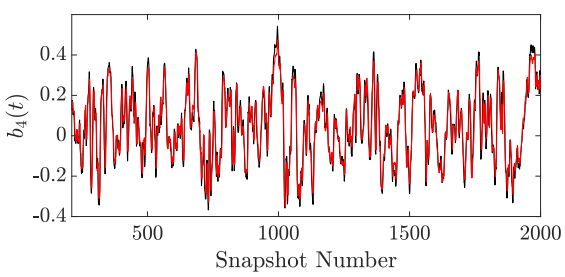

(a) $\tilde{b}_1(t)$



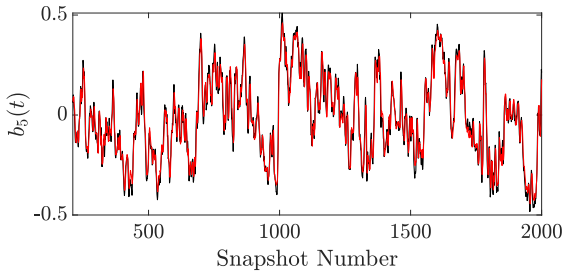

(b) $\tilde{b}_2(t)$



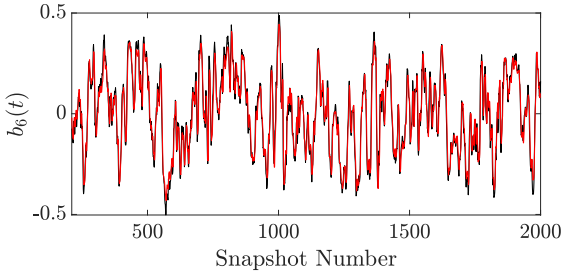

(c) $\tilde{b}_3(t)$



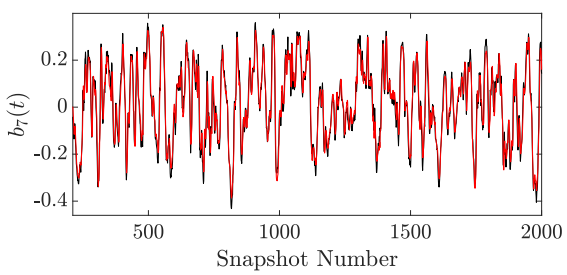

(d) $\tilde{b}_4(t)$



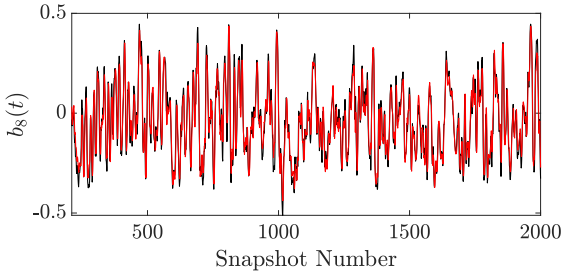

(e) $\tilde{b}_5(t)$



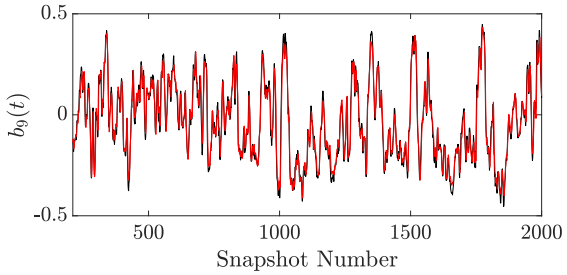

(f) $\tilde{b}_6(t)$



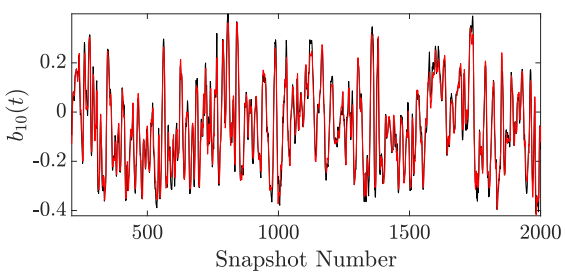

(a) $\tilde{b}_7(t)$



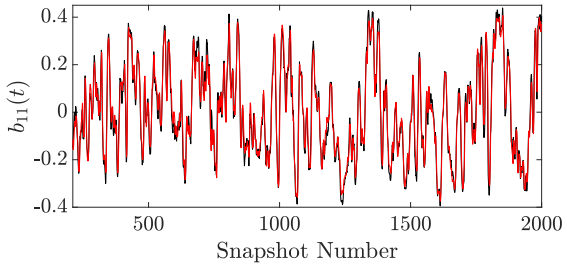

(b) $\tilde{b}_8(t)$



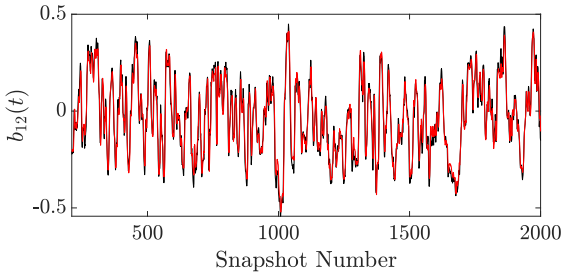

(c) $\tilde{b}_9(t)$



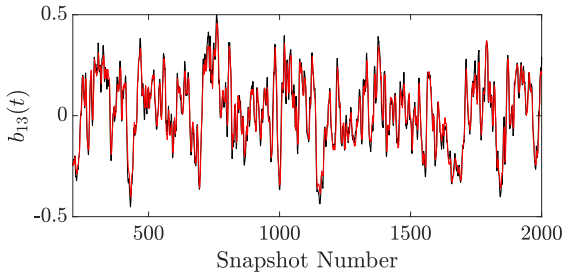

(d) $\tilde{b}_{10}(t)$



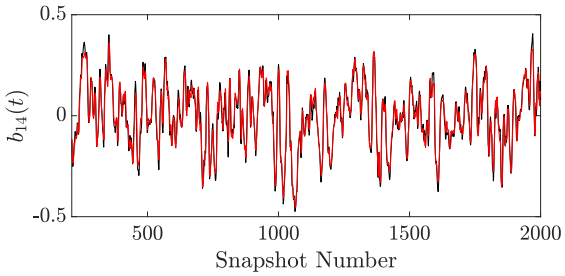

(e) $\tilde{b}_{11}(t)$



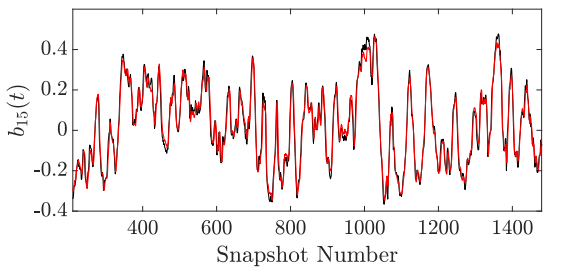

(f) $\tilde{b}_{12}(t)$




(g) $\tilde{b}_{13}(t)$




(h) $\tilde{b}_{14}(t)$




(i) $\tilde{b}_{15}(t)$

Figure 6.3: extracted time series $b_i(t)$ and forecast $\tilde{b}_i(t)$