



**UNIVERSIDADE ESTADUAL DE CAMPINAS**

Faculdade de Engenharia Mecânica

**Hugo Felipe da Silva Lui**

***Construction of reduced order models for fluid flows using deep neural networks***

***Construção de modelos de ordem reduzida para escoamento de fluidos usando rede neurais profundas***

CAMPINAS

2019

Hugo Felipe da Silva Lui

*Construction of reduced order models for fluid flows using deep neural networks*

*Construção de modelos de ordem reduzida para escoamento de fluidos usando rede neurais profundas*

Dissertation presented to the School of Mechanical Engineering of the University of Campinas in partial fulfillment of the requirements for the degree of Master in Mechanical Engineering, in the area of Thermal and Fluids.

Dissertação apresentada à Faculdade de Engenharia Mecânica da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Mecânica, na Área de Térmica e Fluidos.

**Orientador: William Roberto Wolf**

ESTE TRABALHO CORRESPONDE À VERSÃO FINAL DA DISSERTAÇÃO DEFENDIDA PELO ALUNO HUGO FELIPPE DA SILVA LUI, E ORIENTADA PELO PROF. DR. WILLIAM ROBERTO WOLF.

**Assinatura do Orientador**

---

Campinas  
2019

Ficha catalográfica  
Universidade Estadual de Campinas  
Biblioteca da Área de Engenharia e Arquitetura  
Rose Meire da Silva - CRB 8/5974

L968c Lui, Hugo Felipe da Silva, 1993-  
Construction of reduced order models for fluid flows using deep neural networks / Hugo Felipe da Silva Lui. – Campinas, SP : [s.n.], 2019.

Orientador: William Roberto Wolf.  
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica.

1. Fluidodinâmica computacional. 2. Aprendizado de máquina. I. Wolf, William Roberto, 1980-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Mecânica. III. Título.

Informações para Biblioteca Digital

**Título em outro idioma:** Construção de modelos de ordem reduzida para escoamento de fluidos usando rede neurais profundas

**Palavras-chave em inglês:**

Computational fluid dynamics

Machine learning

**Área de concentração:** Térmica e Fluídos

**Titulação:** Mestre em Engenharia Mecânica

**Banca examinadora:**

William Roberto Wolf [Orientador]

Anderson de Rezende Rocha

Alberto Costa Nogueira Junior

**Data de defesa:** 19-07-2019

**Programa de Pós-Graduação:** Engenharia Mecânica

**Identificação e informações acadêmicas do(a) aluno(a)**

- ORCID do autor: <https://orcid.org/0000-0001-7357-7909>

- Currículo Lattes do autor: <http://lattes.cnpq.br/5807484454104406>

**UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA MECÂNICA  
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA  
MECÂNICA  
DEPARTAMENTO DE TERMICA E FLUIDOS**

**DISSERTAÇÃO DE MESTRADO**

***Construction of reduced order models for fluid flows using deep neural networks***

***Construção de modelos de ordem reduzida para escoamento de fluidos usando rede neurais profundas***

Autor: Hugo Felipe da Silva Lui

Orientador: William Roberto Wolf

A Banca Examinadora composta pelos membros abaixo aprovou esta dissertação:

---

Prof. Dr. William Roberto Wolf- presidente FEM/UNICAMP

---

Prof. Dr. Anderson de Rezende Rocha - IC/UNICAMP

---

Dr. Alberto Costa Nogueira Junior - IBM - Research Laboratories

A Ata da defesa com as respectivas assinaturas dos membros encontra-se no processo de vida acadêmica do aluno.

Campinas, 19 de Julho de 2019.

## Agradecimentos

Primeiramente, gostaria de agradecer ao CNPq pelo suporte concedido na forma de bolsa de mestrado por meio do Processo No. 131988/2018-2. Dedico este trabalho à minha mãe, Isamar da Silva, e à minha família, que me apoiaram durante toda a minha vida. Também agradeço aos meus amigos desde a época de escola: Fernando, Eberthyn e Douglas, e aos amigos de faculdade: Paulo (Teco), Schiavon e Yan por todos esses anos de amizade. Um agradecimento especial ao Muk, o cachorro sapeca, que veio alegrar a nossa família.

Gostaria de agradecer os meus amigos no CFD lab: Tulio, Vitão, Diogo, Miotto, Gabriel, Cristiano, Brener e Schiavo. Tanto pelos momentos hilários no Lab, como as resenhas no toca do tatu (medalhão de bacon), marambar (Lokal) e poporale (pizza toscana), além dos churrascos. Aprendi bastante com essa galera. A ajuda deles foi imprescindível para a conclusão deste trabalho. Também gostaria de agradecer a turma do futebol de segunda, por todas as peladas e risadas.

Gostaria de agradecer ao meu orientador, William Roberto Wolf, seus ensinamentos foram de suma importância para a conclusão deste trabalho. O Wolf mudou a minha percepção sobre pesquisa, me abriu os olhos para campos de trabalho que eu não havia ideia de que existiam. Sua mente aberta para explorar novas áreas de pesquisa, além da interdisciplinaridade são exemplos que levarei para a vida.

Gostaria também de agradecer ao professor Anderson de Rezende Rocha e ao Dr. Alberto Costa Nogueira Junior pela discussões construtivas a este trabalho. Destaco também a todos os professores que tive durante a graduação e pós-graduação, agradeço por todos os ensinamentos transmitidos durante a minha vida acadêmica.

*"Chore no começo para sorrir no fim."*

*Marta*

## Resumo

Uma metodologia numérica para construção de modelos de ordem reduzida, ROMs, de escoamentos de fluidos através da combinação de decomposição modal de escoamentos e análise de regressão é apresentada. A decomposição espectral ortogonal própria, SPOD, é aplicada para reduzir a dimensionalidade do modelo e, ao mesmo tempo, filtrar os modos temporais de POD. A etapa de regressão é realizada por uma rede neural profunda, DNN, para obter um sistema de equações diferenciais ordinárias, EDOs, que representam a dinâmica dos modos temporais de POD. A metodologia numérica é implementada em um contexto semelhante ao algoritmo de identificação esparsa de sistemas não-lineares, SINDy. Uma discussão sobre a otimização dos hiperparâmetros da DNN é fornecida para obter os melhores ROMs e uma avaliação desses modelos é apresentada para um oscilador não linear e um escoamento compressível ao redor de um cilindro. Em seguida, o método é testado na reconstrução de um escoamento turbulento obtido por uma simulação de grandes escalas de um aerofólio em condição de estol dinâmico. O modelo de ordem reduzida é capaz de capturar a dinâmica do vórtice de estol sobre o extradorso do aerofólio. Para os casos analisados, a abordagem numérica permite a predição dos campos de escoamento além da janela de treinamento, utilizando incrementos de tempo maiores que os empregados pelo modelo de ordem completa. Demonstrou-se também a robustez dos ROMs obtidos através de redes neurais profundas por meio de uma comparação com os métodos de regressão esparsa. A abordagem usando DNNs é capaz de aprender características transientes do escoamento, além de apresentar previsões de longo prazo mais acuradas e estáveis em comparação com o algoritmo SINDy.

**Palavras-chave:** Modelos de ordem reduzida, redes neurais profundas, decomposição espectral ortogonal própria, dinâmica dos fluidos computacional

# Abstract

A numerical methodology for construction of reduced-order models (ROMs) of fluid flows through the combination of flow modal decomposition and regression analysis is presented. Spectral proper orthogonal decomposition (SPOD) is applied to reduce the dimensionality of the model and, at the same time, filter the proper orthogonal decomposition (POD) temporal modes. The regression step is performed by a deep feedforward neural network (DNN), in order to obtain a system of ordinary differential equations (ODEs) which represents the dynamics of POD temporal modes. The current framework is implemented in a context similar to the sparse identification of nonlinear dynamics algorithm (SINDy). A discussion on the optimization of the DNN hyperparameters is provided for obtaining the best ROMs and an assessment of these models is presented for a canonical nonlinear oscillator and the compressible flow past a cylinder. Then, the method is tested on the reconstruction of a turbulent flow computed by a large eddy simulation of a plunging airfoil under dynamic stall. The reduced-order model is able to capture the dynamics of the leading edge stall vortex and the subsequent trailing edge vortex. For the cases analyzed, the numerical framework allows the prediction of the flowfield beyond the training window using larger time increments than those employed by the full-order model. We also demonstrate the robustness of the current ROMs constructed via deep feedforward neural networks through a comparison with sparse regression. The DNN approach is able to learn transient features of the flow and presents more accurate and stable long-term predictions compared to the SINDy algorithm.

**Keywords:** Reduced-order models, deep neural networks, spectral proper orthogonal decomposition, computational fluid dynamics

## List of Figures

3.1	Schematic of the SINDy algorithm (extracted from Brunton <i>et. al</i> (2016)).	32
3.2	Schematic of the current method used for construction of reduced-order models via DNNs and flow modal decomposition. . . . .	33
3.3	An example of a deep feedforward network with $N$ inputs, several hidden layers (HL), and one output layer with $N$ outputs. . . . .	38
3.4	Cost function for different models. . . . .	44
4.1	Comparison between true model (solid black line) and ROM prediction (dashed cyan line) for nonlinear oscillator. . . . .	47
4.2	Contours of density, $t = 410$ . . . . .	49
4.3	Contours of $\mathbf{x}$ -momentum, $t = 410$ . . . . .	50
4.4	Contours of $\mathbf{z}$ -vorticity, $t = 410$ . . . . .	50
4.5	Contours of divergence of velocity, $t = 410$ . . . . .	51
4.6	Fluctuation time history of density. . . . .	52
4.7	Fluctuation time history of $x$ -momentum. . . . .	52
4.8	Fluctuation time history of $y$ -momentum. . . . .	52
4.9	Reconstruction of POD temporal modes for transient solution of incompressible flow past a cylinder. . . . .	53
4.10	Iso-surfaces of Q criterion colored by pressure at different instants of the plunge motion for the fourth cycle. The ROM was trained using only the first 2 cycles. . . . .	55
4.11	Characteristics of POD and SPOD temporal modes for 3D flow past airfoil under plunging motion. . . . .	56
4.12	Density contours at several phase angles for the fifth plunging cycle. . . . .	59
4.13	Contours of $x$ -component of momentum at several phase angles for the fifth plunging cycle. . . . .	60

4.14	Fluctuation time histories computed by the FOM and ROM for probe locations in the proximity of the leading edge (left column) and trailing edge (right column). . . . .	61
4.15	Reconstruction of SPOD temporal modes using the current DNN approach and the original SINDy algorithm with LASSO. . . . .	62
4.16	Reconstruction of SPOD temporal modes using different finite difference schemes and time steps. . . . .	63
4.17	Phase-averaged aerodynamic coefficients computed by Visbal (2011), the current LES (FOM), and the DNN approach with POD norms based on pressure (P) and kinetic energy (KE). . . . .	64

## List of Tables

3.1	Effects of hyperparameters on model performance. . . . .	42
4.1	Model generation parameters for ROM of nonlinear oscillator . . . . .	47
4.2	Best set of hyperparameters for ROM of nonlinear oscillator . . . . .	47
4.3	Model generation parameters for ROM of compressible flow past a cylinder	48
4.4	Best set of hyperparameters for ROM of compressible flow past a cylinder	48
4.5	Model generation parameters for ROM of plunging airfoil in deep dynamic stall . . . . .	54
4.6	Best set of hyperparameters for ROM of plunging airfoil in deep dynamic stall (3D flow) . . . . .	55
4.7	Best set of hyperparameters for ROM of plunging airfoil in deep dynamic stall (spanwise-averaged flow) . . . . .	57

# List of Acronyms

## Latin characters

$a$	POD temporal modes;
$\dot{a}$	Derivative of the POD temporal modes;
$A$	Matrix of the POD temporal modes;
$\dot{A}$	Matrix of the derivative of the POD temporal modes;
$b$	Biases;
$c_\infty$	Freestream sound speed;
$c_p$	Gas specific heat at constant pressure;
$c_v$	Gas specific heat at constant volume;
$C$	POD correlation matrix;
$\tilde{C}$	Filtered POD correlation matrix;
$E$	Total energy per unit of mass;
$e$	Internal energy per unit of volume;
$f$	Quantity of the spatial discretization scheme;
$f_i$	External force term;
$f_{snap}$	Number of snapshots used in the SPOD filter;
$F$	Nonlinear operator;
$g_{ij}$	Components of the covariant metric tensors;
$g^{ij}$	Components of the contravariant metric tensor;
$\sqrt{g}$	Jacobian of the covariant metric tensor;
$g_k$	Filter function;
$G$	Activation function matrix;
$\left\{ \begin{smallmatrix} i \\ jk \end{smallmatrix} \right\}$	Christoffel symbols of the second kind;
$h$	Frame position (Cross-stream motion of the plunging airfoil);
$h_o$	Motion amplitude;
$\dot{h}$	Frame velocity;
$\ddot{h}$	Frame acceleration;
$k$	Reduced frequency;
$M$	Mach number;
$N$	Number of modes;

$N_p$	Number of grid points;
$N_t$	Number of snapshots;
$N_f$	Size of the filter window;
$p$	Pressure;
$Pr$	Prandtl number;
$q_i$	$i$ -th component of the heat flux vector;
$\mathbf{q}$	Vector of flow variables;
$\bar{\mathbf{q}}$	Mean flow vector;
$\mathbf{q}'$	Fluctuation vector;
$\mathbf{Q}$	Training data;
$R$	Gas constant;
$Re$	Reynolds number;
$S$	Sutherland temperature;
$t$	Time;
$T$	Temperature;
$T_\infty$	Reference temperature;
$u_i$	$i$ -th component of the velocity vector;
$u^i$	$i$ -th component of the contravariant velocity vector;
$x_i$	$i$ -th direction coordinate;
$\mathbf{x}$	Spatial coordinate vector;
$\mathbf{W}$	Weights;

## Greek characters

$\alpha$	Effective angle of attack;
$\alpha_o$	Static angle of attack;
$\gamma$	Ratio of specific heats;
$\delta_{ij}$	Kronecker's delta;
$\Delta$	Spacing between nodes;
$\Delta t$	Time step;
$\kappa$	Thermal conductivity;
$\lambda$	POD engenvalues;
$\mu$	Dynamic viscosity coefficient;
$\mu_\infty$	Dynamic viscosity coefficient at the reference temperature;
$\xi^i$	$i$ -th component of the curvilinear coordinate system;
$\rho$	Fluid density;
$\rho_\infty$	Freestream density;
$\phi$	POD spatial modes;
$\Phi$	Matrix of the POD spatial modes;
$\Omega$	Fluid region;
$\psi$	Angular position in the plunging cycle;
$\Theta$	Matrix of features;
$\sigma$	Activation fuctions;
$\tau_{ij}$	Viscous stress tensor;

## Abbreviations

AIC	Akaike's information criterion
BIC	Bayesian information criterion
CFD	Computational Fluid Dynamics
CFL	Courant-Friedrichs-Lewy number
CNN	Convolutional neural network
DNN	Deep neural network
DNS	Direct Numerical Simulation
DMD	Dynamic mode decomposition
ELU	Exponential linear unit
FNN	Feedforward neural network
FOM	Full-order model
LASSO	Least absolute shrinkage and selection operator
LES	Large-eddy Simulation
LEV	Leading-edge vortex
LCS	Lagrangian coherent structures
LSPG	Least-squares Petrov-Galerkin
LSTM	Long short-term memory
MAE	Mean absolute error
ODE	Ordinary differential equation
POD	Proper orthogonal decomposition
PDE	Partial differential equation
ROM	Reduced-order model
RANS	Reynolds-averaged Navier-Stokes
RELU	Rectified linear unit
RK4	Runge-Kutta fourth-order
RNN	Recurrent neural network
SINDy	Sparse regression of nonlinear dynamics
SPOD	Spectral proper orthogonal decomposition
SVD	Singular value decomposition
TEV	Trailing-edge vortex

# Contents

<b>1</b>	<b>INTRODUCTION</b>	<b>18</b>
1.1	Motivation and previous work . . . . .	18
1.2	Machine learning in fluid mechanics . . . . .	20
1.3	Overview and accomplishments . . . . .	21
1.4	Work organization . . . . .	23
<b>2</b>	<b>FULL ORDER MODEL</b>	<b>24</b>
2.1	Governing equations of compressible flows . . . . .	24
2.2	Numerical methodology for compressible flow simulations . . . . .	26
2.2.1	The contravariant form of the Navier-Stokes equations . . . . .	26
2.2.2	Spatial discretization . . . . .	27
2.2.3	Time integration . . . . .	28
<b>3</b>	<b>REDUCED-ORDER MODEL</b>	<b>30</b>
3.1	Sparse regression of non linear dynamics (SINDy) . . . . .	30
3.2	Numerical framework of proposed approach . . . . .	32
3.2.1	Flow modal decomposition . . . . .	33
3.2.2	Regression step . . . . .	37
3.2.3	Hyperparameter Optimization . . . . .	40
3.3	Data-driven ROMs . . . . .	44
<b>4</b>	<b>RESULTS</b>	<b>46</b>
4.1	Nonlinear oscillator . . . . .	46
4.2	Flow past a cylinder . . . . .	47
4.3	Deep dynamic stall of plunging airfoil . . . . .	53
<b>5</b>	<b>CONCLUSIONS AND RECOMMENDATIONS</b>	<b>65</b>

5.1 Summary . . . . .	65
5.2 Recommendations for future work . . . . .	67
<b>References</b>	<b>68</b>
<b>APPENDIX A : TRIDIAGONAL SYSTEMS</b>	<b>78</b>
<b>APPENDIX B : PENTADIAGONAL SYSTEMS</b>	<b>79</b>
<b>APPENDIX C : LOW-STORAGE RUNGE-KUTTA</b>	<b>81</b>

# 1 INTRODUCTION

## 1.1 Motivation and previous work

Current supercomputers allow the application of high fidelity numerical simulations of turbulent flows over large-scale industrial configurations. The results from these simulations certainly improve the understanding of complex physical phenomena such as mixing enhancement, drag reduction, heat transfer and noise generation, to name a few. Such simulations may lead to discretizations with billions of degrees of freedom in order to resolve the energetically relevant spatial and temporal flow scales. In these cases, data is generally obtained for long periods using small time steps to compute meaningful converged statistics of the flow with sufficient accuracy.

The analysis of unsteady flows by time-resolved simulations and experiments require the acquisition and treatment of large datasets. In recent years, data-driven algorithms have been developed and applied to perform statistical post-processing of such datasets of unsteady fluid flows allowing the investigation of complex physical mechanisms in turbulent flows and improving their analyses. For example, one can cite techniques of flow modal decomposition such as proper orthogonal decomposition (POD) and its variations (Lumley, 1967; Sieber *et. al*, 2016; Ribeiro and Wolf, 2017; Towne *et. al*, 2018), dynamic mode decomposition (DMD) and variations (Schmid, 2010; Tu *et. al*, 2014; Clainche and Vega, 2017), Lagrangian coherent structures (LCS) (Haller, 2015; Green *et. al*, 2007; Rockwood *et. al*, 2016) and resolvent analysis (Luhar *et. al*, 2014; Sharma *et. al*, 2016), among others. Such techniques applied in the context of fluid flows are summarized in a review by Taira *et. al* (2017). Recently, a sequel to this previous paper was published by Taira *et. al* (2019), in which the applications and perspectives of flow modal decomposition techniques are discussed.

The previous techniques of modal analysis can also be employed for the construction of reduced-order models (ROMs) which are appealing since they can be used in the preliminary stages of design due to their inherent reduction in the computational costs compared to large-scale simulations. Such techniques should also be useful in the context of optimization analyzes and studies of flow control which bring potential benefits for energy and transportation. For example, increasing lift and reducing drag due to the transition delay of a laminar boundary layer into a turbulent one, and flow separation

control, would improve the energetic efficiency of aircraft, road vehicles, trains and ships (Brunton and Noack, 2015). Since transportation accounts for a large percentage of global energy consumption, a small enhancement in efficiency would have a significant impact on worldwide economy and emissions of air pollutants (Kim, 2011).

During the last decades, significant developments have been made in the field of feedback flow control of fluids. For example, one can cite studies on flow control for wall flows (Moin and Bewley, 2001; Lumley and Blossey, 1998), for turbulent mixing (Dimotakis, 2005; Aamo *et. al.*, 2003), for bluff bodies (Choi *et. al.*, 2008), for cavity flows (Rowley and Williams, 2006) and for actuators (Cattafesta and Sheplak, 2011; Koizumi *et. al.*, 2018; Rabault *et. al.*, 2019). In most applications, the goal is to design an active flow control strategy to alter the flow in some way, for instance, to control flow separation over aerodynamic bodies or delay transition to turbulent flow in the boundary layer. Although these closed-loop control studies are successful, they are applied for simple geometries. The high dimensionality and computational cost of high fidelity numerical simulations of turbulent flows over large-scale industrial configurations signify that such simulations are unfeasible for real-time feedback flow control (Brunton and Noack, 2015). To address this issue, reduced-order models must be employed as low-dimensional approximation models to these large-scale simulations for prior testing of flow control strategies.

In order to be employed for flow control and optimization applications, ROMs should be able to reproduce the main physical aspects of the full scale models. Several ROM techniques have been discussed in the literature such as Galerkin projection (Rowley *et. al.*, 2004), least-squares Petrov-Galerkin projection (Carlberg *et. al.*, 2011), eigensystem realization analysis (Juang and Pappa, 1985) and sparse regression of nonlinear dynamics (Brunton *et. al.*, 2016). The previous techniques can be used to reduce the original sets of partial differential equations (PDEs) to sets of ordinary differential equations (ODEs). Sparse regression has also been applied for discovering sets of partial differential equations through spatio-temporal data collection (Rudy *et. al.*, 2017). These techniques have mostly been applied for canonical problems and their application to more complex turbulent flows is still a challenging task.

In some cases, ROMs constructed using some of the above techniques may exhibit unstable behavior (Carlberg *et. al.*, 2011). For example, ROMs developed based on Galerkin projection employ POD to rewrite the Navier-Stokes equations as sets of dynamical systems for the evolution of POD temporal modes. Due to the basis truncation typical of such methods, numerical instabilities can be created from imbalance in the turbulent kinetic energy budget in the ROM. This issue may be addressed using turbulence models (Cazemier *et. al.*, 1998; Östh *et. al.*, 2014; Protas *et. al.*, 2015), however, this

approach can destroy consistency between the original PDEs and the ODE system of the ROM. Other alternatives have been also proposed to deal with this problem via minimal rotation of the projected subspace (Balajewicz *et. al*, 2016). This approach is able to account for the contribution of the truncated modes while keeping the consistency between the full- and reduced-order models. Recently, San and Maulik (2018) employed machine learning via neural networks to compute optimal coefficients for an eddy viscosity model which is able to stabilize a POD-Galerkin ROM.

The application of reduced-order modelling techniques in large-scale CFD problems is limited to few cases in literature. Carlberg *et. al* (2013) demonstrated the ability of the Petrov-Galerkin projection associated with gappy POD (Everson and Sirovich, 1995) to reduce the dimension and complexity of an incompressible turbulent flow around the Ahmed body. In another work, Carlberg *et. al* (2017) applied least-squares Petrov-Galerkin projection (LSPG) to generate a reduced-order model of a turbulent compressible flow past an open cavity. In this study, the authors also compared the accuracy and stability of Galerkin and least-squares Petrov-Galerkin approaches. They have shown that LSPG techniques generate stable ROM solutions for turbulent compressible flow problems where standard Galerkin techniques have failed. However, accuracy is a problem of the LSPG method since phase mismatches are often found in numerical solutions.

