

# Prediction of flow-fields by combining high-fidelity CFD data and Machine-Learning algorithms

S. Arntzen

Delft University of Technology





# Prediction of flow-fields by combining high-fidelity CFD data and Machine-Learning algorithms

by

S. Arntzen

to obtain the degree of Master of Science  
at the Delft University of Technology,  
to be defended publicly on Friday November 15, 2019 at 09:30 PM.

Student name:	Stefan Arntzen
Student number:	450568
Project duration:	October 15, 2018 – November 15, 2019
Daily supervisor:	Dr.ir. S.Hermeth, Siemens AG
Thesis committee:	Prof.dr.ir. S.Klein, TU Delft   Chair
	Prof.dr.ir. B.J Boersma, TU Delft
	Dr.ir. R.Pechnik, TU Delft
	Ir. S.H.J.J.Smit, TU Delft

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.



# Preface

Dear Reader,

With this thesis document, I close a two-year chapter at the Delft University of Technology and 17 months with Siemens AG in Mülheim an der Ruhr and Berlin. Both institutions are renowned for their high standards and contribution to innovation. These experiences resulted in the rapid development of my theoretical and practical knowledge in the field of Energy and Flow technology. For that, I am incredibly grateful.

I would like to reflect on the people who have supported this research. First and foremost, I like to express my gratitude to Prof.Klein for supervising my internship and thesis. Moreover, thank you for connecting me to Siemens, your support and valuable feedback. In addition, I would like to thank Dr.Pechnik and Ir.Smit for the fruitful discussions and feedback.

For the daily supervision, feedback and answering my numerous CFD questions, thank you Dr.Hermeth. I am grateful to Dr.Panek for the conversations on thermoacoustic, military dolphins that parachute and revealing hidden gems in Berlin. Furthermore, my gratitude goes out to Dr.Rajaram, Dr.Chen, Dr.Steiner and Dr.Wegner for the numerous hours of sparring and exchanging ideas. To all the proof-readers, your help and support is mostly appreciated.

Lastly, I would like to thank my girlfriend, brother and friends for their support and patience. To my parents, you have my eternal gratitude for your support, encouragements and wise words.

*Stefan Arntzen  
Berlin, November 2019*





# Abstract

High-fidelity optimisation studies are a useful asset in the design of critical components for large gas turbines. These studies require the computation of numerous computationally expensive CFD simulations and result in predominantly optimisation graphs of design objectives (e.g. Pareto-front figure). The quantity of generated data is intractable and therefore in practice not stored. All the physical insight into the flow-fields is lost and no data can be accessed nor any adjustments can be made. This work researches the combination of CFD simulations and Machine Learning (ML) algorithms to simultaneously: 1) reduce computational cost and time, 2) retain physical insight by focusing on the prediction of flow-fields and 3) keep the ability to access information or to make adjustments.

This research starts with the identification of 19 potential holding ML-algorithms. A suitable algorithm is chosen with the help of 4 criteria, that is referred to as the Proper-Orthogonal Decomposition (POD) - Interpolation algorithm. In studied literature it reportedly provided very high speed-up times, good accuracy, additional learning benefit and is relative simple. The POD-Interpolation algorithm is implemented on two test cases: the Von Karman Vortex Street and a Shear Mixing Layer. The results are summarised for both as unphysical and highly inaccurate. Two root-causes are identified: the first is an invalid assumption in the methodology and the second dominant root-cause is a fundamental flaw. Interpolation can never accurately predict dominant frequencies in a flow-field.

With the inability of the POD-Interpolation algorithm to predict physical accurate flow-fields, a self-proposed algorithm is implemented. This is referred to as the Sparse-Dynamic-Mode-Decomposition (sDMD) - Scaling method. This method relies on accurate scaling relations to predict flow-fields. However, almost no application cases have known scaling relations. The sDMD-Scaling algorithm is capable of discovering and using these newly discovered scaling relations to predict flow-fields. This is proven for the Von Karman Vortex Street case with self-made scaling relations: the Mean Residual Error of the predicted flow-fields rarely exceeds 7%. An error reduction factor of  $\approx 10$  is achieved when compared to the POD-Interpolation approach. The design space from the test case is computed  $\approx 5.6$  times faster with the CFD and sDMD-Scaling combination than with the conventional CFD approach. Furthermore, any new design point can be computed  $\approx 309$  times faster with the trained sDMD-Scaling algorithm than with a CFD simulation.

The sDMD-Scaling algorithm proves that it is possible to combine CFD and ML to achieve the 3 aforementioned goals for a canonical fluid dynamic test case. Several recommendations are presented to improve the sDMD-Scaling algorithm for more complex cases. However, given the difficulties and efforts that went into the relative straightforward test cases, it is difficult to foresee any large-scale implementation of any CFD and ML combination for predicting flow-fields.