## 1.2 Machine learning in fluid mechanics

Machine learning is a field that has applications in several areas from data classification to pattern recognition and nonlinear function approximation. Several groups have applied machine learning for investigations concerning fluid flows. Ling and Templeton (2015) used support vector machine, decision trees and random forest to classify regions of high uncertainty in RANS calculations. Ling *et. al* (2016) presented a novel deep neural network (DNN) architecture to improve RANS turbulence models. In a similar context, Wang *et. al* (2017) developed a machine learning approach based on random forests for predicting discrepancies in Reynolds stresses obtained by RANS calculations. Ströfer *et. al* (2019) employed convolutional neural networks (CNNs), a machine learning technique well suited for image pattern recognition, to identify features in fluid flows. Fukami *et. al* (2019) employed CNNs to reconstruct unsteady laminar and turbulent flows from spatially low-resolution data. Koizumi *et. al* (2018) and Rabault *et. al* (2019) presented the first applications of deep reinforcement learning to perform active flow control.

Machine learning is also a natural candidate for the development of ROMs of large-scale dynamical systems, typical of numerical simulations of unsteady flows. A discussion on the application of deep learning in the context just described was presented by Kutz (2017). Recently, several authors have proposed algorithms for the prediction of high-dimensional complex dynamical systems using neural networks and their variants (Rudy *et. al*, 2018; San and Maulik, 2018; Pan and Duraisamy, 2018; Wan *et. al*, 2018; Vlachas *et. al*, 2018). Rudy *et. al* (2018) presented a methodology that is able to learn the dynamics of a particular system and estimate the noise from measurements using feedforward neural networks (FNN). Pan and Duraisamy (2018) performed a comparison between FNNs and sparse regression for modeling of dynamical systems and demonstrated the benefits of the former in terms of adaptability. They computed the Frobenius norm of the Jacobian of the neural network as the regularization term of the cost function to improve the accuracy and robustness of the framework.

Wan *et. al* (2018) developed a ROM methodology capable of modeling extreme events occurring in dynamical systems. These authors employed a long short-term memory (LSTM) approach to regularize a recurrent neural network (RNN) which obtains the complementary dynamics of a non-linear Galerkin projection of the dynamical system. Vlachas *et. al* (2018) combined a LSTM neural network with a mean stochastic model to propose a data-driven algorithm which has desirable short and long-term prediction capabilities. Raissi *et. al* (2019) introduced the physics-informed neural networks, a framework for solving and discovering nonlinear partial differential equations. They use automatic differentiation techniques, same as employed in the back-propagation algorithm, to have a direct connection with the physics of the problem by taking their derivatives with respect to space and time where the physics is described by partial differential equations. They also observed that this approach introduces a regularization mechanism in the neural network.

### 1.3 Overview and accomplishments

The goal of the current study is to develop a data-driven methodology for construction of reduced-order models for unsteady compressible flows. The numerical approach combines flow modal decomposition via spectral proper orthogonal decomposition (SPOD) (Sieber *et. al*, 2016) and regression analysis using DNNs. The framework is implemented in a context similar to that of the sparse identification of nonlinear dynamics (SINDy) algorithm (Brunton *et. al*, 2016). In order to improve the regression step in the current approach, SPOD is employed to extract a low-dimensional representation of the

full-order model. This method filters the POD temporal modes while preserving the information of the FOM by a energy redistribution of the system to higher POD modes.

The regression step is performed by a DNN in order to obtain a system of ODEs that represents the dynamics of the POD temporal modes. The current method is applied for the construction of ROMs of unsteady compressible flows. First, we test the method for a simple two degree of freedom nonlinear dynamical system. Then, we evaluate the capability of the method to reproduce the compressible flow past a cylinder including its noise generation. In this case, we show that the current DNN approach is also able to reproduce the transient stages of the flow. Finally, the method is tested for a turbulent flow involving dynamic stall of a plunging airfoil. We present a discussion on the optimization of hyperparameters for obtaining the best models for the deep neural networks. For both the cylinder and plunging airfoil cases, a comparison between DNN and sparse regression techniques is presented in terms of their predictive capability. The approach presented in this work allows us to predict the flow field beyond the training window and with larger time increments than those used by the full order model, demonstrating the robustness of the current ROMs constructed via deep feedforward neural networks. The current numerical tool for the construction of reduced-order models via DNNs can be downloaded from <http://cces.unicamp.br/software/>.

The principal contributions and findings of the current study are:

- Development and implementation of a data-driven methodology for construction of reduced-order models for unsteady compressible flows. The current numerical capability allows the construction of reduced-order models of complex turbulent flows. The method is tested for two compressible flow cases: a compressible flow past a cylinder and a turbulent flow involving dynamic stall of a plunging airfoil.
- For the first flow evaluated, the ROM shows an excellent agreement with the FOM, being stable and accurate beyond the training region. In this case, the ROM captures most of the dynamics of the von Karman vortex shedding and its respective sound wave propagation.
- For the turbulent flow involving dynamic stall of a plunging airfoil, the complex flow dynamics exhibit unpaired POD modes with high frequency noise that degrades the training of DNNs. We show how the SPOD approach modifies the POD temporal modes allowing their better identification. Reduced-order models are constructed both for full 3-D and spanwise-averaged flow solutions. In both cases, the ROMs are able to capture the dynamics of the leading edge stall vortex, including its formation, transport and ejection, and of the trailing edge vortex.

- Deep learning is still emerging in the field of fluid mechanics. Therefore, simple questions remain open for researchers in this area as addressed by Kutz (2017): (i) How many layers are necessary for a given data set? (ii) How many nodes at each layer are needed? (iii) Can we actually predict data outside of our training data (iv) How to choose the hyperparameters?. A discussion regarding the optimization of hyperparameters for obtaining the best ROMs via DNNs is provided in this study. We also present a description of the effects of individual hyperparameters on model performance. For the cases analyzed, we also demonstrate that ROMs obtained using DNNs present good long-term predictive capabilities with stable and accurate solutions beyond the training window.
- A substantial portion of this work was published in the international leading journal in the field of fluid mechanics as: Lui, H., and Wolf, W. (2019). “Construction of reduced-order models for fluid flows using deep feedforward neural networks”, *Journal of Fluid Mechanics*, 872, pp. 963-994. doi:10.1017/jfm.2019.358

With this work, we expect to provide a contribution to the development of reduced-order models of fluid flows for flow control and optimization applications.

## 1.4 Work organization

The work is organized as follows: chapter 2 provides the theoretical and numerical formulations for compressible flows simulations. Chapter 3 describes the proposed numerical methodology for construction of reduced-order models of fluid flows. In chapter 4, numerical results for ROMs are presented and compared against those obtained by FOMs. Finally, conclusions of the present work and some suggestions for future investigations are presented in chapter 5.

## 2 FULL ORDER MODEL

In this chapter, the governing equations of fluid mechanics and the numerical formulation to solve these equations are described. First, we introduce the compressible form of the Navier-Stokes equations, which consist of a set of PDEs that govern mass and energy conservation beyond Newton's second law. To simulate an airfoil undergoing a prescribed motion for the dynamic stall case, these equations are solved in a non-inertial frame. Here, the Navier-Stokes equations are presented in contravariant form. Then, the full numerical methodology employed for the compressible simulations is described. Schemes used for spatial and temporal discretization, as well as boundary conditions, are discussed in the following subsections.

### 2.1 Governing equations of compressible flows

The equations governing the unsteady motion of a compressible viscous fluid are known as the Navier-Stokes equations. This set of equations is derived on the basis of the continuum hypothesis and from the conservation principles of mass, momentum and energy. These equations can be written in the following conservative differential form as

$$\begin{aligned}
 \text{Continuity equation:} \quad & \frac{\partial \rho}{\partial t} + \frac{\partial \rho u_j}{\partial x_j} = 0 , \\
 \text{Momentum equations:} \quad & \frac{\partial \rho u_i}{\partial t} + \frac{\partial}{\partial x_j} [\rho u_i u_j + p \delta_{ij} - \tau_{ij}] = \rho f_i , \\
 \text{Energy equation:} \quad & \frac{\partial E}{\partial t} + \frac{\partial}{\partial x_j} [(E + p)u_j - \tau_{ij}u_i + q_j] = \rho f_i u_i ,
 \end{aligned} \tag{2.1}$$

where  $t$  is the time,  $x_j$  is the  $j$ -th direction coordinate,  $\rho$  is the fluid density,  $u_i$  is the  $i$ -th component of the velocity vector,  $p$  is the pressure and  $f_i$  is an external force term. The total energy per unit of volume,  $E$ , the viscous stress tensor,  $\tau_{ij}$ , for a Newtonian fluid, and the heat flux for a fluid obeying Fourier's law,  $q_i$ , are defined by

$$E = \rho \left( e + \frac{1}{2} u_i u_i \right) , \tag{2.2}$$

$$\tau_{ij} = \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right), \quad (2.3)$$

$$q_i = -\kappa \frac{\partial T}{\partial x_i}, \quad (2.4)$$

where  $e$  is the internal energy,  $\mu$  is the dynamic viscosity,  $\delta_{ij}$  is the Kronecker delta,  $\kappa$  is the thermal conductivity of the fluid and  $T$  is the temperature.

At this point, the Navier-Stokes equations are not closed; *i.e.*, there are more unknown variables than available equations. In order to close the system of equations it is necessary to establish a relation between thermodynamic variables ( $p, \rho, T, e$ ) as well as to relate them to the transport properties ( $\mu, \kappa$ ). Assuming the fluid to be a calorically perfect gas, the equation of state is defined by

$$p = \rho RT, \quad (2.5)$$

where  $R$  stands for the gas constant. In a calorically perfect gas, the specific heat at constant volume  $c_v$ , the specific heat at constant pressure  $c_p$ , and the ratio of specific heats  $\gamma$  all remain constant, and the following relations exist

$$e = c_v T, \quad \gamma = \frac{c_p}{c_v}, \quad c_v = \frac{R}{\gamma - 1}, \quad c_p = \frac{\gamma R}{\gamma - 1}. \quad (2.6)$$

The dynamic viscosity,  $\mu$ , can be related to temperature,  $T$ , using kinetic theory. This relations is given by Sutherland's law, and provides accurate results for temperatures ranging from  $100K$  to  $1900K$  (Ames Research Staff, 1953). Sutherland's law can be expressed as

$$\mu = \mu_\infty \left( \frac{T}{T_\infty} \right)^{\frac{3}{2}} \frac{T_\infty + S}{T + S}, \quad (2.7)$$

where  $\mu_\infty$  and  $T_\infty$  are reference values taken from a standard atmosphere condition, and  $S = 110.4K$  for air. The Prandtl number

$$Pr = \frac{c_p \mu}{\kappa} \quad (2.8)$$

is used to determine the thermal conductivity once  $\mu$  is known. For air at standard conditions,  $Pr = 0.72$ .

## 2.2 Numerical methodology for compressible flow simulations

### 2.2.1 The contravariant form of the Navier-Stokes equations

For the following investigations of the flow past a cylinder and the dynamic stall of a plunging SD7003 airfoil, the system dynamics are modeled by the compressible Navier-Stokes equations in two dimensions and three dimensions, respectively. To simulate the airfoil undergoing a prescribed motion, the equations are solved in a non-inertial frame. In this form, source terms emerge from the grid curvature and frame movement. Here, all terms are solved in the full contravariant form to allow the use of a curvilinear coordinate system  $\{\xi^1, \xi^2, \xi^3\}$ . For a frame of reference with varying linear velocity, the equations reduce to

$$\frac{\partial}{\partial t}(\sqrt{g}\rho) + \frac{\partial}{\partial \xi^i}(\sqrt{g}\rho u^i) = 0 , \quad (2.9)$$

$$\frac{\partial}{\partial t}(\sqrt{g}\rho u^i) + \frac{\partial}{\partial \xi^j} \left[ \sqrt{g} (\rho u^i u^j - \tau^{ij} + g^{ij} p) \right] + \left\{ \begin{matrix} i \\ jk \end{matrix} \right\} \sqrt{g} (\rho u^k u^j + g^{jk} p - \tau^{kj}) = \sqrt{g} \rho \ddot{h}^i , \quad (2.10)$$

and

$$\frac{\partial}{\partial t}(\sqrt{g}E) + \frac{\partial}{\partial \xi^j} \left\{ \sqrt{g} \left[ (E + p)u^j - \tau^{ij} g_{ik} u^k - \frac{\mu}{Re Pr} g^{ij} \frac{\partial T}{\partial \xi^i} \right] \right\} = \rho \sqrt{g} (h^j + u^j) g_{ji} \ddot{h}^i . \quad (2.11)$$

The set of equations above represents the continuity, momentum and energy equations. In order to close the system of equations the following relations are employed

$$E = \frac{p}{\gamma - 1} + \frac{1}{2} \rho u^i g_{ij} u^j + \frac{1}{2} \rho \dot{h}^i g_{ij} \dot{h}^i , \quad (2.12)$$

$$\tau^{ij} = \frac{\mu}{Re} \left( g^{jk} u^i{}_{|k} + g^{ik} u^j{}_{|k} - \frac{2}{3} g^{ij} u^k{}_{|k} \right) , \quad (2.13)$$

and

$$h = h_o \sin(kt) . \quad (2.14)$$

Here  $u^i$  the  $i$ -th component of the contravariant velocity vector and the term  $h$  is the frame position (cross-stream motion of the plunging airfoil). The dots represent temporal derivatives of the frame position, i.e., frame velocity and acceleration. In the above equations, physical velocity components are normalized by the freestream speed of sound

$c_\infty$ , density is normalized by the freestream density  $\rho_\infty$ , pressure is normalized by  $\gamma\rho_\infty c_\infty^2$  and temperature by  $(\gamma-1)T_\infty$ . When appropriate, length scales are made non-dimensional by the cylinder diameter or airfoil chord. The aforementioned terms,  $g_{ij}$  and  $g^{ij}$  are the covariant and contravariant metric tensors,  $\sqrt{g}$  is the Jacobian of the covariant metric tensor and  $\left\{ \begin{smallmatrix} i \\ jk \end{smallmatrix} \right\}$  represents the Christoffel symbols of the second kind. Further details regarding the present formulation can be found in Aris (1989).

## 2.2.2 Spatial discretization

The numerical scheme for the spatial discretization of the system of governing equations is a high-resolution sixth-order accurate compact scheme (Nagarajan *et. al.*, 2003) implemented on a staggered grid. For any quantity  $f$ , a tridiagonal system is solved as

$$\alpha f'_{i-1} + f'_i + \alpha f'_{i+1} = b \frac{f_{i+3/2} - f_{i-3/2}}{3\Delta} + a \frac{f_{i+1/2} - f_{i-1/2}}{\Delta}, \quad (2.15)$$

where  $f'$  represents the derivative of  $f$  at node  $i$ ,  $\Delta$  is the spacing between nodes,  $\alpha = 9/62$ ,  $a = \frac{3}{8}(3-2\alpha)$  and  $b = \frac{1}{8}(-1+22\alpha)$ . To minimize errors from unresolved scales and preserve stability of numerical schemes, a sixth-order compact low-pass filter is applied according to

$$\bar{\alpha} \bar{f}_{i-1} + \bar{f}_i + \bar{\alpha} \bar{f}_{i+1} = \bar{a} \bar{f}_i + \frac{\bar{b}}{2}(f_{i+1} + f_{i-1}) + \frac{\bar{c}}{2}(f_{i+2} + f_{i-2}) + \frac{\bar{d}}{2}(f_{i+3} + f_{i-3}). \quad (2.16)$$

In the equation above,  $\bar{f}$  is the filtered quantity,  $\bar{a} = \frac{1}{16}(11 + 10\bar{\alpha})$ ,  $\bar{b} = \frac{1}{32}(15 + 34\bar{\alpha})$ ,  $\bar{c} = \frac{1}{16}(-3 + 6\bar{\alpha})$  and  $\bar{d} = \frac{1}{32}(1 - 2\bar{\alpha})$ . The filtered coefficient  $\bar{\alpha}$  ranges from  $[-0.5, 0.5]$ . The lower limit,  $\bar{\alpha} = -0.5$ , represents to maximum filtering while the filter has no effect at  $\bar{\alpha} = 0.5$ .

Due to the staggered grid, interpolations are necessary to evaluate properties at specific grid locations. To maintain schemes with high-order, a sixth-order interpolation based on finite differences is used according to

$$\tilde{\alpha} \tilde{f}_{i-1} + \tilde{f}_i + \tilde{\alpha} \tilde{f}_{i+1} = \frac{\tilde{b}}{2}(f_{i+3/2} + f_{i-3/2}) + \frac{\tilde{a}}{2}(f_{i+1/2} + f_{i-1/2}), \quad (2.17)$$

where  $\tilde{\alpha} = 3/10$ ,  $\tilde{a} = \frac{1}{8}(9+10\tilde{\alpha})$  and  $\tilde{b} = \frac{1}{8}(6\tilde{\alpha}-1)$ . Additional details on the finite-difference schemes used for derivation, filtering and interpolation can be found in (Lele, 1992; Nagarajan, 2004). A complete description of the numerical schemes at boundary and

near boundary nodes is presented by (Nagarajan, 2004).

Near the far-field boundaries, a numerical sponge is used to damp acoustic waves and, at the inlet and outlet boundaries, a Riemann invariant transformation is implemented as far-field condition. No-slip adiabatic wall boundary conditions are applied along the solid surfaces. Derivatives of inviscid fluxes are obtained by forming fluxes between grid nodes on the staggered grid and differentiating each component. Viscous terms are obtained by first computing the derivatives of primitive variables at their respective locations (see Nagarajan (2004) for details). Components of viscous fluxes are then constructed at each node and differentiated by a second application of the compact scheme. For the study of dynamic stall, the airfoil movement is added through source terms and periodic boundary conditions are applied along the spanwise direction of the airfoil.

### 2.2.3 Time integration

After the spatial discretization, the Navier-Stokes equations become a set of ordinary differential equations that can be expressed in the form

$$\frac{d\mathbf{Q}}{dt} = \mathbf{F}(\mathbf{Q}, t) , \quad (2.18)$$

where  $\mathbf{Q}$  is the vector of flow variables. An explicit third-order Runge-Kutta scheme is used for time advancement of the equations in flow regions far away from solid boundaries. The third-order Runge-Kutta scheme is given by

$$\begin{aligned} \mathbf{Q}^{n+1/3} &= \mathbf{Q}^n + \frac{8}{15}\Delta t \mathbf{F}(\mathbf{Q}^n, t^n) \\ \mathbf{Q}^{n+2/3} &= \mathbf{Q}^n + \frac{1}{4}\Delta t \mathbf{F}(\mathbf{Q}^n, t^n) + \frac{5}{12}\Delta t \mathbf{F}(\mathbf{Q}^{n+1/3}, t^{n+1/3}) \\ \mathbf{Q}^{n+1} &= \mathbf{Q}^n + \frac{1}{4}\Delta t \mathbf{F}(\mathbf{Q}^n, t^n) + \frac{3}{4}\Delta t \mathbf{F}(\mathbf{Q}^{n+2/3}, t^{n+2/3}) , \end{aligned} \quad (2.19)$$

where the intermediate time levels are  $t^{n+1/3} = t^n + (8/15)\Delta t$  and  $t^{n+2/3} = t^n + (2/3)\Delta t$ .

The time integration of the fluid equations is carried out by the fully implicit second-order scheme of Beam and Warming (1978) in the near-wall region in order to overcome the time step restriction due to the usual near-wall fine-grid numerical stiffness. The second-order implicit method is given by

$$\frac{3\mathbf{Q}^{n+1} - 4\mathbf{Q}^n + \mathbf{Q}^{n-1}}{2\Delta t} = \mathbf{F}(\mathbf{Q}^{n+1}, t^{n+1}) . \quad (2.20)$$

The right hand side is solved through approximate factorization followed by diagonalization of the implicit matrix in the  $x$  and  $z$  directions. More details about the approximate factorization are presented by Nagarajan (2004).

The low-pass compact filter is applied after each time-step of both schemes. More details about the numerical framework employed can be found in Nagarajan (2004). The numerical tool has been previously validated for several simulations of unsteady compressible flows (Wolf, 2011; Wolf *et. al*, 2012a,b; Ramos *et. al*, 2019).

### 3 REDUCED-ORDER MODEL

This chapter presents the numerical methodologies for construction of reduced-order models (ROMs) of fluid flows through the combination of flow modal decomposition and regression analysis. First, we introduce the sparse identification of non-linear dynamics (SINDy) method developed by Brunton *et. al* (2016). The reduced-order model approach presented in this work is based on the SINDy algorithm; however, in the proposed methodology, SPOD and DNNs are introduced for reduction of model dimensionality and regression, respectively. Spectral proper orthogonal decomposition (SPOD) is applied to reduce the dimensionality of the dynamical system and, at the same time, filter the POD temporal modes. The regression step is performed by a deep feedforward neural network in order to obtain a system of ODEs that represents the dynamics of the POD temporal modes. A discussion on the optimization of the DNN hyperparameters for obtaining the best ROMs is provided. Finally, a discussion of ROMs based on the problem physics and data-driven ROMs is presented at the end of this chapter.

#### 3.1 Sparse regression of non linear dynamics (SINDy)

Recently, Brunton *et. al* (2016) introduced the SINDy algorithm which identifies ordinary differential equations from data. SINDy follows the assumption that there are only a few important terms that define the dynamics of a system, so that the equations are sparse in the space of candidate functions. Sparse regression is then used to determine the features required to accurately reproduce the system dynamics.

Here, we consider a dynamical system of the form

$$\frac{d\boldsymbol{\chi}(t)}{dt} = \mathbf{f}(\boldsymbol{\chi}, t) . \quad (3.1)$$

where  $\boldsymbol{\chi}(t)$  is the state of a system at time  $t$  and the function  $\mathbf{f}(\boldsymbol{\chi}, t)$  represents the features that define the system dynamics. To determine the function  $\mathbf{f}$  from data, we collect a time history of state  $\boldsymbol{\chi}(t)$  and its derivative  $\dot{\boldsymbol{\chi}}(t)$ . The derivatives  $\dot{\boldsymbol{\chi}}(t)$  can be obtained numerically using the data  $\boldsymbol{\chi}(t)$ . The data are stored into two matrices  $\mathbf{X}$  and

$\dot{\mathbf{X}}$ :

$$\mathbf{X} = \begin{bmatrix} \boldsymbol{\chi}^T(t_1) \\ \boldsymbol{\chi}^T(t_2) \\ \vdots \\ \boldsymbol{\chi}^T(t_m) \end{bmatrix} = \begin{bmatrix} \chi_1(t_1) & \chi_2(t_1) & \cdots & \chi_n(t_1) \\ \chi_1(t_2) & \chi_2(t_2) & \cdots & \chi_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \chi_1(t_m) & \chi_2(t_m) & \cdots & \chi_n(t_m) \end{bmatrix}, \quad (3.2)$$

$$\dot{\mathbf{X}} = \begin{bmatrix} \dot{\boldsymbol{\chi}}^T(t_1) \\ \dot{\boldsymbol{\chi}}^T(t_2) \\ \vdots \\ \dot{\boldsymbol{\chi}}^T(t_m) \end{bmatrix} = \begin{bmatrix} \dot{\chi}_1(t_1) & \dot{\chi}_2(t_1) & \cdots & \dot{\chi}_n(t_1) \\ \dot{\chi}_1(t_2) & \dot{\chi}_2(t_2) & \cdots & \dot{\chi}_n(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \dot{\chi}_1(t_m) & \dot{\chi}_2(t_m) & \cdots & \dot{\chi}_n(t_m) \end{bmatrix}, \quad (3.3)$$

where  $m$  is the number of time samples and  $n$  the dimension of the system. Next, we construct a library  $\Theta(\mathbf{X})$  of candidate non linear functions. Each column of  $\Theta(\mathbf{X})$  represents a candidate function for the right-hand side of (3.1). Brunton *et. al* (2016) suggest that  $\Theta(\mathbf{X})$  may consist of constant, polynomial, exponential and trigonometric terms. The choice of candidate functions normally requires some prior knowledge about the dynamical system. An example of  $\Theta(\mathbf{X})$  is shown below:

$$\Theta(\mathbf{X}) = \begin{bmatrix} 1 & \chi_1(t_1) & \chi_1(t_1)\chi_2(t_1) & \cdots & \chi_2^2(t_1) & \cdots & \sin(\chi_n(t_1)) \\ 1 & \chi_1(t_2) & \chi_1(t_2)\chi_2(t_2) & \cdots & \chi_2^2(t_2) & \cdots & \sin(\chi_n(t_2)) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 1 & \chi_1(t_m) & \chi_1(t_m)\chi_2(t_m) & \cdots & \chi_2^2(t_m) & \cdots & \sin(\chi_n(t_m)) \end{bmatrix}. \quad (3.4)$$

Now, one can set up a regression problem to determine the unknown parameters  $\Xi$ .

$$\dot{\mathbf{X}} = \Theta(\mathbf{X})\Xi. \quad (3.5)$$

By using sparse regression, we are able to find what right-hand side terms are small (close to zero), resulting in sparse models. There are a number of algorithms for computing sparse regression. At the present time, the least absolute shrinkage and selection operation (LASSO) (Tibshirani, 1996) is the most popular technique for this type of regression. It performs both feature selection and regularization based on  $\ell_1$  penalty. An alternative algorithm is proposed by Brunton *et. al* (2016), the sequential thresholded least-squares algorithm. In this algorithm, one starts with a least-squares solution for the unknown parameters  $\Xi$  and then threshold all parameters that are smaller than some cutoff value. Next, we compute another least-squares solution for the remaining non-zero parameters. These new parameters are again thresholded, and the sequential thresholded least-squares procedure is continued until some stopping criteria.

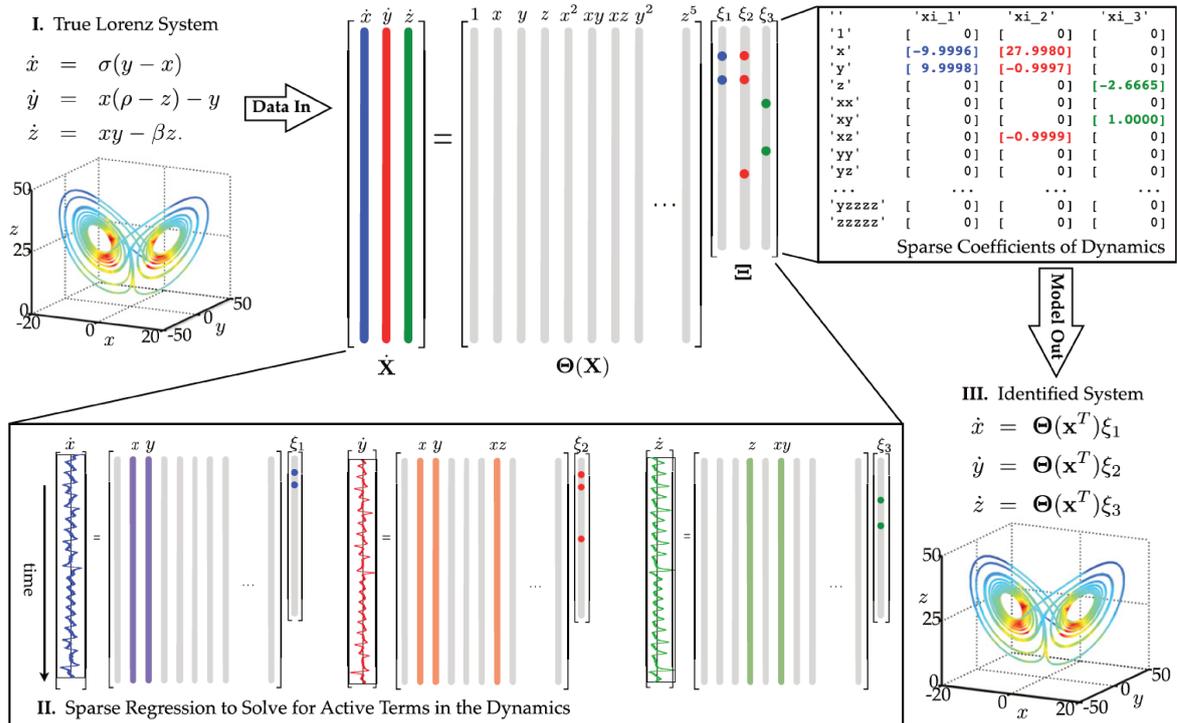


Figure 3.1: Schematic of the SINDy algorithm (extracted from Brunton *et. al* (2016)).

Figure 3.1 shows a schematic of the SINDy algorithm demonstrated for the solution of the Lorenz attractor system. First, data are collected, including a time history of the state  $\mathbf{X}$  and its derivatives  $\dot{\mathbf{X}}$ . Next, a library of candidate functions of the states,  $\Theta(\mathbf{X})$ , is constructed. Then, sparse regression is used to find the fewest terms needed to satisfy Eq. 3.5.

High fidelity numerical simulations of fluid flows may lead to discretizations with millions or billions of degrees of freedom in order to resolve the most energetically relevant scales of the flow. Thus, after spatial discretization of the Navier-Stokes equations, a large system of non linear ODEs results, which can be considered as a high dimensional dynamical system. Here, we generalize the SINDy method to develop a methodology for construction of reduced-order models of fluid flows.

## 3.2 Numerical framework of proposed approach

The present reduced-order modelling approach has the following steps: (i) collection of flow snapshots, (ii) decomposition of data set, (iii) dimensionality reduction of dynamical system, (iv) computation of spatial modes and temporal coefficients using the training data, (v) numerical calculation of derivatives of temporal coefficients, (vi) regression via

deep neural networks to reduce system of PDEs to coupled system of ODEs that represents the dynamics of the temporal coefficients, (vii) spatio-temporal reconstruction of flowfield. A schematic of the current numerical framework can be seen in figure 3.2.

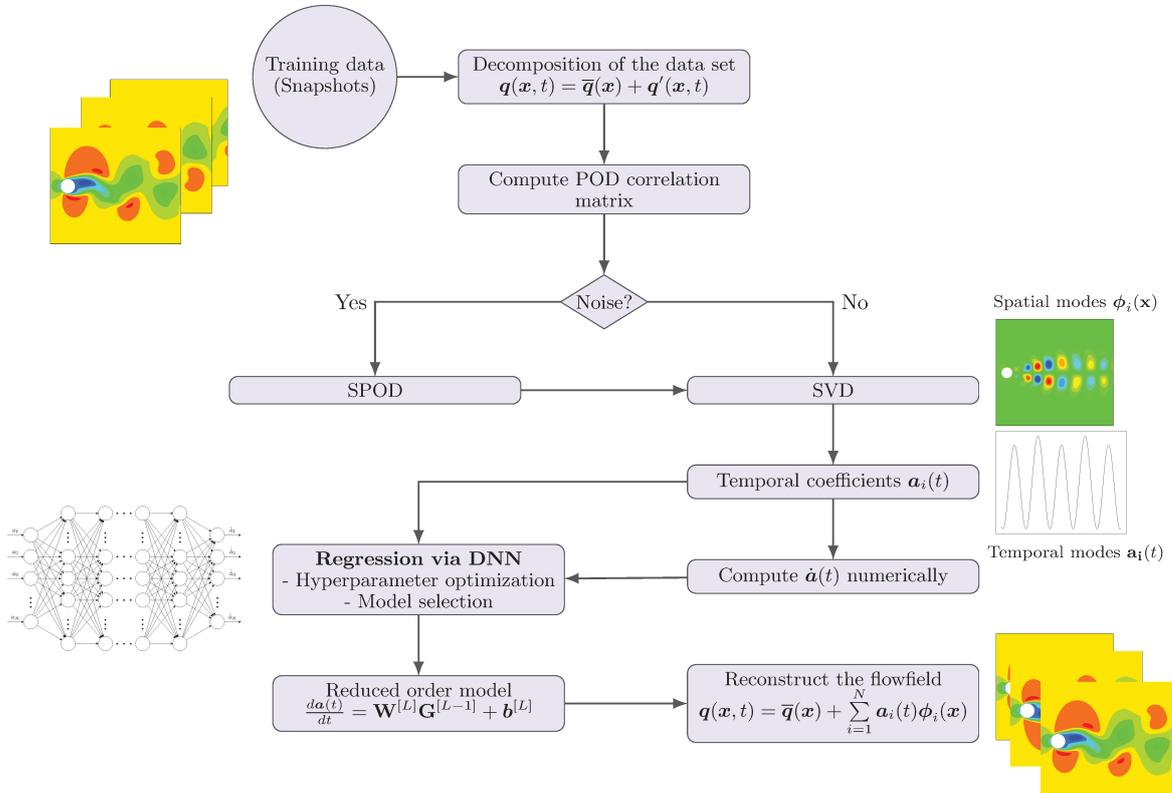


Figure 3.2: Schematic of the current method used for construction of reduced-order models via DNNs and flow modal decomposition.

### 3.2.1 Flow modal decomposition

The equations governing an unsteady three-dimensional compressible flow contain partial derivatives with respect to both the spatial coordinates  $\mathbf{x} = [x \ y \ z]^T$  and time  $t$ . Using the method of lines one can first approximate the spatial derivatives producing a system of nonlinear ordinary differential equations (ODEs). In the most general notation, for each mesh point, these ODEs would be expressed in the form

$$\frac{d\mathbf{q}}{dt} = \mathbf{F}(\mathbf{q}, t), \quad (3.6)$$

where  $\mathbf{F}$  is a nonlinear operator,  $\mathbf{q} = [\rho \ \rho u \ \rho v \ \rho w \ p]^T$  is the vector of flow variables,  $\rho$  is the density,  $u$ ,  $v$  and  $w$  are respectively the  $x$ ,  $y$  and  $z$ -components of the velocity

vector and  $p$  is the pressure.

Here, we consider that a dynamical system given by Eq. 3.6 is solved at each mesh point. To determine the nonlinear operator  $\mathbf{F}$  from data, we follow the ideas of the SINDy algorithm developed by Brunton *et. al* (2016) with some modifications. First, we collect snapshots of the variables which will be our training data. The data set is then arranged into a matrix  $\mathbf{Q}$  as

$$\mathbf{Q} = \begin{bmatrix} \mathbf{q}(\mathbf{x}_1, t_1) & \mathbf{q}(\mathbf{x}_2, t_1) & \dots & \mathbf{q}(\mathbf{x}_{N_p}, t_1) \\ \mathbf{q}(\mathbf{x}_1, t_2) & \mathbf{q}(\mathbf{x}_2, t_2) & \dots & \mathbf{q}(\mathbf{x}_{N_p}, t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{q}(\mathbf{x}_1, t_{N_T}) & \mathbf{q}(\mathbf{x}_2, t_{N_T}) & \dots & \mathbf{q}(\mathbf{x}_{N_p}, t_{N_T}) \end{bmatrix}, \quad (3.7)$$

where  $N_p$  is the number of grid points in the computational domain and  $N_T$  is the number of snapshots. Because of the high dimensionality of the input data  $\mathbf{Q}$ , we first reduce the dimension of the dynamical system using the snapshot POD method (Sirovich, 1986). Proper orthogonal decomposition has been widely applied to obtain optimal bases that represent the most energetic content of the system dynamics with as few basis functions as possible (Lumley, 1967). The snapshot POD approach starts with a decomposition of the vector quantities  $\mathbf{q}(\mathbf{x}, t)$  into the mean flow,  $\bar{\mathbf{q}}(\mathbf{x})$ , and fluctuations,  $\mathbf{q}'(\mathbf{x}, t)$ . The latter can be further expanded into a combination of spatial modes  $\phi_i(\mathbf{x})$  and their temporal coefficients  $\mathbf{a}_i(t)$  for a defined number of  $N$  modes as

$$\mathbf{q}(\mathbf{x}, t) = \bar{\mathbf{q}}(\mathbf{x}) + \mathbf{q}'(\mathbf{x}, t) = \bar{\mathbf{q}}(\mathbf{x}) + \sum_{i=1}^N \mathbf{a}_i(t) \phi_i(\mathbf{x}). \quad (3.8)$$

To calculate the POD correlation matrix of the data set  $\mathbf{Q}$  some specific norm must be used. For an incompressible flow, a kinetic energy norm provides an optimal result, however, for a compressible flow, other norms can be employed (Rowley *et. al*, 2004). Hence, we define a vector  $\boldsymbol{\eta} = [\eta_1 \ \eta_2 \ \eta_3 \ \eta_4 \ \eta_5]^T$  that determines which norm should be used to compute the correlation between two snapshots. For example, a pressure-based norm uses  $\boldsymbol{\eta} = [0 \ 0 \ 0 \ 0 \ 1]^T$  and a kinetic energy norm uses  $\boldsymbol{\eta} = [0 \ 1 \ 1 \ 1 \ 0]^T$ . The correlation between two snapshots is computed using the  $L^2$  inner product. Therefore, considering  $\mathbf{q}'(\mathbf{x}, t_i) = \mathbf{q}'_i$  and  $\mathbf{q}'(\mathbf{x}, t_j) = \mathbf{q}'_j$ , the elements of the correlation matrix  $\mathbf{C}$  are given by

$$C_{ij} = \int_{\Omega} [\eta_1 \rho'_i \rho'_j + \eta_2 (\rho u)'_i (\rho u)'_j + \eta_3 (\rho v)'_i (\rho v)'_j + \eta_4 (\rho w)'_i (\rho w)'_j + \eta_5 p'_i p'_j] d\Omega, \quad (3.9)$$

where  $\Omega$  is the fluid region of interest for the reconstruction and the matrix  $\mathbf{C}$  is of size  $N \times N$ . For the problems studied in this work, we employ norms based on kinetic energy

and pressure. Despite the changes in the POD modes computed for the different norms, we observed that the ROMs obtained by either kinetic energy or pressure norms were similar. In both cases, stable and accurate models could be reconstructed by the DNNs and further comments are provided in the results chapter.

In turbulent flows, the POD temporal modes may contain contributions from several frequencies, including high-frequency noise. For such cases, in order to provide a better identification of the individual modes and to smooth out the temporal coefficients, we employ the spectral proper orthogonal decomposition (SPOD) described by Sieber *et. al* (2016). The SPOD technique is able to filter the temporal modes while preserving the information of the FOM by a redistribution of the energy of the system to higher POD modes. The technique consists of applying a filter function to the POD correlation matrix which results in a matrix  $\tilde{\mathbf{C}}$  with elements given as

$$\tilde{C}_{ij} = \sum_{k=-N_f/2}^{N_f/2} g_k C_{i+k, j+k} . \quad (3.10)$$

Here,  $g_k$  is the filter function and  $N_f$  is the size of the filter window. We consider a periodic temporal series and assume that the correlation matrix is also periodic (see Ribeiro and Wolf (2017) for details). Hence,  $N_f = f_{snap}/N_T$ , where  $f_{snap}$  is the number of snapshots used in the SPOD filter. Following this notation, if we apply 50% of filter to the correlation matrix, it means that we are filtering 50% of the total number of snapshots. Several filter functions can be applied to the POD correlation matrix and their effects are reported by Ribeiro and Wolf (2017). The filter function employed in this work is the box filter represented by  $g_k = 1/(2N_f + 1)$ .

The temporal coefficients  $\mathbf{a}_i = [a_i(t_1) \dots a_i(t_{N_T})]^T$  and the POD eigenvalues  $\lambda_i$  are determined from the filtered correlation matrix  $\tilde{\mathbf{C}}$  as

$$\tilde{\mathbf{C}}\mathbf{a}_i = \lambda_i\mathbf{a}_i . \quad (3.11)$$

Singular value decomposition (SVD) can be employed to compute the eigenvalues  $\lambda_i$  and eigenvectors  $\mathbf{a}_i$  of  $\tilde{\mathbf{C}}$  since the matrix is real symmetric positive-definite. The spatial modes are obtained from the projection of the fluctuation quantities onto the temporal coefficients (see Cordier and Bergmann (2008) for details).

$$\phi_i(\mathbf{x}) = \frac{1}{\lambda_i} \sum_{j=1}^N \mathbf{a}_i(t_j) \mathbf{q}'(\mathbf{x}, t_j) . \quad (3.12)$$

Finally, the temporal coefficients and spatial modes can be stored in matrices  $\mathbf{A}$  and  $\Phi$

as

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}^T(t_1) \\ \mathbf{a}^T(t_2) \\ \vdots \\ \mathbf{a}^T(t_{N_T}) \end{bmatrix} = \begin{bmatrix} a_1(t_1) & a_2(t_1) & \dots & a_N(t_1) \\ a_1(t_2) & a_2(t_2) & \dots & a_N(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ a_1(t_{N_T}) & a_2(t_{N_T}) & \dots & a_N(t_{N_T}) \end{bmatrix}, \quad (3.13)$$

and

$$\Phi = \begin{bmatrix} \phi_1(\mathbf{x}_1) & \phi_1(\mathbf{x}_2) & \dots & \phi_1(\mathbf{x}_{N_p}) \\ \phi_2(\mathbf{x}_1) & \phi_2(\mathbf{x}_2) & \dots & \phi_2(\mathbf{x}_{N_p}) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_N(\mathbf{x}_1) & \phi_N(\mathbf{x}_2) & \dots & \phi_N(\mathbf{x}_{N_p}) \end{bmatrix}, \quad (3.14)$$

where the temporal coefficients are the columns of  $\mathbf{A}$  and the spatial modes are the rows of  $\Phi$ . Hence, the matrix of fluctuation quantities can be written as

$$\mathbf{Q}' = \mathbf{A} \Phi. \quad (3.15)$$

Taking the time derivative of Eq. 3.8, we arrive at the following set of equations

$$\frac{d\mathbf{q}(t)}{dt} = \sum_{i=1}^N \phi_i(\mathbf{x}) \frac{d\mathbf{a}_i(t)}{dt}. \quad (3.16)$$

The last term of Eq. 3.16 represents the temporal evolution of coefficients  $\mathbf{a}_i(t)$  associated with the  $N$  modes retained in the SPOD modal basis. We can express this system of coupled ODEs as

$$\frac{d\mathbf{a}(t)}{dt} = \dot{\mathbf{a}}(t) = \mathbf{F}(\mathbf{a}(t)). \quad (3.17)$$

Next, we compute the derivative  $\dot{\mathbf{a}}(t)$  numerically using the data  $\mathbf{a}(t)$  for each temporal mode. Different numerical schemes are tested for the computation of the temporal derivatives and a discussion of the application of explicit schemes and compact schemes will be provided in the results section. For now, let us consider that the numerical scheme employed for the temporal derivatives is a 10th-order accurate compact scheme (Lele, 1992) which provides high spectral resolution being non-dissipative and low-dispersive. The derivative  $\dot{\mathbf{a}}(t)$  is then obtained as

$$\delta_1 \dot{\mathbf{a}}_{i-2} + \delta_2 \dot{\mathbf{a}}_{i-1} + \dot{\mathbf{a}}_i + \delta_2 \dot{\mathbf{a}}_{i+1} + \delta_1 \dot{\mathbf{a}}_{i+2} = \delta_3 \frac{\mathbf{a}_{i+3} - \mathbf{a}_{i-3}}{6h} + \delta_4 \frac{\mathbf{a}_{i+2} - \mathbf{a}_{i-2}}{4h} + \delta_5 \frac{\mathbf{a}_{i+1} - \mathbf{a}_{i-1}}{2h}. \quad (3.18)$$

In the equation above,  $h$  is the time step and the coefficients of the numerical scheme are set as  $\delta_1 = 1/20$ ,  $\delta_2 = 1/2$ ,  $\delta_3 = 1/100$ ,  $\delta_4 = 101/150$  and  $\delta_5 = 17/12$  for a 10th-order discretization. The system of Eqs. 3.18 written for each temporal mode can be solved as a pentadiagonal linear system for the unknown derivatives  $\dot{\mathbf{a}}(t)$ . The boundary data

points can be computed from the interior points as follow

- at  $i = 4$  and  $i = N_T - 3$

$$\delta_1 = \frac{0.451390625}{9.38146875}, \delta_2 = \frac{4.63271875}{9.38146875}, \delta_3 = 2 \frac{6.66984375}{9.38146875}, \delta_4 = 4 \frac{1.53}{9.38146875}, \delta_5 = 6 \frac{0.015}{9.38146875} .$$

- at  $i = 3$  and  $i = N_T - 2$

$$\delta_1 = \frac{0.2964375}{10.67175}, \delta_2 = \frac{4.7435}{10.67175}, \delta_3 = 2 \frac{7.905}{10.67175}, \delta_4 = 4 \frac{1.23515625}{10.67175}, \delta_5 = 0 .$$

- at  $i = 2$  and  $i = N_T - 1$

$$\delta_1 = 0, \delta_2 = \frac{1}{4}, \delta_3 = \frac{3}{2}, \delta_4 = 0, \delta_5 = 0 .$$

- at  $i = 1$  and  $i = N_T$

$$\begin{aligned} \dot{\mathbf{a}}_1 + 2\dot{\mathbf{a}}_2 &= \frac{1}{h} \left( -\frac{5}{2}\mathbf{a}_1 + 2\mathbf{a}_2 + \frac{1}{2}\mathbf{a}_3 \right) . \\ \dot{\mathbf{a}}_{N_T} + 2\dot{\mathbf{a}}_{N_T-1} &= \frac{1}{h} \left( \frac{5}{2}\mathbf{a}_{N_T} - 2\mathbf{a}_{N_T-1} - \frac{1}{2}\mathbf{a}_{N_T-2} \right) . \end{aligned} \quad (3.19)$$

The derivatives  $\dot{\mathbf{a}}(t)$  are then arranged into a matrix  $\dot{\mathbf{A}}$

$$\dot{\mathbf{A}} = \begin{bmatrix} \dot{\mathbf{a}}^T(t_1) \\ \dot{\mathbf{a}}^T(t_2) \\ \vdots \\ \dot{\mathbf{a}}^T(t_{N_T}) \end{bmatrix} = \begin{bmatrix} \dot{a}_1(t_1) & \dot{a}_2(t_1) & \dots & \dot{a}_N(t_1) \\ \dot{a}_1(t_2) & \dot{a}_2(t_2) & \dots & \dot{a}_N(t_2) \\ \vdots & \vdots & \ddots & \vdots \\ \dot{a}_1(t_{N_T}) & \dot{a}_2(t_{N_T}) & \dots & \dot{a}_N(t_{N_T}) \end{bmatrix} . \quad (3.20)$$

### 3.2.2 Regression step

Once the matrix of temporal derivatives is computed, we can set up a regression problem to find the weights  $\mathbf{W}$  and biases  $\mathbf{b}$  that determine the function  $F(\mathbf{a}(t))$  presented in Eq. 3.17

$$\dot{\mathbf{A}} = \mathbf{W}\Theta(\mathbf{A}) + \mathbf{b} , \quad (3.21)$$

where  $\Theta(\mathbf{A})$  is the matrix of features. In the SINDy algorithm, Brunton *et. al* (2016) suggest that  $\Theta(\mathbf{A})$  may consist of constant, polynomial, exponential and trigonometric functions. However, in many cases, it is difficult to know what set of features should be extracted from the data. Hence, we use machine learning to circumvent the problem of finding the functions which represent the dynamics of the problem. Therefore, the methodology can discover not only the weights  $\mathbf{W}$  and biases  $\mathbf{b}$  but also the features

$\Theta(\mathbf{A})$ . The “learned” features often result in a better performance when compared to those obtained using “engineered” features. A learning algorithm can find a proper set of features in minutes or hours, depending on the task complexity. On the other hand, manually engineered features would require a great amount of human time and effort for complex tasks (Goodfellow *et. al*, 2016).