# Contents

<b>List of figures</b>	<b>xi</b>
<b>List of tables</b>	<b>xiii</b>
<b>Nomenclature</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Problem description . . . . .	3
1.3 Research questions . . . . .	4
1.4 Report outline . . . . .	5
<b>2 Theoretical background</b>	<b>7</b>
2.1 Fundamentals of fluid dynamics . . . . .	7
2.1.1 Conservation equations . . . . .	7
2.1.2 Turbulence . . . . .	9
2.2 Fundamentals of Computational Fluid Dynamics . . . . .	11
2.2.1 Large Eddy Simulation . . . . .	12
2.2.2 Discretization . . . . .	13
2.3 Conclusions theoretical background . . . . .	14
<b>3 Literature review</b>	<b>15</b>
3.1 What is Machine Learning? . . . . .	15
3.2 Machine Learning for fluid dynamics . . . . .	16
3.2.1 Flow control . . . . .	17
3.2.2 Identification of essential dynamical features . . . . .	18
3.3 Machine Learning for prediction of flow-fields . . . . .	19
3.3.1 Reduced order models . . . . .	19
3.3.2 Parameterized Reduced Order Models . . . . .	20
3.3.3 Neural Networks . . . . .	22
3.3.4 Regression methods . . . . .	22
3.3.5 Selection . . . . .	23
3.4 Modal Decomposition Methods . . . . .	26
3.4.1 Proper-orthogonal decomposition . . . . .	26
3.4.2 Dynamic mode decomposition . . . . .	29
3.4.3 Comparison algorithms: POD versus DMD . . . . .	32
3.5 What this work adds . . . . .	32
3.6 Conclusions literature review . . . . .	33
<b>4 Application cases</b>	<b>35</b>
4.1 Von Karman Vortex Street . . . . .	35
4.1.1 CFD set-up . . . . .	37
4.1.2 Results . . . . .	39
4.2 Turbulent Shear Mixing Layer . . . . .	40
4.2.1 CFD set-up . . . . .	40
4.3 Consequences of quasi two-dimensional CFD . . . . .	44
4.4 Conclusions application cases . . . . .	44

<b>5</b>	<b>POD-Interpolation</b>	<b>45</b>
5.1	Methodology . . . . .	46
5.2	Von Karman Vortex Street . . . . .	47
5.2.1	Flow-field decomposition . . . . .	47
5.2.2	Dimension reduction . . . . .	49
5.2.3	Prediction and reconstruction . . . . .	50
5.2.4	Results . . . . .	51
5.3	Turbulent shear mixing layer . . . . .	55
5.3.1	Flow-field decomposition . . . . .	56
5.3.2	Results . . . . .	58
5.4	Discussion results . . . . .	60
5.4.1	A (lack of a) common set of spatial modes . . . . .	60
5.4.2	Interpolation. . . . .	61
5.4.3	Prediction of the downstream wake . . . . .	63
5.4.4	Attempts to overcome the limitations . . . . .	63
5.5	Conclusions POD-Interpolation . . . . .	64
<b>6</b>	<b>Sparse DMD-Scaling</b>	<b>65</b>
6.1	Methodology . . . . .	65
6.2	Von Karman Vortex Street . . . . .	66
6.2.1	Mode selection, truncation and visualisation . . . . .	66
6.2.2	Prediction and reconstruction . . . . .	68
6.2.3	Results . . . . .	69
6.3	Extending the algorithm . . . . .	72
6.3.1	Mode amplitudes . . . . .	73
6.3.2	Temporal coefficients . . . . .	73
6.3.3	Spatial modes . . . . .	75
6.3.4	Results . . . . .	75
6.3.5	Potential speed-up . . . . .	76
6.4	Discussion results . . . . .	77
6.4.1	POD-Interpolation vs sDMD-Scaling . . . . .	77
6.4.2	Potential. . . . .	77
6.5	Conclusions Sparse DMD-Scaling. . . . .	79
<b>7</b>	<b>Conclusions and recommendations</b>	<b>81</b>
7.1	Conclusions. . . . .	81
7.1.1	Machine Learning algorithms . . . . .	81
7.1.2	Assumptions . . . . .	82
7.1.3	Potential speed-up times. . . . .	83
7.1.4	Accuracy losses . . . . .	83
7.1.5	Limitations . . . . .	83
7.1.6	Error quantification . . . . .	83
7.1.7	Main conclusion . . . . .	84
7.2	Recommendations . . . . .	85
7.2.1	Future work . . . . .	85
7.2.2	Discussion on CFD and Machine Learning . . . . .	85
<b>A</b>	<b>Appendices - Chapter 2</b>	<b>89</b>
A.1	Reynolds-Averaged Navier-Stokes . . . . .	89
<b>B</b>	<b>Appendices - Chapter 3</b>	<b>91</b>
B.1	Design of Experiments . . . . .	91
B.2	A brief history of Machine Learning . . . . .	92
B.3	Additional Modal Decomposition Methods . . . . .	93
<b>C</b>	<b>Appendices - Chapter 4</b>	<b>95</b>
C.1	CFD Physics settings. . . . .	95
C.2	Pope and CFL criteria . . . . .	95
C.3	Mesh study Mixing Layer. . . . .	98