Deep learning methods are feature learning algorithms that can find a proper set of features using multiple layers, from higher layer features defined in terms of lower layer features. Automatically learning features at multiple processing layers allows the learning of complex functions through mapping the input to the output directly from a given data (Bengio, 2009). In the present work, a deep feedforward neural network (DNN) is used to learn the weights  $\mathbf{W}$ , biases  $\mathbf{b}$  and features  $\Theta(\mathbf{A})$  of the dynamical systems investigated. Figure 3.3 shows a sample DNN architecture where the input  $\mathbf{X}$  of the DNN is the matrix  $\mathbf{A}$  and the target  $\mathbf{Y}$  is the matrix  $\dot{\mathbf{A}}$ . The DNN calculation procedure is presented in the following algorithm form 1. In the current work, the open source machine learning framework Tensorflow (Abadi *et. al*, 2016) is used for training the DNN.

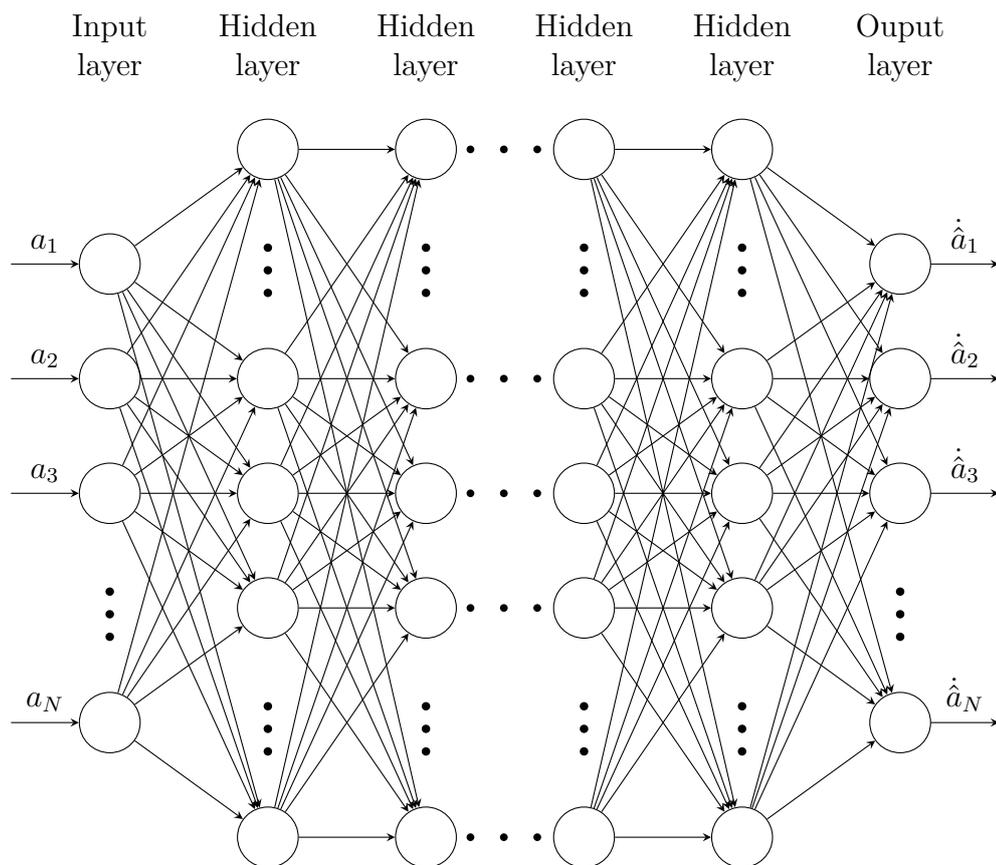


Figure 3.3: An example of a deep feedforward network with  $N$  inputs, several hidden layers (HL), and one output layer with  $N$  outputs.

---

**Algorithm 1:** Deep feedforward network (DNN).
 

---

**Input** : Training data  $\mathbf{X}$ , Target  $\mathbf{Y}$ , network depth  $L$ , activation function  $\sigma$ , number of hidden units  $n^{[l]}$  for layer  $l$ , learning rate  $\alpha$ , regularization parameter  $\lambda$ , maximum number of iterations  $n_{iter}$ , exponential decay rate for the 1st moment estimates  $\beta_1$ , exponential decay rate for the 2nd moment estimates  $\beta_2$ , and small constant for numerical stability  $\epsilon$

**Output:**  $\mathbf{W}^{[l]}$ ,  $\mathbf{b}^{[l]}$ ,  $\hat{\mathbf{Y}}$

- 1 **Initialization of parameters:** All the weights  $\mathbf{W}^{[l]}$  are initialized using Xavier's initialization (Xavier and Yoshua, 2010). The biases  $\mathbf{b}^{[l]}$  are initialized to zero. The matrix  $\mathbf{W}^{[l]}$  is of size  $n^{[l+1]} \times n^{[l]}$  and the vector  $\mathbf{b}^{[l]}$  is of size  $n^{[l]} \times 1$ ;
- 2 **Initialization of Adam parameters:** Adam parameters  $\mathbf{V}_{dW}^{[l]}$ ,  $\mathbf{V}_{db}^{[l]}$ ,  $\mathbf{S}_{dW}^{[l]}$  and  $\mathbf{S}_{db}^{[l]}$  are initialized to zero. They have the same dimensions as  $\mathbf{W}^{[l]}$  and  $\mathbf{b}^{[l]}$ ;
- 3 **for**  $iter = 1$  **to**  $n_{iter}$  **do**
- 4     **Forward propagation ;**
- 5      $\mathbf{G}^{[0]} = \mathbf{X}$  ;
- 6     **for**  $l = 1$  **to**  $L$  **do**
- 7          $\mathbf{Z}^{[l]} = \mathbf{W}^{[l]} \mathbf{G}^{[l-1]} + \mathbf{b}^{[l]}$ ;
- 8          $\mathbf{G}^{[l]} = \sigma(\mathbf{Z}^{[l]})$ ;  $\triangleright$  for the last layer  $L$ :  $\mathbf{G}^{[L]} = \mathbf{Z}^{[L]}$
- 9     **end**
- 10      $\hat{\mathbf{Y}} = \mathbf{Z}^{[L]}$ ;
- 11     **Cost function:**  $J = \frac{1}{2N_T} \left[ \sum_{j=1}^N \sum_{i=1}^{N_T} (\hat{Y}_{i,j} - Y_{i,j})^2 + \lambda \sum_{l=1}^L \sum_{k=1}^{n^{[l]}} \sum_{j=1}^{n^{[l+1]}} (W_{j,k}^{[l]})^2 \right]$ ;
- 12     **Backward propagation ;**
- 13      $d\mathbf{G}^{[L]} = d\mathbf{Z}^{[L]}$ ;  $\triangleright$  as  $\mathbf{G}^{[L]} = \mathbf{Z}^{[L]}$  then  $d\mathbf{Z}^{[L]} = \hat{\mathbf{Y}} - \mathbf{Y}$
- 14      $d\mathbf{W}^{[L]} = \frac{1}{N_T} [d\mathbf{Z}^{[L]} \cdot (\mathbf{G}^{[L-1]})^T]$ ;
- 15      $d\mathbf{b}^{[L]} = \frac{1}{N_T} \sum_{i=1}^{N_T} d\mathbf{Z}^{[L]}$ ;
- 16     **for**  $l = L - 1$  **to**  $1$  **do**
- 17          $d\mathbf{G}^{[l+1]} = (\mathbf{W}^{[l+1]})^T \cdot (d\mathbf{Z}^{[l+1]})$ ;
- 18          $d\mathbf{Z}^{[l]} = d\mathbf{G}^{[l]} * \sigma'(\mathbf{Z}^{[l]})$ ;
- 19          $d\mathbf{W}^{[l]} = \frac{1}{N_T} [d\mathbf{Z}^{[l]} \cdot (\mathbf{G}^{[l-1]})^T]$ ;
- 20          $d\mathbf{b}^{[l]} = \frac{1}{N_T} \sum_{i=1}^{N_T} d\mathbf{Z}^{[l]}$ ;
- 21     **end**
- 22     **Adam optimization (Kingma and Ba, 2014) ;**
- 23     **for**  $l = 1$  **to**  $L$  **do**
- 24          $\mathbf{V}_{dW}^{[l]} = \beta_1 \mathbf{S}_{dW}^{[l]} + (1 - \beta_1) d\mathbf{W}^{[l]}$ ;      $\mathbf{V}_{db}^{[l]} = \beta_1 \mathbf{S}_{db}^{[l]} + (1 - \beta_1) d\mathbf{b}^{[l]}$ ;
- 25          $\mathbf{S}_{dW}^{[l]} = \beta_2 \mathbf{S}_{dW}^{[l]} + (1 - \beta_2) (d\mathbf{W}^{[l]})^2$ ;      $\mathbf{S}_{db}^{[l]} = \beta_2 \mathbf{S}_{db}^{[l]} + (1 - \beta_2) (d\mathbf{b}^{[l]})^2$ ;
- 26          $\mathbf{V}_{dW}^{[l]} := \frac{\mathbf{V}_{dW}^{[l]}}{1 - \beta_1^{iter}}$ ;      $\mathbf{V}_{db}^{[l]} := \frac{\mathbf{V}_{db}^{[l]}}{1 - \beta_1^{iter}}$ ;      $\mathbf{S}_{dW}^{[l]} := \frac{\mathbf{S}_{dW}^{[l]}}{1 - \beta_2^{iter}}$ ;      $\mathbf{S}_{db}^{[l]} := \frac{\mathbf{S}_{db}^{[l]}}{1 - \beta_2^{iter}}$ ;
- 27          $\mathbf{W}^{[l]} := \mathbf{W}^{[l]} - \alpha \frac{\mathbf{V}_{dW}^{[l]}}{\sqrt{\mathbf{S}_{dW}^{[l]} + \epsilon}}$ ;
- 28          $\mathbf{b}^{[l]} := \mathbf{b}^{[l]} - \alpha \frac{\mathbf{V}_{db}^{[l]}}{\sqrt{\mathbf{S}_{db}^{[l]} + \epsilon}}$ ;
- 29     **end**
- 30 **end**

---

Once we have learned the parameters  $\mathbf{W}^{[l]}$  and  $\mathbf{b}^{[l]}$  of our model 3.21, we can use them to predict the temporal coefficients  $\mathbf{a}(t)$  given the initial conditions  $\mathbf{a}(t_1)$ . The system of coupled ODEs 3.17 can now be given by

$$\frac{d\hat{\mathbf{a}}(t)}{dt} = \mathbf{W}^{[L]} \mathbf{G}^{[L-1]} + \mathbf{b}^{[L]} = \mathbf{F}(\mathbf{a}(t)) , \quad (3.22)$$

where  $L$  is the index of the last layer,  $\mathbf{G}^{[L-1]}$  is the activation function matrix of the penultimate layer,  $\mathbf{W}^{[L]}$  is the weight matrix of the last layer, and  $\mathbf{b}^{[L]}$  is the bias vector of the last layer.

This system of coupled ODEs is integrated using an explicit 5 stage 4th-order Runge-Kutta scheme derived by Kennedy *et. al* (1999). As we can see in algorithm 1,  $\mathbf{G}^{[L-1]}$  depends on the weights  $\mathbf{W}$  and biases  $\mathbf{b}$  from previous layers. Thus for each stage of the 4th-order Runge Kutta, we need to perform forward propagation to obtain  $\mathbf{F}(\mathbf{a}(t))$ . As we have the temporal coefficients  $\mathbf{a}(t)$ , one can reconstruct the flow field using Eq. 3.8. However, we are interested in using a reduced-order model in circumstances other than simply reproducing the training data. The approach presented in this work allows us to predict the flow field beyond the training window because  $\bar{\mathbf{q}}(\mathbf{x})$  and  $\phi(\mathbf{x})$  depend only on the spatial coordinates  $\mathbf{x}$  and they are calculated using only the training data.

### 3.2.3 Hyperparameter Optimization

The performance of the DNN described in algorithm 1 depends dramatically on the selection of hyperparameters such as the network depth,  $L$ , number of hidden units for each layer,  $n^{[l]}$ , regularization parameter,  $\lambda$ , and learning rate  $\alpha$ . Indeed, finding an optimal set of hyperparameters which minimizes the loss function over a hyperparameter space is a challenging task given the substantial number of free parameters involved.

The manual search, grid search, random search (Bergstra and Bengio, 2012) and Bayesian optimization (Brochu *et. al*, 2010) are the most widely used procedures for the hyperparameter optimization. The manual search consists of a direct human trial and error procedure in the search for an optimal configuration of hyperparameters. This procedure is entirely based on prior experience of the user and there is a high probability that an optimal set of hyperparameters is not found. However, a manual search is still useful if the effect of a specific hyperparameter on the model performance can be monitored on the fly. In the grid search method, several combinations of hyperparameter values are tested in a range evenly spaced. This method is extremely time-consuming because the number of trials increases exponentially with the number of hyperparameters. In a random

search, one randomly selects each hyperparameter from a defined range and evaluates the model performance. It is time-consuming when a high-dimensional hyperparameter space is analyzed, however, Bergstra and Bengio (2012) empirically show that a random search outperforms a grid search for the hyperparameter optimization both in terms of computational time and model performance.

One of the recent strategies to find an optimal set of hyperparameters is the Bayesian optimization. It is a technique that involves constructing a probabilistic surrogate model to the data in order to determine the most promising hyperparameters to evaluate. Snoek *et. al* (2012) showed that Bayesian optimization was able to find optimal hyperparameters for a three-layer convolutional neural network considerably faster than previous approaches and outperformed the state of the art performance at selecting the set of hyperparameters on the CIFAR-10 data set (Krizhevsky, 2009).

In this work, we use two hyperparameter optimization strategies: random search and Bayesian optimization. For random search, the model generation procedure is presented in the following algorithm form 2. Likewise, Bayesian optimization has the same inputs as random search. To report the performance of each model from a set of candidates, we compute the mean absolute error (MAE) over the training data. The candidate models with lower MAE values are most likely those which will provide the “best” models. It is possible, however, that the model with the lowest MAE suffers from overfitting.

The current metric does not assess the generalization of the model. However, it is the only one available since we cannot split our data into training and validation sets. The validation data set should provide an unbiased evaluation of the ROM. In our case, we employ the temporal coefficients of the POD modes for the training stage of the model. These modes are computed using a correlation of different snapshots and if we construct the ROM using POD modes obtained with data including the validation set, our model would use a biased set for the training stage. This occurs because the POD correlation matrix would be computed for snapshots of both the training and validation sets. In order to overcome this issue, we reconstruct the flow field of the candidate models with lowest MAEs and then compute  $E_r = \|\mathbf{q}_{FOM} - \mathbf{q}_{ROM}\|$  over the validation set to select the best candidate model. This procedure could even be improved if  $E_r$  were computed for all models. However, the computational cost would considerably increase if such metric were employed.

An alternative to the current MAE procedure is to use Akaike’s information criterion (AIC) (Akaike, 1973) or the Bayesian information criterion (BIC) (Schwarz, 1978) as the model selection criteria. These methods try to balance the quality of the fit and model complexity and the main advantage is that there is no need for a validation set. The

Hyperparameter	Improves performance if	Reason	Warning
Number of hidden units, $n_{hidden}$	Increased	Increasing the number of hidden units augments the capacity of the model to represent more complex functions	Increasing this parameter may cause overfitting to the training data
Number of layers, $n_{layers}$	Increased	Same as above	Same as above. One should also be aware that a DNN with a large number of layers and a very small number of hidden units will not work properly.
Regularization parameter, $\lambda$	Reduced	Reducing the regularization parameter allows larger weights for the model features. One should expect that some of these features are the most relevant for the model.	Reducing the regularization parameter causes the model to be more prone to overfitting to the training data
Learning rate, $\alpha$	Tuned	If $\alpha$ is too small, the optimization process can be slow. If $\alpha$ is too high, the optimization method may lead to overshoot of local minima. This parameter is chosen by monitoring the learning curve.	If $\alpha$ is too large, the learning curve will show strong oscillations. If it is too small, the learning curve may stuck with a high value of the cost function.
Number of iterations, $n_{iter}$	Tuned	The number of iterations is strictly related to the learning rate. It is chosen by monitoring the learning curve.	As the number of iterations increases, the model goes from underfitting to optimal and, then, to overfitting the training data.

Table 3.1: Effects of hyperparameters on model performance.

---

**Algorithm 2:** Model generation procedure
 

---

**Input** : Number of candidate models  $n_{models}$ , minimum number of layers  $n_{layers_{min}}$ , maximum number of layers  $n_{layers_{max}}$ , number of inputs  $n_{inputs}$ , minimum number of hidden units  $n_{hidden_{min}}$ , maximum number of hidden units  $n_{hidden_{max}}$ , minimum order of magnitude of the regularization parameter  $\theta_{min}$ , and maximum order of magnitude of the regularization parameter  $\theta_{max}$ .

**Output:** Network depth  $L$ , number of hidden units for each layer  $n^{[l]}$  and regularization parameter  $\lambda$

```

1 for  $i = 1$  to  $n_{models}$  do
2    $\lambda_i = 10^{-rand(\theta_{min}, \theta_{max})}$ ;  $\triangleright$  Regularization parameter
3    $L_i = int(rand(n_{layers_{min}}, n_{layers_{max}}))$ ;  $\triangleright$  Network depth
4    $n_i^{[1]} = n_{inputs}$ ;  $n_i^{[L]} = n_{inputs}$ ;  $\triangleright$  Number of units for the input and output layers
5   for  $l = 2$  to  $L_i - 1$  do
6      $n_i^{[l]} = int(rand(n_{hidden_{min}}, n_{hidden_{max}}))$ ;  $\triangleright$  Number of hidden units for layer  $l$ 
7   end
8 end

```

---

downside is that AIC and BIC impose a penalty for model complexity which is related to the number of parameters. This can be a problem when dealing with deep neural networks due to their large number of parameters.

In our experiments, we noticed that a few of the hyperparameters employed in algorithm 1 need to be tuned for the best performance of the DNN. These are listed in algorithm 2. The remaining hyperparameters are determined according to the following procedures in order to reduce the hyperparameter search space. For instance, hyperparameters related to the Adam optimization, such as the exponential decay rates  $\beta_1$  and  $\beta_2$ , and the constant for numerical stability  $\epsilon$ , are set as described by Kingma and Ba (2014). The learning rate  $\alpha$  and the number of iterations  $n_{iter}$  are chosen by monitoring the loss function. For instance, we train a fraction of the models, selected randomly, with a initial learning rate and number of iterations, and then we plot the loss function. Next, we reduce the learning rate by a factor of 10, and train the models while still monitoring the loss function. We repeat this step until the loss function drops drastically and approaches  $10^{-6}$  with gentle oscillations as can be seen in figure 3.4. One should note that automated approaches such as learning rate decay and adaptive learning rate are more robust strategies to configure the learning rate (Goodfellow *et. al.*, 2016).

As can be observed in the results section, we use the same values of the learning rate and number of iterations for all flow simulations studied in this work and these values should serve as references for other flows. Following Goodfellow *et. al.* (2016), we provide table 3.1 with information of the hyperparameters presented in algorithm 2 based on

their effects on model performance. This table also presents a similar discussion for the hyperparameters chosen via manual search. We expect that it serves as a guideline for choosing the range of values to be explored for the hyperparameters presented in algorithm 2.

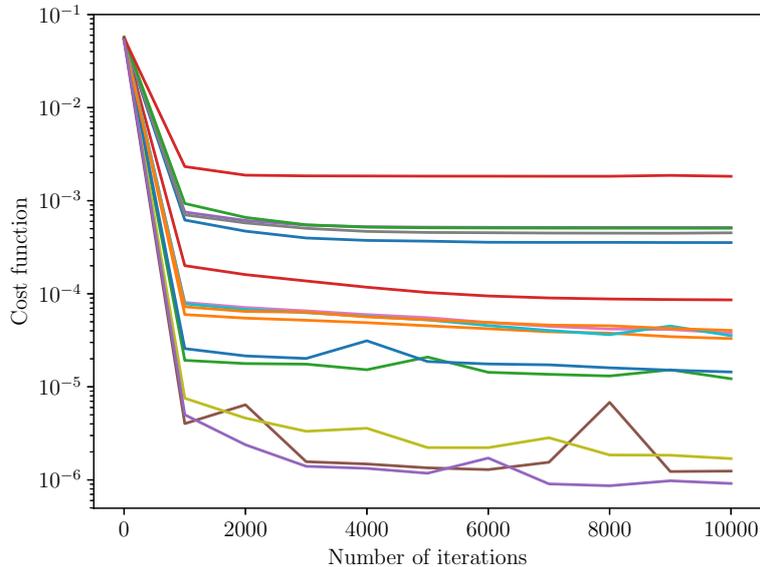


Figure 3.4: Cost function for different models.

Here, we consider that the activation function is also a hyperparameter since it plays a key role in the DNN performance. There are several available activation functions such as sigmoid, hyperbolic tangent (tanh), rectified linear unit (ReLU) (Nair and Hinton, 2010), exponential linear unit (ELU) (Clevert *et. al*, 2015), to name a few. For general regression problems, tanh and ELU are the most popular functions since they possess nonlinear properties and are continuously differentiable. Clevert *et. al* (2015) show that the ELU function reduces the vanishing gradient effect since its positive part returns the identity. Thus, in the positive part, the derivative is unitary and it is not contractive. On the other hand, tanh is contractive almost everywhere. Furthermore, in our experiments, the ELU function provided results with fewer iterations than a corresponding tanh network and, therefore, we use ELU as the activation function.

### 3.3 Data-driven ROMs

As previously discussed, the construction of reduced-order models is an active area of research in fluid mechanics and different methodologies for the development of ROMs are available. For example, POD-Galerkin based methods are directly related to the physics

of the problem through the projection of the Navier Stokes equations into a system of ordinary differential equations. Galerkin projection methods require the treatment of the linear and nonlinear spatial terms appearing in the full order model. Another issue with these methods relates to their expensive application for dynamical systems with strong nonlinearities where one should employ, for example, hyper-reduction techniques (Chaturantabut and Sorensen, 2010; Carlberg *et. al*, 2011; Zimmermann and Willcox, 2016).

In the current approach, we employ DNNs for the regression step in a context similar to the SINDy algorithm. Both the DNN and the original SINDy approaches are data-driven methods and, instead of having a direct connection with the physics of the problem, these techniques learn from data. An advantage of these methods is that the nonlinearities of the problem are considered in the temporal derivatives of the primitive variables, which are obtained from the full order model. Therefore, neither the SINDy nor the DNN method require the treatment of spatial derivatives.

In general, the resulting reduced-order models constructed via DNNs are not (directly) physically interpretable due to the nonlinearity of the matrix of features and their weights and biases, which are not sparse. We hope that interpretability of complex models can be improved by new topics of research such as “accountable machine learning” (Navarro *et. al*, 2018). The physics of systems with complicated models can also be interpreted by the structure of the state space, e.g., fixed points, Lyapunov exponents, fractal dimensions, bifurcations, and DNN models can also be useful for computing such measures. Learning from models simply uses the fact that many realizations of the physical system can be computed very quickly to explore parts of the state space which are not contained in the training data. In some cases, such as the flow past a cylinder, models constructed using sparse regression could directly lead to physically interpretable results (Brunton *et. al*, 2016). As expected, the application of DNNs for the regression step adds a penalty cost compared to sparse regression. However, it is shown in this work that, for the cases analyzed, models obtained using neural networks present better long-term predictive capabilities compared to sparse regression. Both these issues will be discussed in the results chapter.

## 4 RESULTS

In this chapter, the proposed ROM methodology is first tested in the reconstruction of the dynamics of a damped cubic oscillator. Then, we evaluate the capability of DNN-ROMs to reproduce the dynamics of a compressible flow past a cylinder including its noise generation. A comparison between the current DNN technique against sparse regression is also presented for the transient regime of an incompressible flow past a cylinder. Finally, the method is employed to create a ROM of turbulent flow involving dynamic stall of a plunging airfoil. Again, the DNN solution is compared against that obtained by sparse regression. We also evaluate the reconstruction of SPOD temporal modes using different finite difference schemes and times steps. In each example, we show the ability and limitations of the methods to identify the dynamics of the different nonlinear systems comparing the ROM and full order model (FOM) solutions.

### 4.1 Nonlinear oscillator

For this first example, we consider a canonical problem in system identification (Brunton *et. al*, 2016; Rudy *et. al*, 2018): the two-dimensional nonlinear oscillator. The system dynamics are given by

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -0.1 & 2 \\ -2 & -0.1 \end{bmatrix} \begin{bmatrix} x_1^3 \\ x_2^3 \end{bmatrix}. \quad (4.1)$$

with initial conditions  $\begin{bmatrix} x_1 & x_2 \end{bmatrix}^T = \begin{bmatrix} 2 & 0 \end{bmatrix}^T$ .

We generate 4000 snapshots from  $t = 0$  to  $t = 40$  by integrating equation 4.1 using an explicit 5 stage 4th-order Runge-Kutta scheme (Kennedy *et. al*, 1999) with a time step of  $h = 0.01$ . The training window spans the period  $0 \leq t \leq 10$  and the remaining data is used as the test set. The system reconstruction is obtained following the procedures defined in section 3.2. For this first example, it is not necessary to use POD due to the low dimensionality of the system. The parameters employed in algorithm 2 are presented in table 4.1 and the best set of hyperparameters obtained via random search is listed in table 4.2.

Figure 4.1 presents the solutions obtained by the FOM (true model) and ROM for the

Table 4.1: Model generation parameters for ROM of nonlinear oscillator

$n_{models}$	$n_{layers_{min}}$	$n_{layers_{max}}$	$n_{hidden_{min}}$	$n_{hidden_{max}}$	$\theta_{min}$	$\theta_{max}$
300	3	10	8	36	2	6

Table 4.2: Best set of hyperparameters for ROM of nonlinear oscillator

DNN architecture	$\sigma$	$\alpha$	$\lambda$	$n_{iter}$	$\beta_1$	$\beta_2$	$\epsilon$
2 - 10 - 2	ELU	0.001	$1.4062 \times 10^{-6}$	20000	0.9	0.999	$1.0 \times 10^{-8}$

damped cubic oscillator. Results show that the proposed algorithm accurately reproduces the system dynamics during the training window and beyond, for the test set.

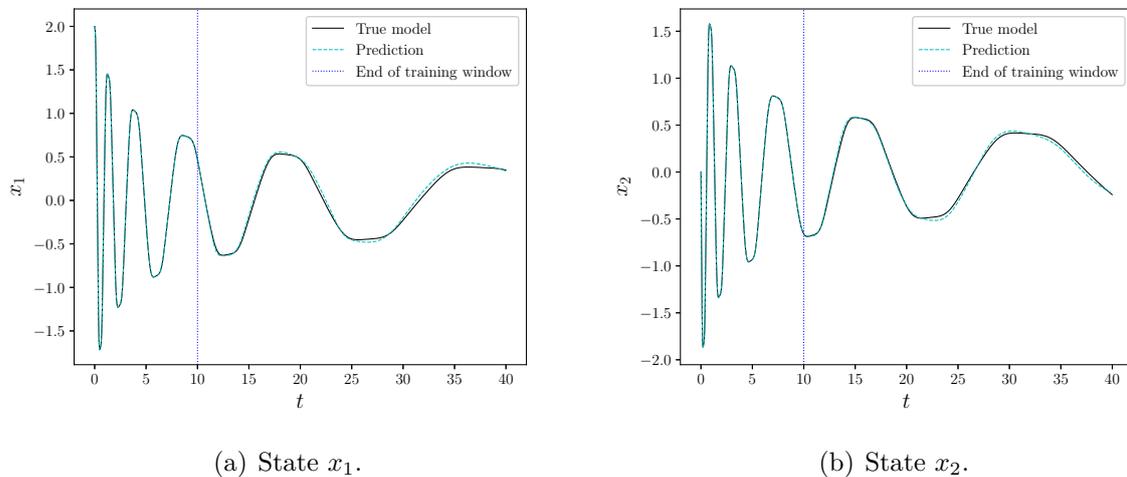


Figure 4.1: Comparison between true model (solid black line) and ROM prediction (dashed cyan line) for nonlinear oscillator.

## 4.2 Flow past a cylinder

In this case, the full order model (FOM) is obtained by solving the compressible Navier Stokes equations as detailed in section 2.2. The numerical simulations are conducted for Reynolds and Mach numbers  $Re = 150$  and  $M = 0.4$ , respectively. These dimensionless parameters are computed based on freestream quantities. The grid configuration consists of a body-fitted O-grid with  $421 \times 751$  points in the streamwise and wall-normal directions, respectively.

The flow is recorded for 1120 snapshots with dimensionless time steps of  $h = 0.05$ . The snapshots are collected after an initial transient period of the simulation is discarded. The reduced-order model (ROM) is obtained following the procedure described in section

Table 4.3: Model generation parameters for ROM of compressible flow past a cylinder

$n_{models}$	$n_{layers_{min}}$	$n_{layers_{max}}$	$n_{hidden_{min}}$	$n_{hidden_{max}}$	$\theta_{min}$	$\theta_{max}$
500	6	10	10	64	2	6

Table 4.4: Best set of hyperparameters for ROM of compressible flow past a cylinder

DNN architecture	$\sigma$	$\alpha$	$\lambda$	$n_{iter}$	$\beta_1$	$\beta_2$	$\epsilon$
10 - 31 - 51 - 25 - 32 - 42 - 51 - 27 - 26 - 10	ELU	0.001	$4.4085 \times 10^{-5}$	10000	0.9	0.999	$1.0 \times 10^{-8}$

3.2 using the pressure norm for the POD correlation matrix. It is important to mention that the use of a kinetic energy norm produced similar results for this case. The training data comprises the first 280 snapshots of the FOM and the remaining data is employed as the test set. The set of hyperparameters employed in algorithm 2 is listed in table 4.3 and the best set of these parameters obtained via random search is presented in table 4.4. For this case, Bayesian optimization was also able to produce accurate models but at a higher computational cost compared to random search. One can see that the best architecture of the neural network for this case has 10 layers. However, we also found several other stable and accurate models with six layers, for example. The fact that the best model was found with 10 layers is a coincidence.

Figures 4.2 to 4.5 show contours of density,  $x$ -momentum,  $z$ -vorticity, and divergence of velocity, respectively, along the cylinder and wake regions at time  $t = 410$ , which is beyond the training window. The snapshots allow a comparison of the results between the FOM and ROM using 2 and 10 POD modes out of 280 modes. Hence, the flow is reconstructed using 0.7% and 3.5% of the total information available from the full order model. Although the current simulation is performed for a compressible flow at  $M = 0.4$  and  $Re = 150$ , we could verify that the POD spatial eigenfunctions are almost identical to those from Noack *et. al* (2003), which were obtained for an incompressible flow and  $Re = 100$ . Reconstruction of the individual flow variables with 2 POD modes could recover between 50 and 80% of the total modal energy, depending on the variable. For example, density is reconstructed using 50% of the total energy of the system dynamics while  $y$ -momentum is reconstructed with 80% of the total modal energy. Here, we use the term “modal energy” to refer to the ratio between the sum of  $N$  POD eigenvalues used in a particular reconstruction over the entire range  $N_T$  of eigenvalues available,  $\sum_{i=1}^N \lambda_i / \sum_{i=1}^{N_T} \lambda_i$ . For 10 POD modes, the reconstructions could recover 99% of the energy for all variables and, therefore, should lead to an accurate flow representation. For this particular case, the spectral proper orthogonal decomposition technique is not required since the modes are almost monochromatic and, hence, do not require filtering.

One can observe from the figures that the computations of the flow using the ROM

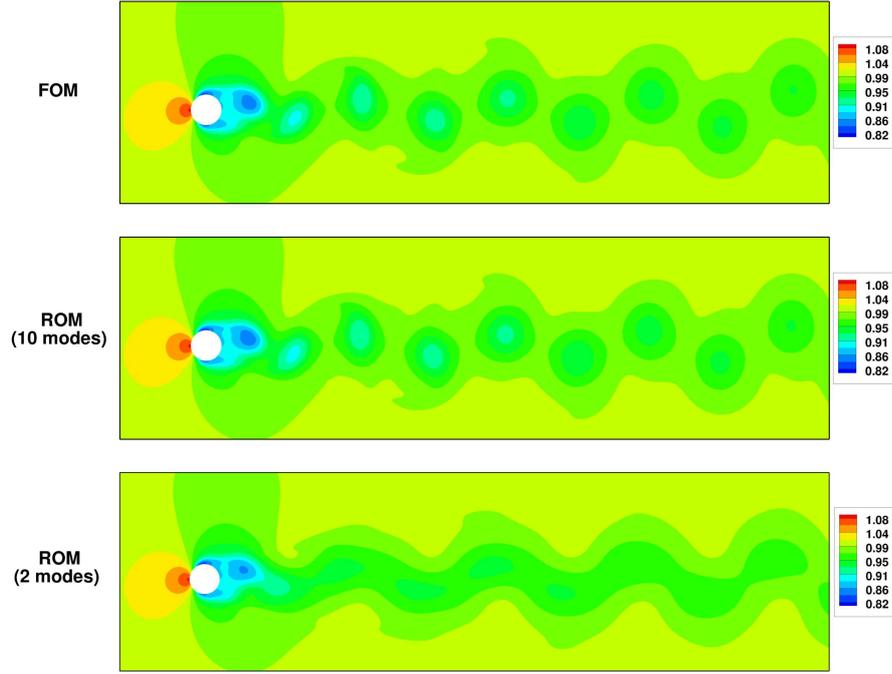


Figure 4.2: Contours of density,  $t = 410$ .

framework show good agreement compared to those obtained by the FOM. For the current Reynolds number, the flow develops a typical von Kármán vortex street along the cylinder wake. The periodical pattern of the vortex shedding can be observed in the contours of  $z$ -vorticity. Noise generation also occurs in the current unsteady compressible flow simulation. In this case, pressure fluctuations along the cylinder surface are scattered to the far-field and can be observed in the contours of dilatation shown in figure 4.5. Both the near-field hydrodynamics and the far-field acoustics are recovered by the ROM. The reconstruction using 10 POD modes show an excellent agreement with the FOM. When 2 POD modes are employed in the flow reconstruction, discrepancies between the ROM and FOM are evident from the figures. However, the main features of the flow are still recovered by the model. One should note that, despite the use of only two modes, the dynamical system is still stable beyond the training region.

In order to show a more qualitative evaluation of the model reconstructions, the density and  $x$ -momentum fluctuation time histories are presented for the FOM and ROMs in figures 4.6 and 4.7, respectively. The figures on the left column show results for a probe located just behind the cylinder, close to the surface, at  $(x, y) = (0.55, -0.06)$ . The cylinder has radius 0.5 and its center is positioned in the origin of the Cartesian system. On the right column, results are obtained for a probe downstream the cylinder wake, at  $(x, y) = (1.1, -0.06)$ . Results are shown for both the training window period and beyond. When 2 POD modes are employed, the solutions show a less accurate representation of

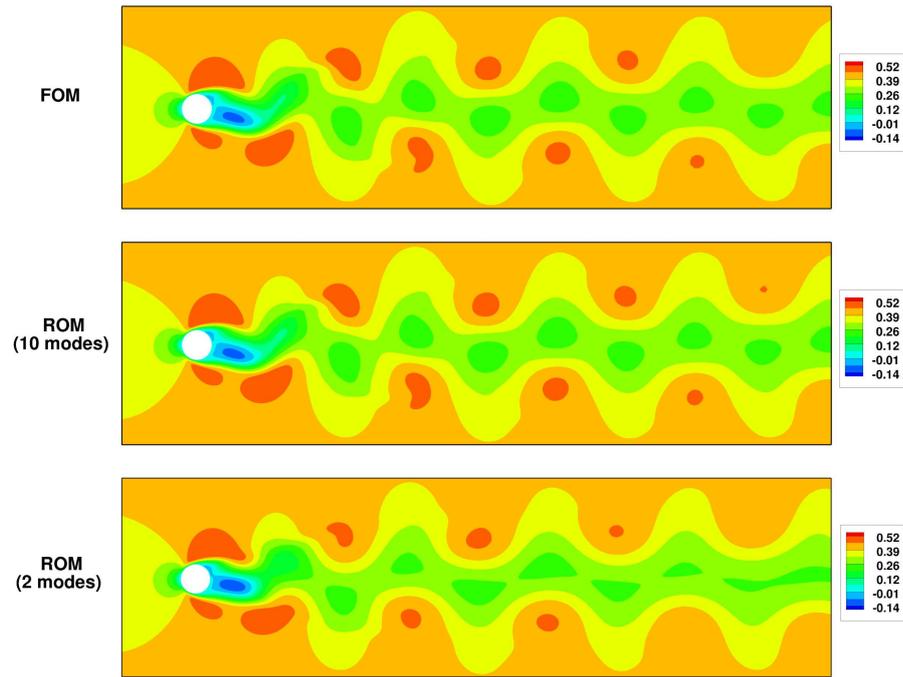


Figure 4.3: Contours of  $x$ -momentum,  $t = 410$ .

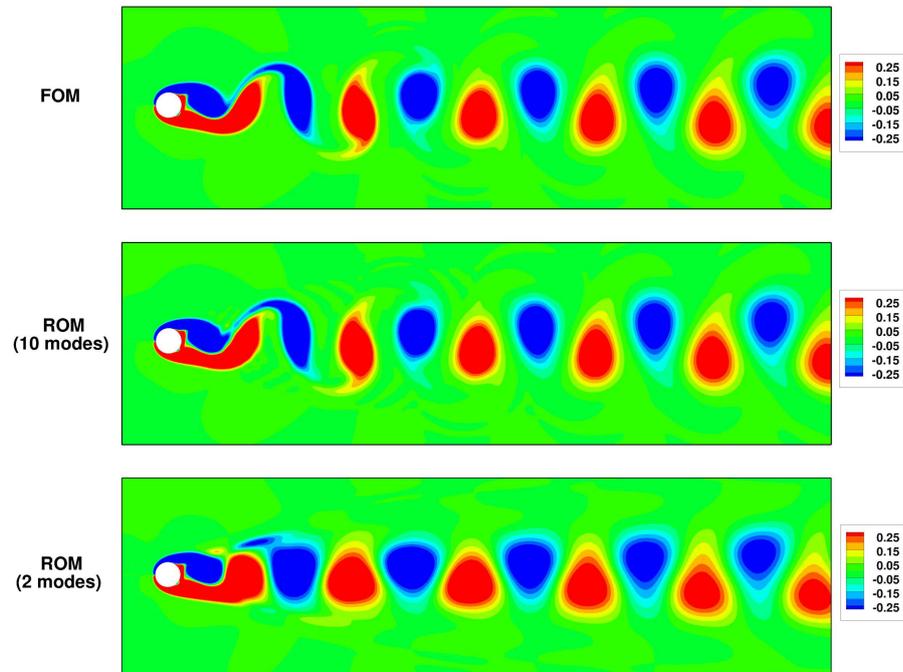


Figure 4.4: Contours of  $z$ -vorticity,  $t = 410$ .

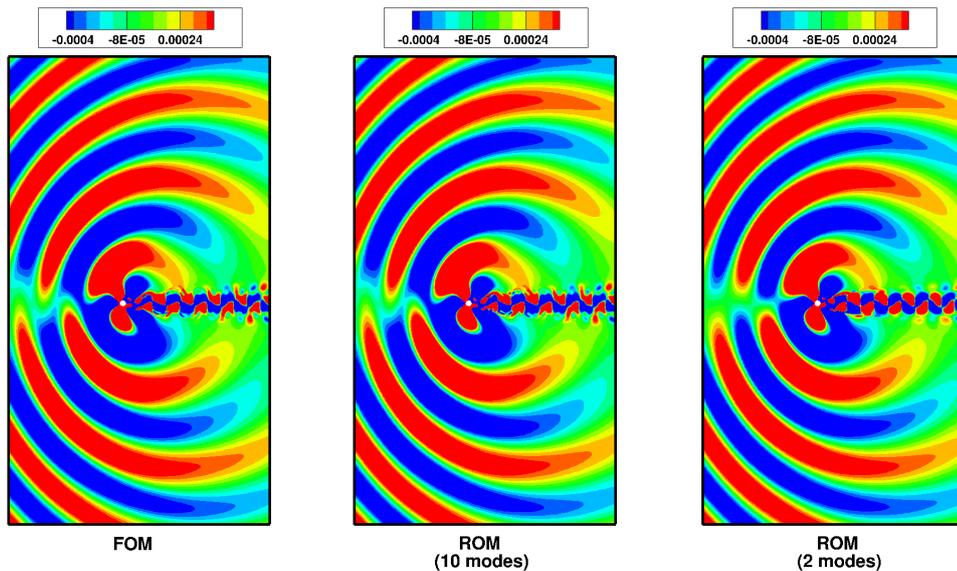
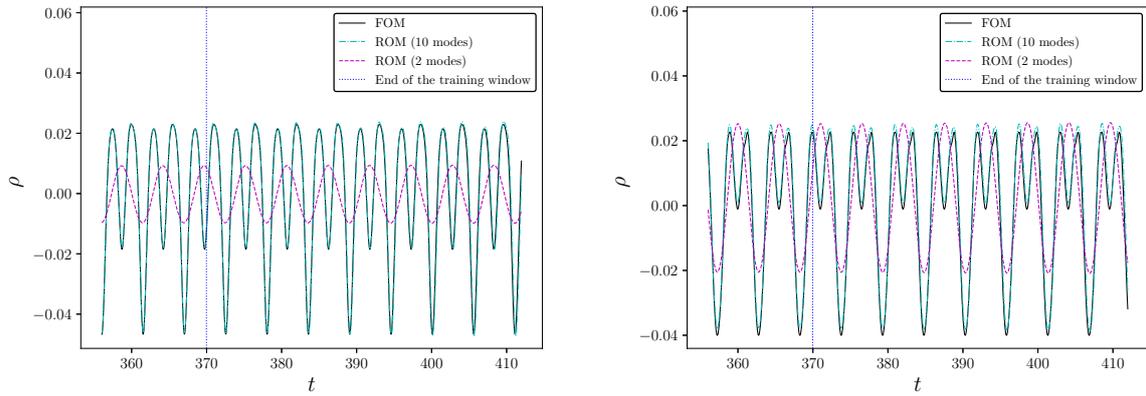


Figure 4.5: Contours of divergence of velocity,  $t = 410$ .

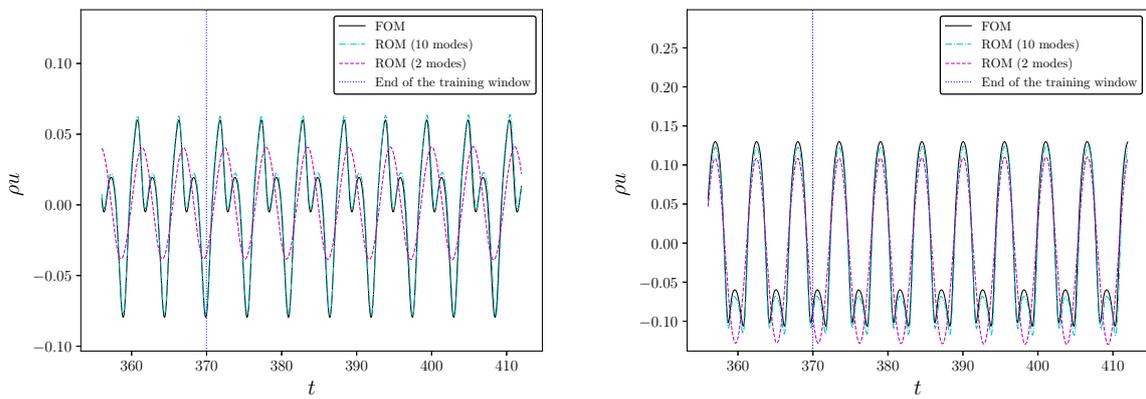
the dynamics observed in the FOM. The density reconstruction is that with the highest discrepancy and that is attributed to the lower energetic content achieved by the first 2 POD modes. One can notice that the reduced-order model accurately reproduces the full order model results during and beyond the training window when 10 POD modes are employed in the reconstruction.

In order to test the robustness of the method, we employ the DNN approach for the reconstruction of the transient regime of an incompressible cylinder flow. For this study, the 2 most energetic POD modes containing both the transient and limit cycle dynamics of the flow are obtained from Brunton *et. al* (2016). These modes contain both the transient and limit cycle dynamics of the flow. Figure 4.9 shows the POD temporal modes used for training and testing the DNNs. The first 5000 temporal instants are used to train the models and one can see this data set represented by the black line to the left of the vertical line marking the end of the training window. To the right of the training window, the black line represents the test data which is the correct solution for the temporal dynamics. Figure 4.9 also shows the results obtained by the current DNN approach and by sparse regression. In the case of sparse regression, we employ the same model obtained by Brunton *et. al* (2016) in table 11 of the Supporting Information report. As one can observe, the DNN is able to recover accurately both the transient and limit cycle solutions of the test data for both POD modes. On the other hand, sparse regression reconstructs the transient portion of the dynamics but not the long term prediction of the limit cycle. Several models obtained by the DNNs presented similar results compared to those shown in figure 4.9 and some had similar neural network architectures compared to that of table



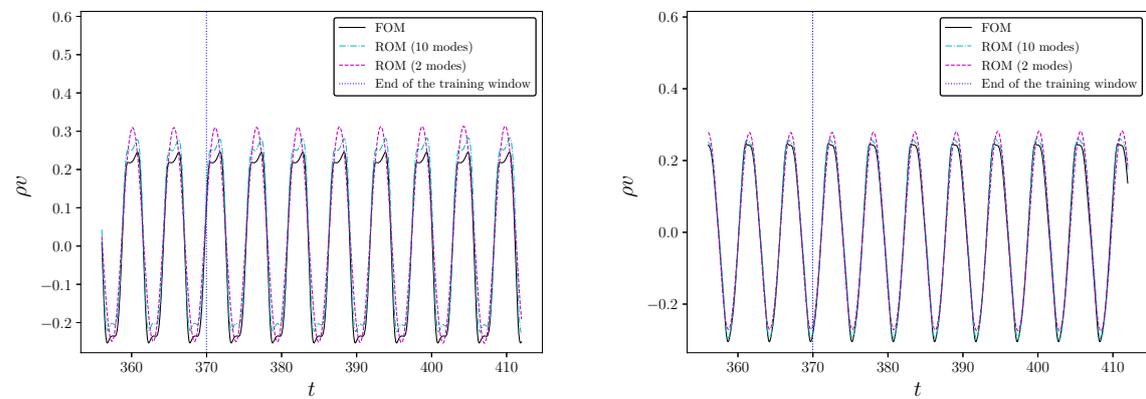
(a) Probe positioned just behind the cylinder. (b) Probe away from the cylinder, along the wake.