---

<b>D Appendices - Chapter 5</b>	<b>99</b>
D.1 Probe points . . . . .	99
D.2 Error quantification . . . . .	101
<b>Bibliography</b>	<b>103</b>



# List of figures

1.1	Arbitrary 3D design space . . . . .	3
2.1	A water jet which transitions from laminar to turbulent . . . . .	9
2.2	Ranking of different CFD solvers . . . . .	11
2.3	Comparison between an actual turbulent flow and its computed LES version . . . . .	12
3.1	Development and distribution of ML patents . . . . .	15
3.2	Categorisation of ML groups and algorithms . . . . .	16
3.3	Illustration of the shallow-NN decoder . . . . .	18
3.4	3-D flow-field prediction: representing the iso-surfaces of the Q-criterion . . . . .	19
3.5	Comparison between a high-fidelity CFD and Machine Learning flow-field . . . . .	20
3.6	Methodology proposed by Mak et al. . . . .	21
3.7	Comparison between the CFD flow-field and the CNN approximation . . . . .	22
3.8	POD analysis of a von Karman Vortex Street . . . . .	27
3.9	Comparison between two dominant POD and DMD modes for a 3D turbulent flow over a flat-plate wing . . . . .	30
4.1	Wind-driven clouds in the shape of a Vortex Street . . . . .	35
4.2	The Strouhal-Reynolds relation for rigid circular cylinders . . . . .	36
4.3	Cross-section of the mesh used for Karman Vortex Street CFD simulations . . . . .	37
4.4	Residuals and normalised averaged values of quantities of interest . . . . .	38
4.5	Six snapshots of a simulated von Karman flow-field . . . . .	39
4.6	Comparison between the CFD and experimental vortex shedding frequency . . . . .	39
4.7	Visualization of a turbulent shear mixing layer. . . . .	40
4.8	Mesh used for the Mixing Layer cases . . . . .	41
4.9	Six snapshots of the Mixing Layer base case . . . . .	42
4.10	Velocity profiles of line probes at different axial positions . . . . .	42
4.11	Pressure probe comparison of inside and outside the wake . . . . .	43
5.1	The essence of the POD-Interpolation algorithm . . . . .	46
5.2	The mean flow-field and 1 <sup>st</sup> spatial mode shown for 2 operating points . . . . .	47
5.3	The 9 <sup>th</sup> mode for operating point 8 and 13 . . . . .	48
5.4	Variance plots for training point 13 . . . . .	49
5.5	Time averaged MRE that compares the reconstruction quality to the amount of modes . . . . .	49
5.5	Temporal coefficients for training point 1 and 13 and objective point 8 . . . . .	51
5.6	Mean pressure flow-fields for objective points 5 and 8 . . . . .	52
5.7	Fluctuating flow-field comparison for objective point 8 . . . . .	52
5.8	Pressure probe signals at two different position for objective point 8 . . . . .	53
5.9	Pressure probe spectra at two different positions for objective point 8 . . . . .	53
5.10	Pressure spectra for objective point 4 and 6 . . . . .	54
5.11	RMS error for operating point 8 . . . . .	54
5.12	Mean Residual Error for the POD-Interpolation approach . . . . .	55
5.13	Spatial modes and temporal coefficients for operating point 1 . . . . .	56
5.14	Mean flow-field, and the the 1 <sup>st</sup> spatial mode and temporal coefficients . . . . .	57
5.15	Mean flow-field comparison for operating point 3 . . . . .	58
5.16	Comparison LES flow-field and predicted ML flow-field . . . . .	58
5.17	Pressure probe figures for operating point 2 and 5 . . . . .	59
5.18	Spatial modes comparison for the von Karman case . . . . .	60
5.19	Simplified example of interpolation of temporal coefficients . . . . .	61