Figure 4.6: Fluctuation time history of density.



(a) Probe positioned just behind the cylinder. (b) Probe away from the cylinder, along the wake.

Figure 4.7: Fluctuation time history of  $x$ -momentum.



(a) Probe positioned just behind the cylinder. (b) Probe away from the cylinder, along the wake.

Figure 4.8: Fluctuation time history of  $y$ -momentum.

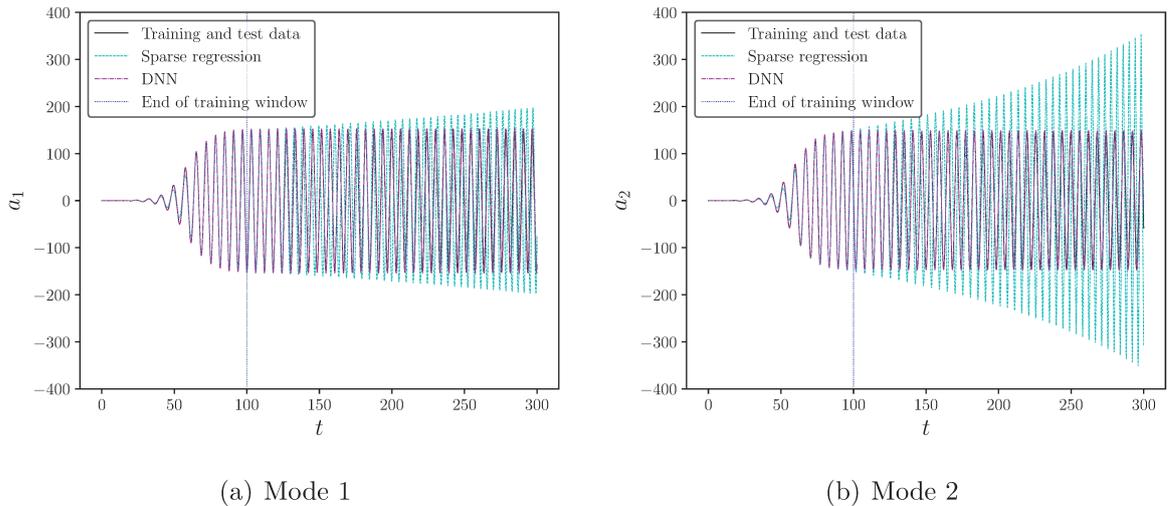


Figure 4.9: Reconstruction of POD temporal modes for transient solution of incompressible flow past a cylinder.

4.4. These results show that the proposed DNN approach has good long-term predictive capabilities and can learn the transient features of the flow.

### 4.3 Deep dynamic stall of plunging airfoil

The present study concerns a plunging SD7003 airfoil in deep dynamic stall. The flow conditions have a reduced frequency  $k = \pi f L / U_\infty = 0.5$ , where  $f$  is the plunging frequency,  $L$  is the chord of the airfoil and  $U_\infty$  is the reference free stream velocity. The motion amplitude is set as  $h_o / L = 0.5$  with a static angle of attack  $\alpha_0 = 8$  deg. The chord Reynolds number based on the freestream velocity is  $Re = 60,000$  and the freestream Mach number is  $M = 0.1$ . This flow condition is relevant for micro air vehicle applications and this case is selected based on the availability of published results from other high fidelity simulations (Visbal, 2011) and particle image velocimetry data (Baik *et. al*, 2019; Ol *et. al*, 2009). In a previous work, Ramos *et. al* (2019) performed a mesh refinement study and validation of the numerical solutions against the references above.

The present mesh configuration consists of a body-fitted O-grid with  $441 \times 300 \times 64$  points in the streamwise, wall-normal and spanwise directions, respectively. The grid is generated with 70% of the surface points located in the suction side of the airfoil to improve the capturing of finer flow scales developing along the turbulent region of the flow. Due to the favorable pressure gradients in the pressure side of the airfoil, the flow remains laminar along the entire cycle of the plunging motion. The trailing edge of the SD7003

Table 4.5: Model generation parameters for ROM of plunging airfoil in deep dynamic stall

$n_{models}$	$n_{layers_{min}}$	$n_{layers_{max}}$	$n_{hidden_{min}}$	$n_{hidden_{max}}$	$\theta_{min}$	$\theta_{max}$
700	6	12	10	64	2	6

airfoil is rounded in the current simulations with an arc of radius  $r/L = 0.0008$ . This procedure is required for retaining the smoothness of the metric terms computed by the high-order compact scheme. The spanwise domain is set as  $z/c = 0.4$  similarly to Visbal (2011) and the dimensionless time step of the simulation is set as  $\Delta t^* = \frac{\Delta t U_\infty}{L} = 0.00008$ .

The plunge motion occurs with an effective angle of attack in the range of  $-6 \text{ deg.} \leq \alpha \leq 22 \text{ deg.}$  Defining  $\psi$  as the angular position in the plunging cycle, we say that at  $\psi = 0 \text{ deg.}$  the airfoil has no velocity in the  $y$ -direction and is at the top-most position of the plunging motion. At  $\psi = 90 \text{ deg.}$  it has the highest velocity in the  $y$ -direction downwards and, at  $\psi = 180 \text{ deg.}$ , it has no velocity and is at the bottom-most position of the plunging motion. Finally, at  $\psi = 270 \text{ deg.}$  it has the highest velocity in the  $y$ -direction upwards. In summary, during the down-stroke, instabilities begin to form in the suction side of the airfoil, growing and eventually breaking into finer structures. While this takes place, the dynamic stall vortex forms along the leading edge and is transported through the airfoil suction side increasing the overall lift and creating a nose-down pitching moment. As the leading-edge vortex (LEV) approximates the trailing edge, a trailing-edge vortex (TEV) forms pushing the LEV away from the airfoil. A more complete discussion of the flow dynamics can be found in Visbal (2011) and Ramos *et. al* (2019).

Figure 4.10 shows iso-surfaces of Q-criterion colored by pressure and it is possible to observe the main flow features described above. In figures 4.10(c) and (d), it is possible to compare the 3-D solutions of the FOM and ROM, respectively, for the leading edge vortex formation. Figures 4.10(e) and (f) show a similar comparison for the instant where the trailing edge vortex forms. For both instants, one can observe that the ROM is able to reconstruct the larger scale features of the 3-D flow. The ROM is trained using the first 2 cycles of the plunging motion and the solutions presented in figure 4.10 are computed for the fourth cycle, showing that the model is able to reproduce the 3-D flow dynamics beyond the training window. For this study, the parameters employed in algorithm 2 are presented in table 4.5 and the best set of hyperparameters, obtained via random search, is listed in table 4.6. It is important to mention that, for this case, Bayesian optimization was unable to produce stable and accurate models.

The first 16 SPOD modes are employed to reduce the dimensionality of the input data in the 3-D flow reconstruction. Other POD reconstructions were tested with a different

Table 4.6: Best set of hyperparameters for ROM of plunging airfoil in deep dynamic stall (3D flow)

DNN architecture	$\sigma$	$\alpha$	$\lambda$	$n_{iter}$	$\beta_1$	$\beta_2$	$\epsilon$
16 – 19 – 52 – 50 – 31 – 16	ELU	0.001	$8.0938 \times 10^{-5}$	10000	0.9	0.999	$1.0 \times 10^{-8}$

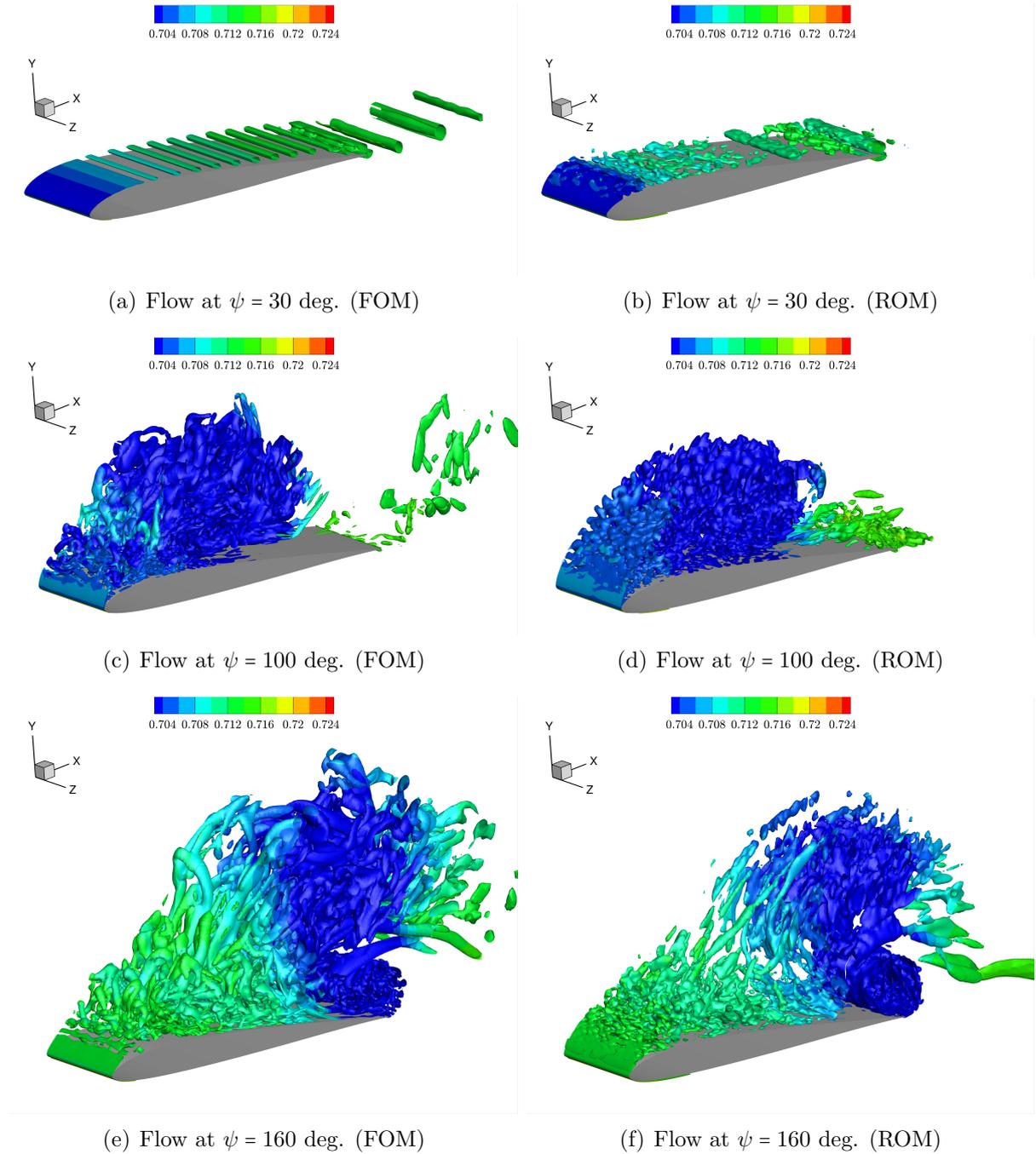
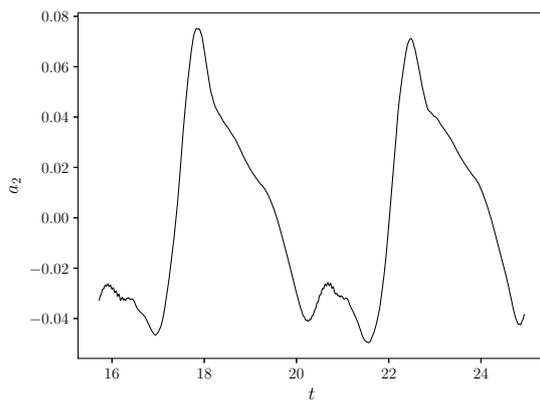
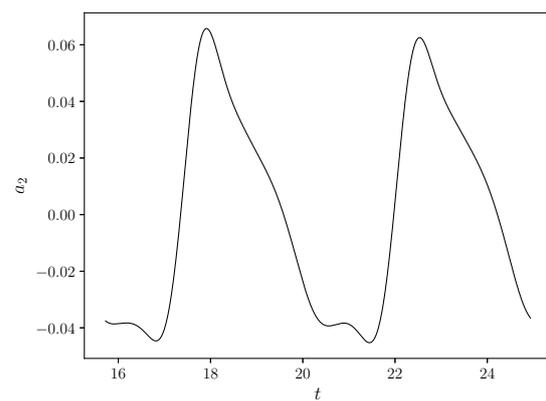


Figure 4.10: Iso-surfaces of  $Q$  criterion colored by pressure at different instants of the plunge motion for the fourth cycle. The ROM was trained using only the first 2 cycles.

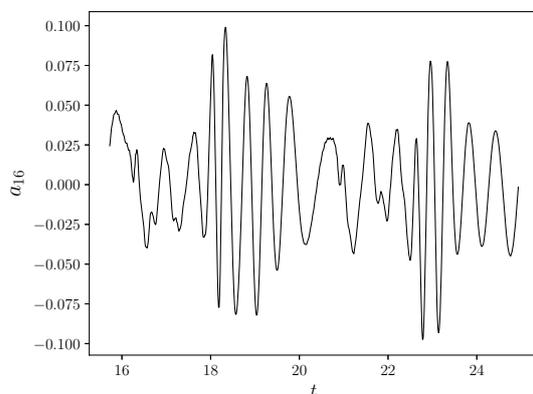
number of modes and it was observed that the first 16 modes contained the main features of the leading and trailing edge vortex formation. For example, adding 10 more modes did not improve significantly the solution and, beyond mode 26, the SPOD temporal modes presented complex behavior, being composed of several frequencies and difficulting the training stage of the DNNs. This issue could be improved running the simulation for a longer period with a lower time step to improve convergence of the POD modes. One should note that the spectral proper orthogonal decomposition allows an energy shift of the modes to obtain frequency filtered temporal dynamics. In this sense, the high-frequency noise observed in the POD temporal modes is not discarded but is shifted to higher SPOD modes. This procedure allows a better pairing of POD modes if the coherent structures have periodicity. In the present study, due to the unsteady boundary conditions and the lack of symmetry in the flow, we do not expect pairing of the SPOD modes.



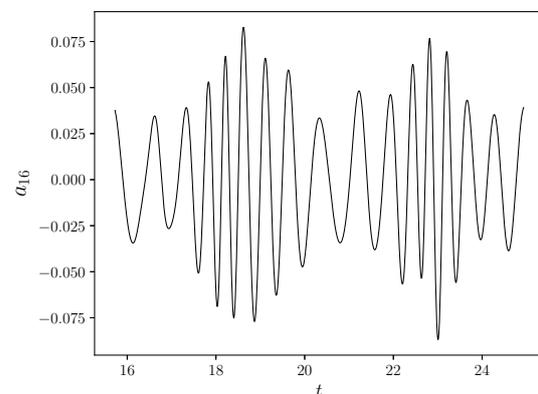
(a) POD temporal mode 2.



(b) SPOD temporal mode 2.



(c) POD temporal mode 16.



(d) SPOD temporal mode 16.

Figure 4.11: Characteristics of POD and SPOD temporal modes for 3D flow past airfoil under plunging motion.

Table 4.7: Best set of hyperparameters for ROM of plunging airfoil in deep dynamic stall (spanwise-averaged flow)

DNN architecture	$\sigma$	$\alpha$	$\lambda$	$n_{iter}$	$\beta_1$	$\beta_2$	$\epsilon$
10 – 46 – 23 – 17 – 19 – 10	ELU	0.001	$2.3365 \times 10^{-5}$	10000	0.9	0.999	$1.0 \times 10^{-8}$

For the present turbulent flow, the temporal modes of the standard snapshot POD are composed by several frequencies and, moreover, contain some noise that degrades the training of the DNNs. The SPOD provides the most energetic modes for specific frequency bands, allowing a better identification of the individual modes and smoothing out the temporal coefficients. Here, a box filter is applied to 25% of the POD correlation matrix. We observed that filter values in the range of 20% to 40% provide good results for the model reconstructions. If a lower filter window is employed, noise can still be present in the temporal modes. On the other hand, higher filter windows may considerably modify the temporal modes transforming them into quasi-sinusoidal signals. This is undesirable since the relevant dynamics of the flow can be lost and too many SPOD modes may be required for the construction of the ROM. An example of the impact of SPOD on the characteristics of temporal modes 2 and 16 can be seen in figure 4.11. In this example, the second POD mode exhibits high-frequency noise which is filtered by the SPOD. Meanwhile, POD mode 16 shows a complicated pattern due to the contribution of several frequencies. This temporal coefficient is considerably simplified by the application of SPOD as can be observed in the figure. It is clear that, different from the cylinder case, for the plunging airfoil, the POD modes contain more complex dynamics which are composed of multiple Fourier modes.

In order to further evaluate the present DNN approach, the ROM reconstructions are also performed for spanwise-averaged solutions of the flow and the ROM is constructed following the procedures defined in section 3.2. We employ 6 plunging cycles (2244 snapshots) for computing the models, discarding initial flow transients. The training data contains the first two cycles (748 snapshots) of the FOM and the remaining data is used as the test set. The parameters employed in algorithm 2 are the same as those presented in table 4.5 and the best set of hyperparameters, obtained via random search, is listed in table 4.7. Here, we apply the box filter to 25% of the POD correlation matrix and employ the first 10 SPOD modes for the ROM construction, which was sufficient to recover the most important features of the dynamic stall solution for the spanwise-averaged study.

Figures 4.12 and 4.13 present snapshots of density and x-component of momentum, respectively, for the FOM and ROM. The flow features of the dynamic stall can be observed for different stages of the plunging cycle, given by different values of  $\psi$  in the

figures. It is possible to track the LEV over the suction side of the airfoil. The current ROM is able to recover the important dynamics of the leading edge vortex formation, its transport and ejection, besides the trailing edge vortex formation and ejection. One should note that the results presented in these figures are obtained for a plunging cycle beyond the training region. Therefore, the current ROM is capable of reproducing the flow dynamics for the test set. The high wavenumber features shown in the FOM are not present in the SPOD modes and, hence, they are not recovered by the ROM. If additional SPOD modes were employed in the model reconstruction, these features would appear. However, the overall cost of the simulations would increase considerably if accurate higher POD modes were required. For the present study, we observed that the fine scale flow dynamics are related to higher modes since most of the energetic content is related to the airfoil plunging motion represented by the first POD modes.

Fluctuation time histories are presented in figure 4.14 for density and  $x$ -momentum. These properties are computed by the FOM and ROM at probe locations in the proximity of the leading and trailing edges, at  $(x, y) = (0.07, 0.07)$  and  $(x, y) = (0.97, 0.07)$ , respectively. It is important to mention that the airfoil leading edge is at the origin of the Cartesian system. The plots show the training and test regions demonstrating that the ROM is stable beyond the training set and can accurately reproduce the main dynamics of the flow. The fast oscillations resolved by the FOM are not represented due to the SPOD basis truncation. However, all the relevant features of the flow fluctuations are captured by the DNN model obtained with the first 10 SPOD modes.

In figure 4.15, one can observe a comparison between ROMs built using the current DNN approach and sparse regression. For this case, both the LASSO and Ridge regression techniques were tested with the original SINDy algorithm. Although the LASSO presented a higher computational cost, it provided better models compared to the Ridge regression. For this case, the computational cost of the DNN regression is, on average, 40 times higher than that of SINDy. However, as shown for SPOD modes 1 and 8, the SINDy algorithm was not able to provide stable models. The same can be said for the other SPOD modes. Sparse regression can learn the dynamics during the training window and also for a few cycles in the test set but its solution is not stable for long-time predictions. On the other hand, despite the higher cost, the DNN approach presents good long-time predictive capabilities with stable and accurate solutions beyond the training window.

In order to evaluate the robustness of the present framework, we test the capability of the ROM in reconstructing the flow field using a time step different than that of the simulation. Figure 4.16 shows the reconstruction of SPOD temporal modes using different finite difference schemes and time steps. The derivatives of the temporal modes

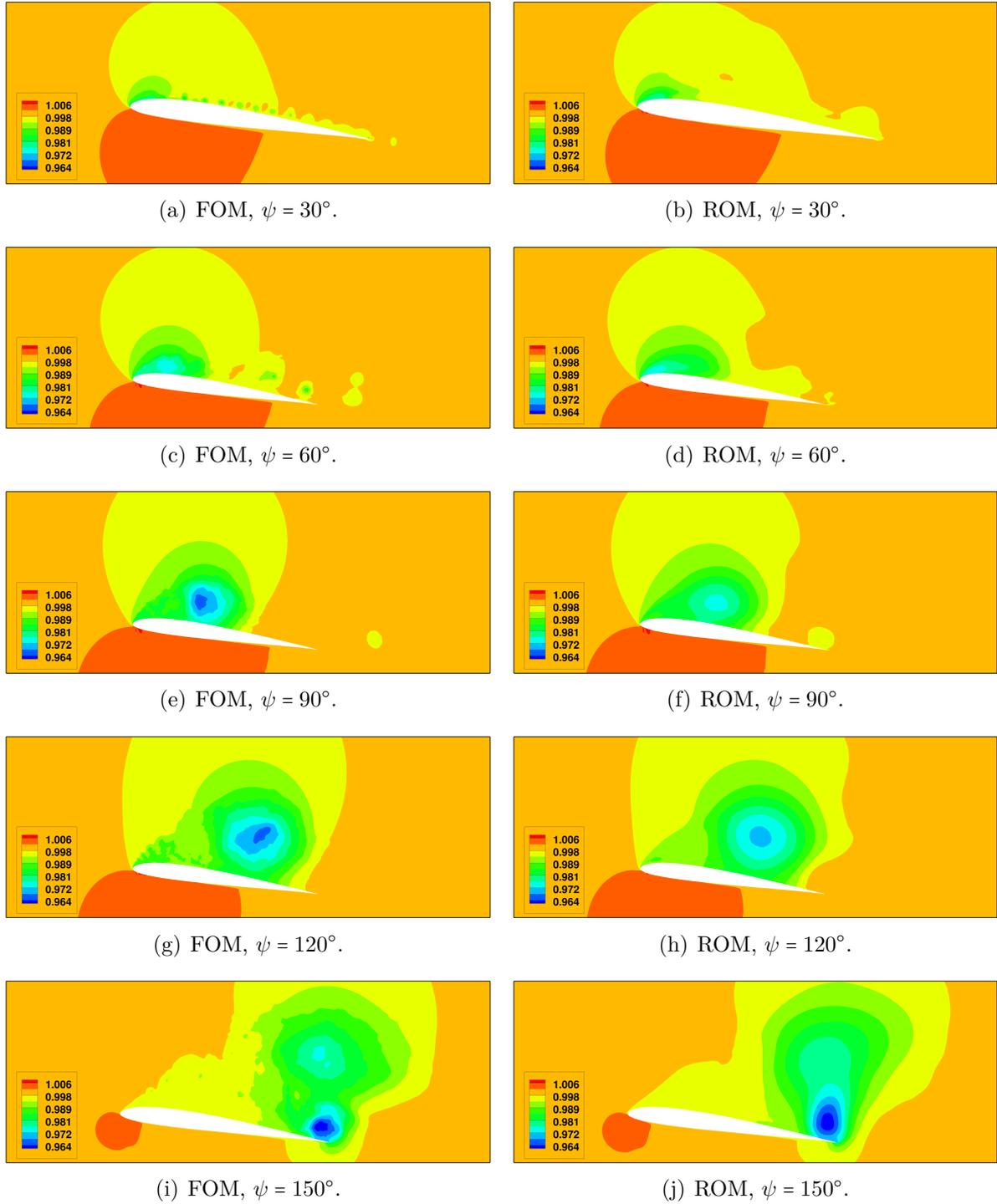


Figure 4.12: Density contours at several phase angles for the fifth plunging cycle.

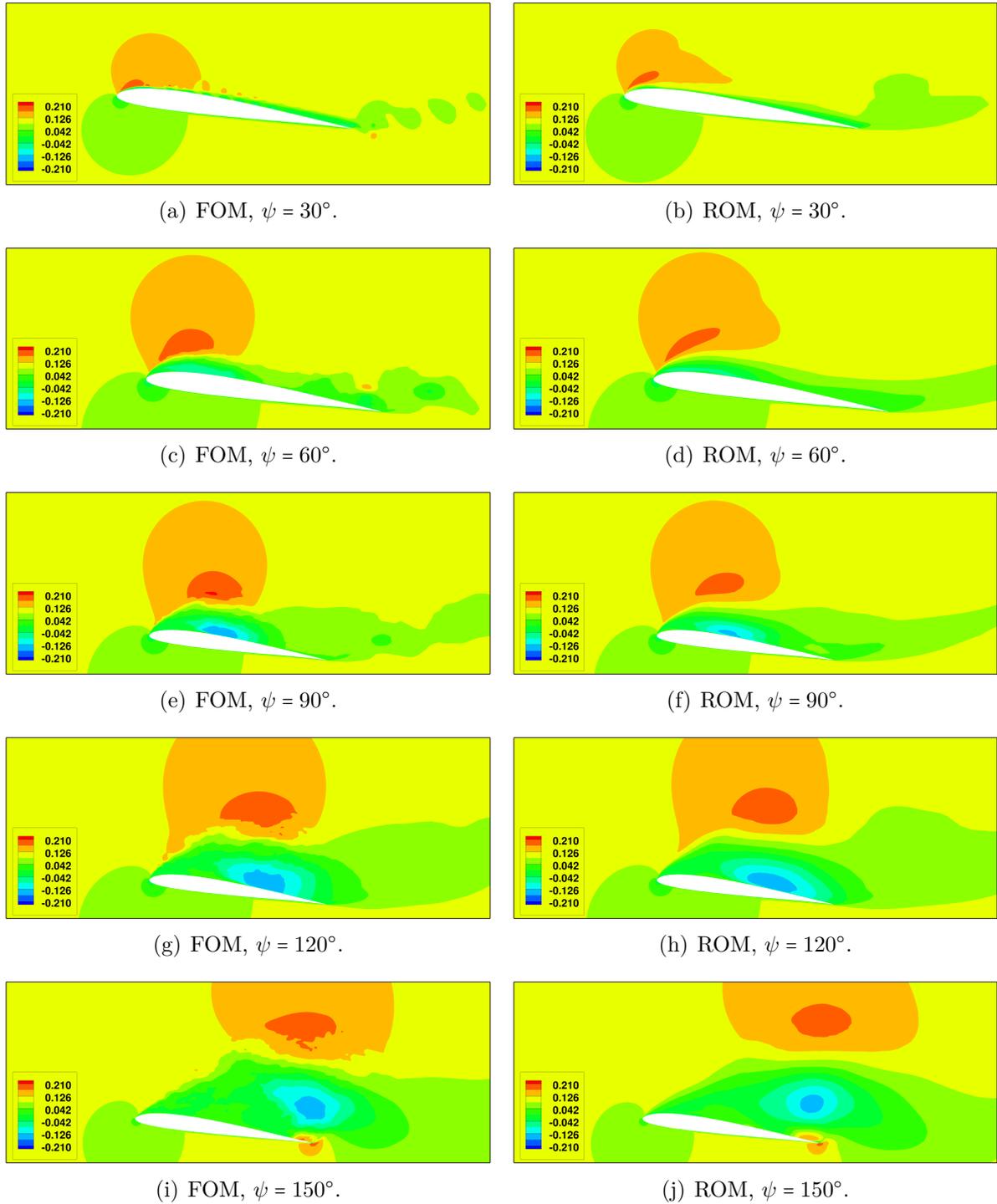
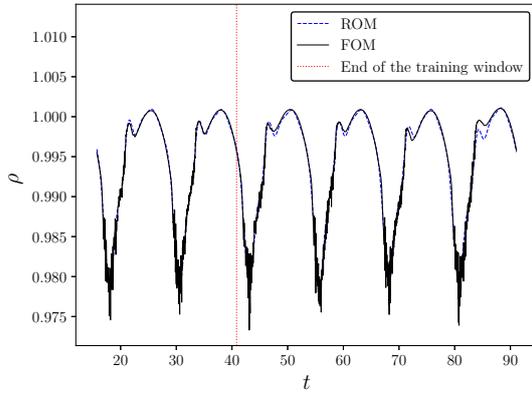
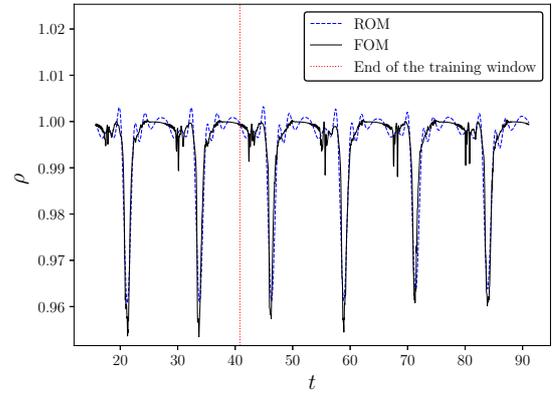


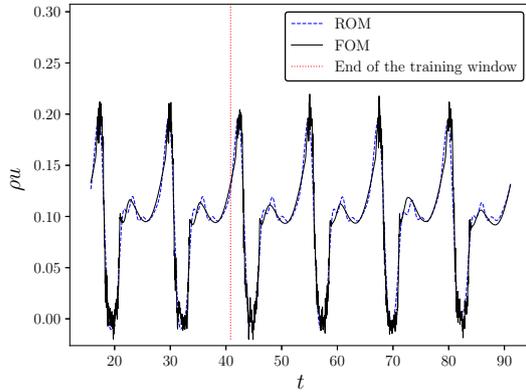
Figure 4.13: Contours of  $x$ -component of momentum at several phase angles for the fifth plunging cycle.



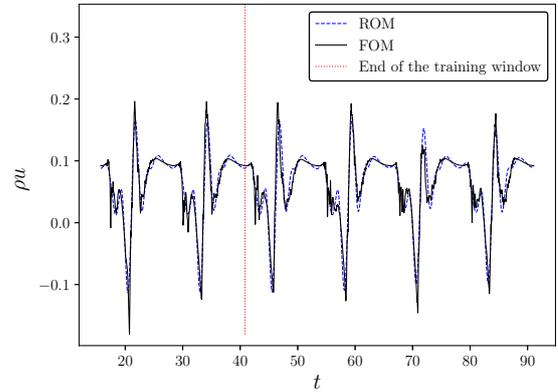
(a) Density fluctuations.



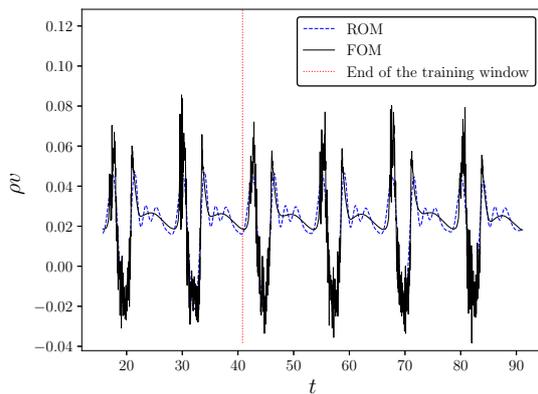
(b) Density fluctuations.



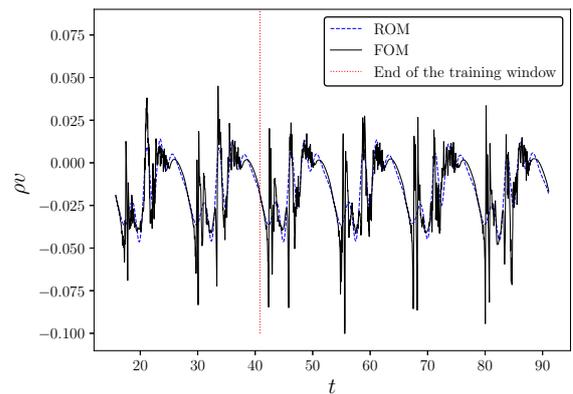
(c) Fluctuations of x-component of momentum.



(d) Fluctuations of x-component of momentum.



(e) Fluctuations of y-component of momentum.



(f) Fluctuations of y-component of momentum.

Figure 4.14: Fluctuation time histories computed by the FOM and ROM for probe locations in the proximity of the leading edge (left column) and trailing edge (right column).

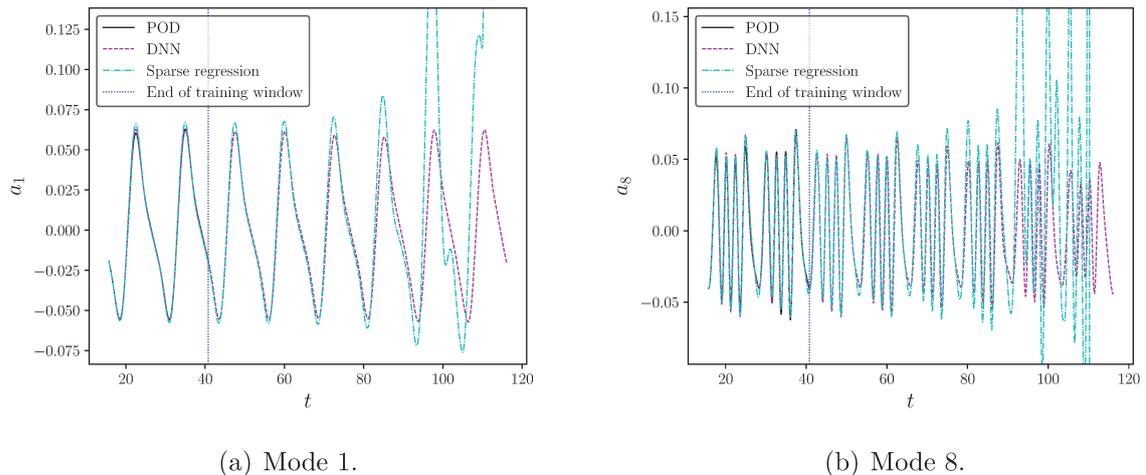


Figure 4.15: Reconstruction of SPOD temporal modes using the current DNN approach and the original SINDy algorithm with LASSO.

are computed via an explicit second-order scheme and by sixth- and tenth-order compact schemes. When the ROM is constructed using the same time step of the numerical simulation, the first 10 SPOD modes are highly resolved in time. For this case, all three schemes for computing the temporal derivatives provide accurate results and the random search is able to find a set of hyperparameters which results in an accurate ROM. However, when the DNN is trained using SPOD modes obtained by a time step five times larger than that of the simulation, the resolution of the higher POD temporal modes can become compromised. Then, it is important to use the compact schemes to obtain accurate representations of the derivatives. In this case, the random search could only find suitable ROMs using the compact schemes. As shown in figure 4.16, the models are still stable and reproduce the same dynamics observed with lower time steps. This shows that the current ROM framework is robust and can be constructed using a subset of the data of the FOM, with lower temporal resolution.

Finally, we compute the phase-averaged aerodynamic coefficients for the plunging SD7003 airfoil. Lift and drag coefficients are shown in figure 4.17 for the FOM and ROM using the six cycles computed in the training and test sets. Both the pressure and friction forces are accounted for in the calculation of the aerodynamic coefficients. However, for this case, the pressure component dominates the aerodynamic loads. The effective incidence angles are depicted in the figures and the aerodynamic coefficients are calculated using the 10th-order compact scheme for the derivatives of the temporal modes and the first 10 SPOD modes. As a validation of the current solutions, the results from Visbal (2011) are also shown. Results are presented for the ROMs computed using both the pressure and kinetic energy norms. The DNNs could find stable and accurate models

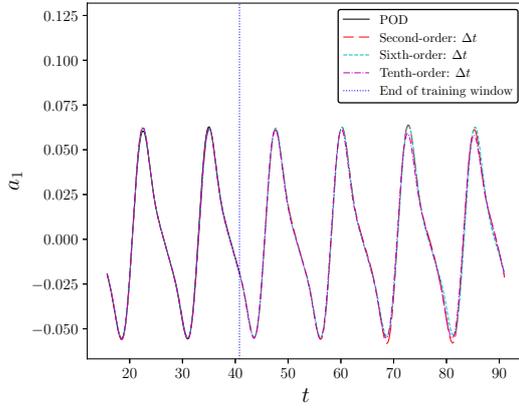
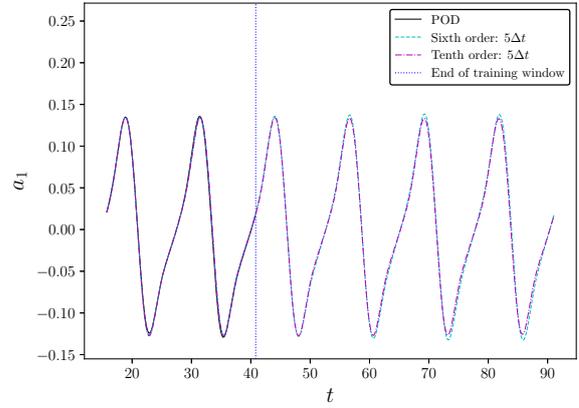
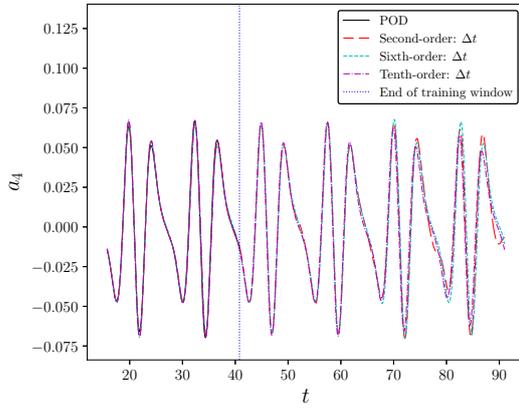
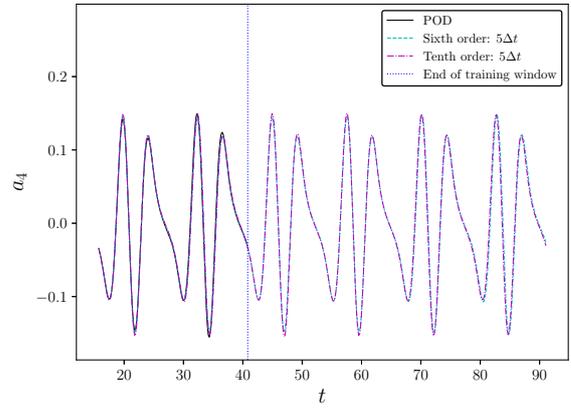
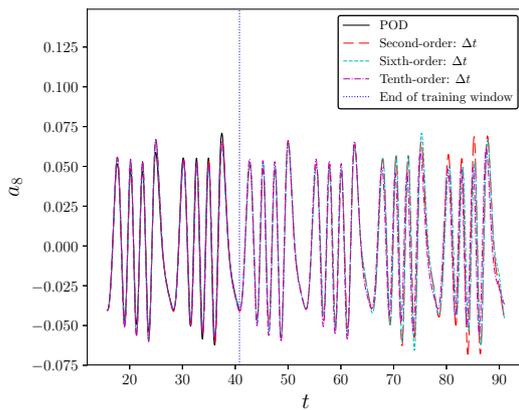
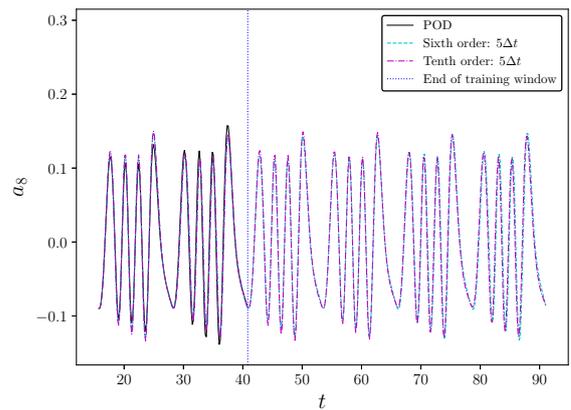
(a) Mode 1 obtained using  $\Delta t$  as in the simulation.(b) Mode 1 obtained using  $5 \times \Delta t$ .(c) Mode 4 obtained using  $\Delta t$  as in the simulation.(d) Mode 4 obtained using  $5 \times \Delta t$ .(e) Mode 8 obtained using  $\Delta t$  as in the simulation.(f) Mode 8 obtained using  $5 \times \Delta t$ .

Figure 4.16: Reconstruction of SPOD temporal modes using different finite difference schemes and time steps.

for both POD norms and the aerodynamic coefficients obtained for the best models are compared in the figure. As one can see, the present ROM solutions show good comparisons to the current large eddy simulation and Visbal (2011). The total simulation cost of the FOM for this case was 100,000 hours. On the other hand, the full cost of the ROM, including the optimization of the hyperparameters, plus the training and evaluation of 700 models, required approximately 7 hours. Hence, the computational cost for the training procedure is around 100 models per hour in a single GPU. Once a ROM is chosen, the cost of the simulation is reduced to a few seconds.

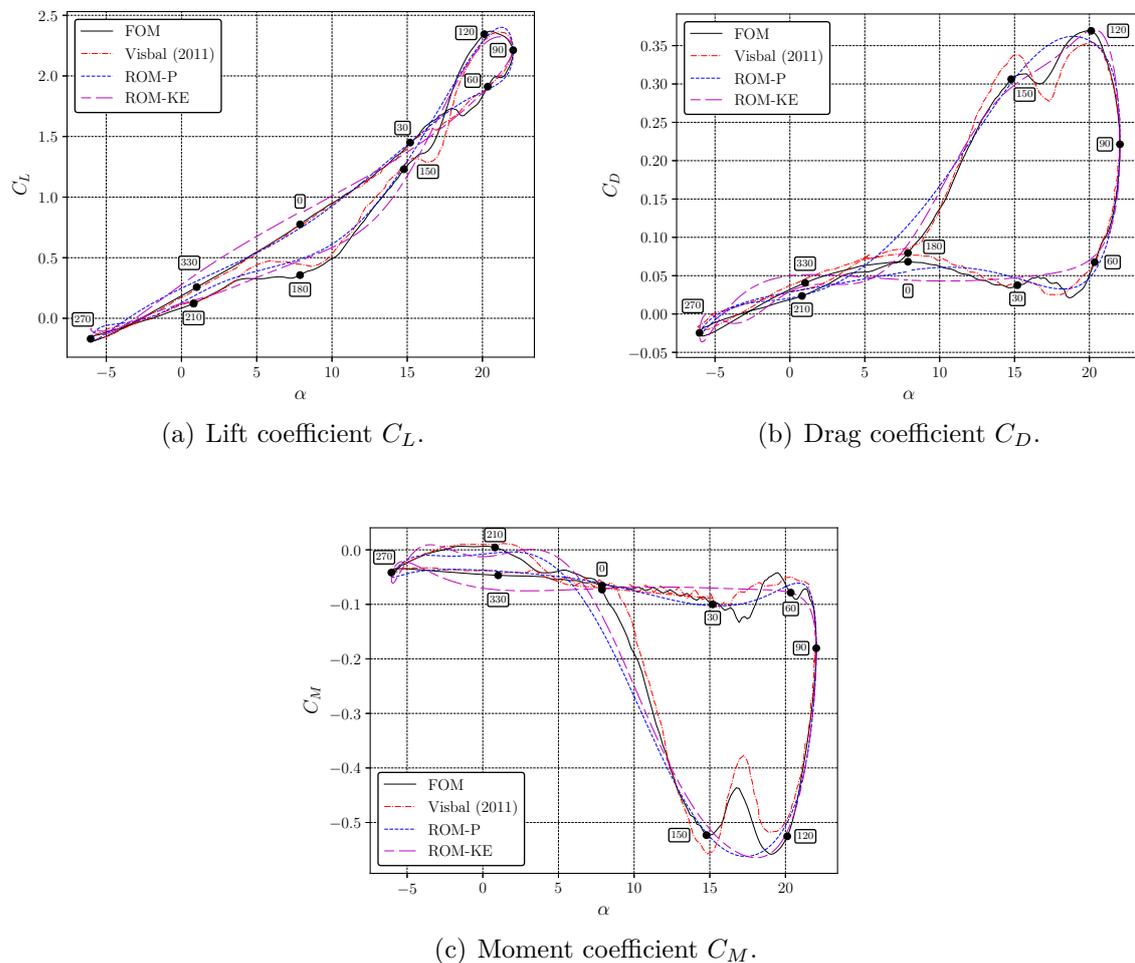


Figure 4.17: Phase-averaged aerodynamic coefficients computed by Visbal (2011), the current LES (FOM), and the DNN approach with POD norms based on pressure (P) and kinetic energy (KE).

## 5 CONCLUSIONS AND RECOMMENDATIONS

### 5.1 Summary

We present a methodology for constructing ROMs combining flow modal decomposition and regression analysis via DNNs. The framework is implemented in a context similar to that of the SINDy algorithm recently proposed in literature. The details of the methodology are described including algorithm charts which facilitate the understanding and implementation of the proposed framework. The source code can be downloaded from <http://cces.unicamp.br/software/>.

The method is tested for different problems involving nonlinear dynamical systems: the canonical nonlinear damped oscillator, the flow past a circular cylinder at a low Reynolds number and the turbulent flow past a plunging SD7003 airfoil under deep dynamic stall. For the previous two cases, the compressible Navier Stokes equations are solved in full contravariant form and additional non-inertial terms are added to the equations to simulate the airfoil plunging motion. Numerical simulations are performed using a high-order compact finite difference flow solver. Then, high fidelity results from the simulations are used as the input data for the construction of the ROMs.

To create stable and accurate ROMs of more complex flows, the application of the SPOD was found necessary. Hence, in order to filter high frequency content present in the temporal modes of the classical snapshot POD, we apply a filter function to the correlation matrix (SPOD approach). The proposed numerical framework allows the prediction of the flow field beyond the training window using larger time increments than those employed by the FOM, which demonstrates the robustness of the current ROMs constructed via DNNs. The resolution of the numerical schemes used for computation of the POD temporal mode derivatives is shown to have an important role when larger time increments are employed in the construction of the ROMs. In this case, the higher POD modes are composed by a broad range of frequencies and the accurate representation of the temporal derivatives is crucial for obtaining ROMs via regression analysis.

A discussion regarding the optimization of hyperparameters for obtaining the best ROMs via DNNs is provided together with a description of the effects of individual hyperparameters on model performance. Here, we test the random search and the Bayesian optimization, which are the most widely used procedures for hyperparameter

optimization. To report the performance of each model from a set of parameters, we compute the MAE over the training data and choose the candidate models with lower MAE values. Then, the best model is chosen based on the differences between ROM and FOM solutions over the validation set. Following this approach, the random search produced the best models at lower computational costs for all cases investigated in this work. It is worth mentioning that, in order to reduce the set of parameters for optimization, we first selected the hyperparameters related to the optimization step, such as the learning rate and the number of iterations, by manual search. We also chose the exponential linear unit activation function over the hyperbolic tangent function for the current problems since it provided results with fewer iterations.

Using 10 POD modes, 99% of the flow energy is recovered in the cylinder flow study. The ROM obtained for this case shows an excellent agreement with the FOM. Even when 2 POD modes are employed in the reconstruction, the ROM is still stable beyond the training set and captures most of the dynamics of the vortex shedding and sound wave propagation. Both the DNN and the original SINDy approaches are tested for the reconstruction of the transient regime of an incompressible flow past a cylinder. In this case, regression is performed for the 2 most energetic POD modes of the flow, obtained from Brunton *et. al* (2016). The DNN model is able to recover both the transient and limit cycle solutions of the test data for both POD modes. On the other hand, sparse regression reconstructs the transient portion of the dynamics but not the long term prediction of the limit cycle.

In the dynamic stall configuration, the flow is turbulent along the airfoil suction side, mostly during the downstroke motion. The complex flow dynamics of this case exhibit unpaired POD modes with high frequency noise. We show how the SPOD approach modifies the temporal modes for this study. Reduced-order models are constructed both for the full 3-D and the spanwise-averaged flow solutions. In both cases, the ROMs are able to capture the dynamics of the leading edge stall vortex, including its formation, transport and ejection, and of the trailing edge vortex. The total simulation cost of the FOM for this case was 100 000 hours. On the other hand, the full cost of the ROM was approximately 7 hours in a single GPU. This cost already includes the optimization of hyperparameters, plus the training and evaluation of 700 models. When the best model is chosen, the cost of the simulation is reduced to a few seconds. The data obtained by the DNN-ROMs were used to compute aerodynamic coefficients of the dynamic stall and good agreement was found compared to the LES used as the FOM. Again, we compare models obtained by DNNs and sparse regression and show that, despite the higher computational cost, the DNN approach presents good long-time predictive capabilities with stable and accurate solutions beyond the training window. On the other hand, the best model obtained from

sparse regression could learn the dynamics during the training window and also for a few cycles in the test set but its solution was not stable for long-time predictions. We expect that, in a future work, the current methodology can be further improved for applications in flow control and extrapolation of flow configurations other than those used for training.

## 5.2 Recommendations for future work

The following improvements in the current methodology would significantly enhance the ability to construct reduced-order models useful for applications in flow control and optimization.

- Further analyses need to be made for a better understanding of how to link POD-based reduced-order models with flow control techniques.
- Use of convolutional neural networks (CNNs) in the regression step may allow the construction of reduced-order models directly connected with flow variables as dimensionality reduction via POD would no longer be necessary.
- ROMs lack robustness with respect to parameter changes and therefore must be rebuilt for each new configuration. Use of an interpolation method based on Grassmann manifolds may allow the construction of reduced-order models to new set of physical parameters such as Reynolds and Mach numbers.
- Proper orthogonal decomposition can be expressed as a neural network, called an autoencoder. It is a nonlinear alternative to the POD. Autoencoders may be able to adequately capture the highly non-linear features present in turbulent flows which can lead to significant improvements for reduced-order modeling of fluid flows.
- Improvement in the hyperparameter optimization process may lead to a speed up in training time.

## References

AAMO, O.; KRSTIC, M. and SHYY, W. Flow control by feedback: Stabilization and mixing. **Applied Mechanics Reviews - APPL MECH REV**, vol. 56, 01 2003.

ABADI, M.; AGARWAL, A.; BARHAM, P.; BREVDO, E.; CHEN, Z.; CITRO, C.; CORRADO, G.S.; DAVIS, A.; DEAN, J.; DEVIN, M.; GHEMAWAT, S.; GOODFELLOW, I.J.; HARP, A.; IRVING, G.; ISARD, M.; JIA, Y.; JÓZEFOWICZ, R.; KAISER, L.; KUDLUR, M.; LEVENBERG, J.; MANÉ, D.; MONGA, R.; MOORE, S.; MURRAY, D.G.; OLAH, C.; SCHUSTER, M.; SHLENS, J.; STEINER, B.; SUTSKEVER, I.; TALWAR, K.; TUCKER, P.A.; VANHOUCKE, V.; VASUDEVAN, V.; VIÉGAS, F.B.; VINYALS, O.; WARDEN, P.; WATTENBERG, M.; WICKE, M.; YU, Y. and ZHENG, X. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. **CoRR**, vol. abs/1603.04467, 2016.

AKAIKE, H. **Information Theory and an Extension of the Maximum Likelihood Principle**, pp. 199–213. Springer New York, New York, NY, 1973.

AMES RESEARCH STAFF. Equations, tables, and charts for compressible flow. Technical Report 1135, National Advisory Committee for Aeronautics, NACA, Moffett Field, CA, USA, 1953.

ARIS, R. **Vectors, Tensors, and the Basic Equations of Fluid Mechanics**. Dover Publications, 1989.

BAIK, Y.S.; RAUSCH, J.; BERNAL, L. and OL, M. Experimental investigation of pitching and plunging airfoils at reynolds number between  $1 \times 10^4$  and  $6 \times 10^4$ . In **39th AIAA Fluid Dynamics Conference**. 2019.

BALAJEWICZ, M.; TEZAUR, I. and DOWELL, E. Minimal subspace rotation on the Stiefel manifold for stabilization and enhancement of projection-based reduced order

models for the compressible Navier-Stokes equations. **Journal of Computational Physics**, vol. 321, 224–241, 2016.

BEAM, R.M. and WARMING, R.F. An implicit factored scheme for the compressible navier-stokes equations. **AIAA Journal**, vol. 16, n. 4, 393–402, Apr 1978.

BENGIO, Y. Learning deep architectures for ai. **Found. Trends Mach. Learn.**, vol. 2, n. 1, 1–127, january 2009.

BERGSTRA, J. and BENGIO, Y. Random search for hyper-parameter optimization. **J. Mach. Learn. Res.**, vol. 13, 281–305, february 2012.

BROCHU, E.; CORA, V.M. and FREITAS, N. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. **CoRR**, vol. abs/1012.2599, 2010.

BRUNTON, S.L. and NOACK, B.R. Closed-loop turbulence control: Progress and challenges. **Applied Mechanics Reviews**, vol. 67, 050801, 2015.

BRUNTON, S.L.; PROCTOR, J.L. and KUTZ, N.J. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. **Proceedings of the National Academy of Sciences**, vol. 113, n. 15, 3932–3937, 2016.

CARLBERG, K.; BARONE, M. and ANTIL, H. Galerkin v. least-squares petrov–galerkin projection in nonlinear model reduction. **Journal of Computational Physics**, vol. 330, 693 – 734, 2017.

CARLBERG, K.; BOU-MOSLEH, C. and FARHAT, C. Efficient non-linear model reduction via a least-squares petrov–galerkin projection and compressive tensor approximations. **International Journal for Numerical Methods in Engineering**, vol. 86, n. 2, 155–181, 2011.

CARLBERG, K.; FARHAT, C.; CORTIAL, J. and AMSALLEM, D. The gnat method for nonlinear model reduction: Effective implementation and application to computational

fluid dynamics and turbulent flows. **Journal of Computational Physics**, vol. 242, 623–647, 2013.

CATTAFESTA, L.N. and SHEPLAK, M. Actuators for active flow control. **Annual Review of Fluid Mechanics**, vol. 43, n. 1, 247–272, 2011.

CAZEMIER, W.; VERSTAPPEN, R.W.C.P. and VELDMAN, A.E.P. Proper orthogonal decomposition and low-dimensional models for driven cavity flows. **Physics of Fluids**, vol. 10, n. 7, 1685–1699, 1998.

CHATURANTABUT, S. and SORENSEN, D. Nonlinear model reduction via discrete empirical interpolation. **SIAM Journal of Scientific Computing**, vol. 32, 2737–2764, 2010.

CHENEY, E.W. and KINCAID, D.R. **Numerical Mathematics and Computing**. Brooks/Cole Publishing Co., Pacific Grove, CA, USA, 6th ed., 2007. ISBN 0495114758.

CHOI, H.; JEON, W.P. and KIM, J. Control of flow over a bluff body. **Annual Review of Fluid Mechanics**, vol. 40, n. 1, 113–139, 2008.

CLAINCHE, S. and VEGA, J.M. Higher order dynamic mode decomposition to identify and extrapolate flow patterns. **Physics of Fluids**, vol. 29, 084102, 2017.

CLEVERT, D.; UNTERTHINER, T. and HOCHREITER, S. Fast and accurate deep network learning by exponential linear units (elus). **CoRR**, vol. abs/1511.07289, 2015.

CORDIER, L. and BERGMANN, M. Proper Orthogonal Decomposition: an overview. In **Lecture series 2002-04, 2003-03 and 2008-01 on post-processing of experimental and numerical data, Von Karman Institute for Fluid Dynamics, 2008.**, p. 46 pages. VKI, 2008.

DIMOTAKIS, P.E. Turbulent mixing. **Annual Review of Fluid Mechanics**, vol. 37, n. 1, 329–356, 2005.

EVERSON, R. and SIROVICH, L. Karhunen-loève procedure for gappy data. **Optical Society of America**, vol. 12, 1657–1664, 1995.

FUKAMI, K.; FUKAGATA, K. and TAIRA, K. Super-resolution reconstruction of turbulent flows with machine learning. **Journal of Fluid Mechanics**, vol. 870, 106–120, 2019.

GOODFELLOW, I.; BENGIO, Y. and COURVILLE, A. **Deep Learning**. The MIT Press, 2016. ISBN 0262035618, 9780262035613.

GREEN, M.A.; ROWLEY, C.W. and HALLER, G. Detecting vortex formation and shedding in cylinder wakes using lagrangian coherent structures. **Journal of Fluid Mechanics**, vol. 572, 111–120, 2007.

HALLER, G. Lagrangian coherent structures. **Annual Review of Fluid Mechanics**, vol. 47, 137–162, 2015.

JUANG, J.N. and PAPPA, R.S. An eigensystem realization algorithm for modal parameter identification and model reduction. **Journal of Guidance, Control, and Dynamics**, vol. 8, 620–627, 1985.

KENNEDY, C.A.; CARPENTER, M.H. and LEWIS, R.M. Low-storage, explicit Runge-Kutta schemes for the compressible Navier-Stokes equations. ICASE Report No. 99-22, 1999.

KIM, J. Physics and control of wall turbulence for drag reduction. **Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences**, vol. 369, n. 1940, 1396–1411, 2011.

KINGMA, D.P. and BA, J. Adam: A method for stochastic optimization. **Proceedings of the 3rd International Conference on Learning Representations (ICLR)**, 2014.

KOIZUMI, H.; TSUTSUMI, S. and SHIMA, E. Feedback control of karman vortex shedding from a cylinder using deep reinforcement learning. In **AIAA Aviation 2018**. AIAA, 2018.

KRIZHEVSKY, A. Learning multiple layers of features from tiny images. Technical report, 2009.

KUTZ, J. Deep learning in fluid dynamics. **Journal of Fluid Mechanics**, vol. 814, 1–4, 2017.

LELE, S.K. Compact finite difference schemes with spectral-like resolution. **Journal of Computational Physics**, vol. 103, n. 1, 16–42, 1992.

LING, J.; KURZAWSKI, A. and TEMPLETON, J. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. **Journal of Fluid Mechanics**, vol. 807, 155–166, 2016.

LING, J. and TEMPLETON, J. Evaluation of machine learning algorithms for prediction of regions of high Reynolds averaged Navier Stokes uncertainty. **Physics of Fluids**, vol. 27, 085103, 2015.

LUHAR, M.; SHARMA, A.S. and MCKEON, B.J. On the structure and origin of pressure fluctuations in wall turbulence: predictions based on the resolvent analysis. **Journal of Fluid Mechanics**, vol. 751, 38–70, 2014.

LUMLEY, J. and BLOSSEY, P. Control of turbulence. **Annual Review of Fluid Mechanics**, vol. 30, n. 1, 311–327, 1998.

LUMLEY, J.L. The structure of inhomogeneous turbulence. In **Atmospheric Turbulence and Wave Propagation**, pp. 166–178. Nauka, Moscow, 1967.

MOIN, P. and BEWLEY, T. Feedback control of turbulence. **Applied Mechanics Review**, vol. 47, S3–S13, 04 2001.

NAGARAJAN, S., **Leading edge effects in bypass transition**, PhD Thesis, Stanford University, 2004.

NAGARAJAN, S.; LELE, S.K. and FERZIGER, J.H. A robust high-order compact method for large eddy simulation. **Journal of Computational Physics**, vol. 191, n. 2, 392–419, 2003.

NAIR, V. and HINTON, G.E. Rectified linear units improve restricted boltzmann machines. In **Proceedings of the 27th International Conference on International Conference on Machine Learning**, ICML'10, pp. 807–814. Omnipress, USA, 2010. ISBN 978-1-60558-907-7.

NAVARRO, L.C.; NAVARRO, A.K.W.; ROCHA, A. and DAHAB, R. Connecting the dots: Toward accountable machine-learning printer attribution methods. **Journal of Visual Communication and Image Representation**, vol. 53, 257–272, 2018.

NOACK, B.R.; AFANASIEV, K.; MORZYNSKI, M.; TADMOR, G. and THIELE, F. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. **Journal of Fluid Mechanics**, vol. 497, 335–363, 2003.

OL, M.V.; BERNAL, L.; KANG, C. and SHYY, W. Shallow and deep dynamic stall for flapping low reynolds number airfoils. **Experiments in Fluids**, vol. 46, n. 5, 883–901, May 2009.

ÖSTH, J.; NOACK, B.R.; KRAJNOVIĆ, S.; BARROS, D. and BORÉE, J. On the need for a nonlinear subscale turbulence term in POD models as exemplified for a high-Reynolds-number flow over an Ahmed body. **Journal of Fluid Mechanics**, vol. 747, 518–544, 005 2014.

PAN, S. and DURAISAMY, K. Long-time predictive modeling of nonlinear dynamical systems using neural networks. **ArXiv e-prints**, may 2018.

PROTAS, B.; NOACK, B.R. and ÖSTH, J. Optimal nonlinear eddy viscosity in Galerkin models of turbulent flows. **Journal of Fluid Mechanics**, vol. 766, 337–367, 003 2015.

RABAULT, J.; KUCHTA, M.; JENSEN, A.; RÉGLADE, U. and CERARDI, N. Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. **Journal of Fluid Mechanics**, vol. 865, 281–302, 2019.

RAISSI, M.; PERDIKARIS, P. and KARNIADAKIS, G. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. **Journal of Computational Physics**, vol. 378, 686 – 707, 2019.

RAMOS, B.L.; WOLF, W. and TAIRA, K. High-fidelity simulation and flow control of a plunging airfoil under deep dynamic stall. In **AIAA Scitech 2019 Forum**. 2019.

RIBEIRO, J.H.M. and WOLF, W.R. Identification of coherent structures in the flow past a naca0012 airfoil via proper orthogonal decomposition. **Physics of Fluids**, vol. 29, n. 8, 085104, 2017.

ROCKWOOD, M.P.; TAIRA, K. and GREEN, M.A. Detecting vortex formation and shedding in cylinder wakes using lagrangian coherent structures. **AIAA Journal**, vol. 55, 15–23, 2016.

ROWLEY, C.W.; COLONIUS, T. and MURRAY, R.M. Model reduction for compressible flows using POD and galerkin projection. **Physica D: Nonlinear Phenomena**, vol. 189, n. 1–2, 115 – 129, 2004.

ROWLEY, C.W. and WILLIAMS, D.R. Dynamics and control of high-reynolds-number flow over open cavities. **Annual Review of Fluid Mechanics**, vol. 38, n. 1, 251–276, 2006.

RUDY, S.H.; BRUNTON, S.L.; PROCTOR, J.L. and KUTZ, N.J. Data-driven discovery of partial differential equations. **Science Advances**, vol. 3, n. 4, 2017.

RUDY, S.H.; KUTZ, J.N. and BRUNTON, S.L. Deep learning of dynamics and signal-noise decomposition with time-stepping constraints. **ArXiv e-prints**, august 2018.

SAN, O. and MAULIK, R. Extreme learning machine for reduced order modeling of turbulent geophysical flows. **Phys. Rev. E**, vol. 97, 042322, Apr 2018.

SCHMID, P.J. Dynamic mode decomposition of numerical and experimental data. **Journal of Fluid Mechanics**, vol. 656, 5–28, 2010.

SCHWARZ, G. Estimating the dimension of a model. **The Annals of Statistics**, vol. 6, 461–464, 1978.

SHARMA, A.S.; MOARREF, R.; MCKEON, B.J.; PARK, J.S.; GRAHAM, M.D. and WILLIS, A.P. Low-dimensional representations of exact coherent states of the navier-stokes equations from the resolvent model of wall turbulence. **Physical Review E**, vol. 93, 021102, 2016.

SIEBER, M.; PASCHEREIT, C.O. and OBERLEITHNER, K. Spectral proper orthogonal decomposition. **Journal of Fluid Mechanics**, vol. 792, 798–828, 2016.

SIROVICH, L. Turbulence and the dynamics of coherent structures. Part I: Coherent structures. **Quarterly of Applied Mathematics**, vol. 45, 561–571, 1986.

SNOEK, J.; LAROCHELLE, H. and ADAMS, R.P. Practical bayesian optimization of machine learning algorithms. In **Advances in Neural Information Processing Systems 25**, pp. 2951–2959. Curran Associates, Inc., 2012.

STRÖFER, C.M.; WU, J.L.; XIAO, H. and PATERSON, E. Data-driven, physics based feature extraction from fluid flow fields using convolutional neural networks. **Communications in Computational Physics**, vol. 25, 625–650, 2019.

TAIRA, K.; BRUNTON, S.L.; DAWSON, S.T.M.; ROWLEY, C.W.; COLONIUS, T.; MCKEON, B.J.; SCHMIDT, O.T.; GORDEYEV, S.; THEOFILIS, V. and UKEILEY, L.S. Modal analysis of fluid flows: An overview. **AIAA Journal**, vol. 47, 1–28, 2017.

TAIRA, K.; HEMATI, M.S.; BRUNTON, S.L.; SUN, Y.; DURAISAMY, K.; BAGHERI, S.; DAWSON, S.T.M. and YEH, C.A. Modal Analysis of Fluid Flows: Applications and Outlook. **arXiv e-prints**, arXiv:1903.05750, Mar 2019.

TIBSHIRANI, R. Regression shrinkage and selection via the lasso. **Journal of the Royal Statistical Society. Series B (Methodological)**, vol. 58, n. 1, 267–288, 1996.

TOWNE, A.; SCHMIDT, O.T. and COLONIUS, T. Spectral proper orthogonal decomposition and its relationship to dynamic mode decomposition and resolvent analysis. **Journal of Fluid Mechanics**, vol. 847, 821–867, 2018.

TU, J.H.; ROWLEY, C.W.; LUCHTENBURG, D.M.; BRUNTON, S.L. and KUTZ, N.J. On dynamic mode decomposition: Theory and applications. **Journal of Computational Dynamics**, vol. 1, 391–421, 2014.

VISBAL, M.R. Numerical investigation of deep dynamic stall of a plunging airfoil. **AIAA Journal**, vol. 49, 2152–2170, 2011.

VLACHAS, P.R.; BYEON, W.; WAN, Z.Y.; SAPSIS, T.P. and KOUMOUTSAKOS, P. Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. **Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences**, vol. 474, n. 2213, 2018.

WAN, Z.Y.; VLACHAS, P.; KOUMOUTSAKOS, P. and SAPSIS, T. Data-assisted reduced-order modeling of extreme events in complex dynamical systems. **PLOS ONE**, vol. 13, n. 5, 1–22, 05 2018.

WANG, J.X.; WU, J.L. and XIAO, H. Physics-informed machine learning approach for reconstructing reynolds stress modeling discrepancies based on dns data. **Physical Review Fluids**, vol. 2, 034603, 2017.

WOLF, W.R., **Airfoil Aeroacoustics: LES and Acoustic Analogy Predictions**, PhD Thesis, Stanford University, 2011.

WOLF, W.R.; AZEVEDO, J.L.F. and LELE, S.K. Convective effects and the role of quadrupole sources for aerofoil aeroacoustics. **Journal of Fluid Mechanics**, vol. 708, 502–538, 10 2012a.

WOLF, W.R.; LELE, S.K.; JOTHIPRASARD, G. and CHEUNG, L. Investigation of noise generated by a du96 airfoil. In **18th AIAA/CEAS Aeroacoustics Conference (33th AIAA Aeroacoustics Conference)**. 2012b.

XAVIER, G. and YOSHUA, B. In Y.W. Teh and M. Titterington, editors, **Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics**, vol. 9 of *Proceedings of Machine Learning Research*, pp. 249–256. PMLR, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010.

ZIMMERMANN, R. and WILLCOX, K. An accelerated greedy missing point estimation procedure. **SIAM Journal of Scientific Computing**, vol. 38, 2827–2850, 2016.





The back substitution step solves for  $x_n, x_{n-1}, \dots, x_1$  as follows

$$d'_n := d'_n - \frac{a'_{n-1}}{d'_{n-1}} c'_{n-1} \quad (\text{B.9})$$

$$x_i = \begin{cases} \frac{(b'_n - \frac{a'_{n-1}}{d'_{n-1}} b'_{n-1})}{d'_n} & i = n \\ \frac{(b'_{n-1} - c'_{n-1} x_n)}{d'_{n-1}} & i = n - 1 \\ \frac{(b'_i - f'_i x_{i+2} - c'_i x_{i+1})}{d'_i} & i = n - 2, n - 3, \dots, 1 . \end{cases} \quad (\text{B.10})$$

Further details regarding the present algorithm can be found in Cheney and Kincaid (2007).

## APPENDIX C : LOW-STORAGE RUNGE-KUTTA

Here, we consider an initial value problem

$$\frac{d\mathbf{U}}{dt} = \mathbf{F}(\mathbf{U}, t) ; \quad \mathbf{U}(t_o) = \mathbf{U}_o . \quad (\text{C.1})$$

The discrete approximation can be made with an explicit five-step fourth-order Runge-Kutta method Kennedy *et. al* (1999). The scheme is

$$d\mathbf{U}^\eta = \Delta t \mathbf{F}(\mathbf{U}^\eta, t^\eta) + A^\eta d\mathbf{U}^{\eta-1} \quad \eta = 1, \dots, 5 \quad (\text{C.2})$$

$$\mathbf{U}^\eta = \mathbf{U}^{\eta-1} + B^\eta d\mathbf{U}^\eta \quad (\text{C.3})$$

where  $\Delta t$  is the time step,  $\eta$  is the RK4 stage, and  $A^\eta$  and  $B^\eta$  are constants defined below. Only the  $d\mathbf{U}$  and  $\mathbf{U}$  vector must be stored, which results in a low-storage algorithm.

$$\begin{aligned} A^1 &= 0 \\ A^2 &= -6234157559845/12983515589748 \\ A^3 &= -6194124222391/4410992767914 \\ A^4 &= -31623096876824/15682348800105 \\ A^5 &= -12251185447671/11596622555746 \\ B^1 &= 494393426753/4806282396855 \\ B^2 &= 4047970641027/5463924506627 \\ B^3 &= 9795748752853/13190207949281 \\ B^4 &= 4009051133189/8539092990294 \\ B^5 &= 1348533437543/7166442652324 . \end{aligned}$$

The time  $t^\eta$  for which the solution advances after each substep is

$$t^\eta = t + C^\eta \Delta t \quad (\text{C.4})$$

where  $C^\eta$  is

$$\begin{aligned} C^1 &= 494393426753/4806282396855 \\ C^2 &= 4702696611523/9636871101405 \\ C^3 &= 3614488396635/5249666457482 \\ C^4 &= 9766892798963/10823461281321 \\ C^5 &= 1 . \end{aligned}$$