5.20	Two time-steps of the predicted flow-field . . . . .	62
5.21	Time averaged MRE that compares the prediction quality to the amount of modes . . .	63
6.1	sDMD Mode selection for operating points 1 and 13 . . . . .	66
6.2	sDMD mode selection, single-sided spectrum . . . . .	67
6.3	Spatial modes for operating point 13 . . . . .	67
6.4	Initial scaling relations for the Von Karman Vortex Street . . . . .	68
6.5	Comparison between the LES and sDMD-Scaling flow-field . . . . .	69
6.6	Comparison between the LES and sDMD-Scaling flow-field . . . . .	70
6.7	Pressure signal and spectra at two different probe positions, objective point 8 . . . .	71
6.8	The pressure spectra for operating point 4 and 6 . . . . .	71
6.9	Mean Residual Error for the two governing algorithms . . . . .	72
6.10	Mode amplitudes for the four leading mode pairs . . . . .	73
6.11	Frequency of the 20 leading modes with respect to variations in the flow parameter .	74
6.12	The decay rates of the leading modes . . . . .	74
6.13	Spatial modes corresponding to the fundamental frequency . . . . .	75
6.14	Comparison of the MRE for the governing ML algorithms . . . . .	75
6.15	The pressure spectra for operating point 8 . . . . .	76
6.16	Two representative applications cases. . . . .	78
A.1	Temporal development of a turbulent flow field . . . . .	89
A.2	Comparison between an actual turbulent flow and its computed RANS version . . . .	90
C.1	Cross-section mesh for the Von Karman Vortex Street . . . . .	95
C.2	Pope for the base-case of the V.Karman Street. . . . .	96
C.3	CFL for the base-case of the V.Karman Street. . . . .	96
C.4	Cross-section mesh for the Mixing Layer . . . . .	97
C.5	Pope for the base-case of the Mixing Layer . . . . .	97
C.6	CFL for the base-case of the Mixing layer . . . . .	97
C.7	Velocity profiles for 3 different meshes . . . . .	98
D.1	5 probe points indicated by the red dots, in the Von Karman domain . . . . .	99
D.2	Different defined domains within the Von Karman geometry . . . . .	99
D.3	5 probe points indicated by the red dots, in the Mixing Layer domain . . . . .	100
D.4	Comparison between a high-fidelity CFD and Machine Learning flow-field . . . . .	101



# List of tables

- 2.1 Properties of the macro- and microstructure . . . . . 10
- 2.2 Discretization schemes used throughout this work . . . . . 13
  
- 3.1 Table of potential algorithms . . . . . 24
  
- 4.1 Domain Von Karman Vortex Street . . . . . 37
- 4.2 Mesh statistics Von Karman Vortex Street . . . . . 37
- 4.3 Design space of the von Karman Vortex Street application case . . . . . 38
- 4.4 Domain Mixing Layer . . . . . 41
- 4.5 Mesh statistics Mixing Layer . . . . . 41
- 4.6 Design space for the Turbulent Shear Mixing Layer application case . . . . . 41
  
- 5.1 Design points with their respective training points for the von Karman Vortex Street . 50
- 5.2 Objective points with their respective training points . . . . . 55
  
- 6.1 Estimated times for every computational step . . . . . 76
  
- C.1 Physics models for both the von Karman Vortex Street and Mixing Layer . . . . . 95
  
- D.1 Probe points exact position in the Von Karman domain . . . . . 99
- D.2 Probe points exact position in the Mixing Layer domain . . . . . 100



# Nomenclature

The following list provides an overview of the symbols in this thesis, their meaning and their unit.

## Acronyms

ADMM	Alternating-Direction Method of Multipliers
ANN	Artificial Neural Network
BPOD	Balanced Proper-Orthogonal Decomposition
CAD	Computational Aided Design
CFD	Computational Fluid Dynamics
CFL	Courant-Friedrichs-Lewy
CNN	Convolutional Neural Network
CPOD	Common Proper-orthogonal decomposition
DEIM	Discrete Empirical Interpolation Method
DMD	Dynamic mode decomposition
DNN	Deep Neural Network
DNS	Direct Navier Stokes
DoE	Design of Experiments
FD	Finite Differences
FE	Finite Elements
FEM	Finite Element Method
FFT	Fast Fourier Transform
FOM	Full Order Model
FV	Finite Volumes
GA	Genetic Algorithm
GIF	Graphics Interchange Format
GT	Gas Turbine
KH	Kelvin-Helmholtz
KSPOD	Kernel Smoothed Proper-orthogonal decomposition
LES	Large Eddy Simulations
LSSVR	Least-Squares Support Vector Regression
MDM	Modal decomposition method
ML	Machine Learning

NN	Neural Network
NS	Navier Stokes
ODE	Ordinary Differential Equations
P.P	Probe Points
PCA	Principal Component Analysis
PDE	Partial Differential Equations
POD	Proper-Orthogonal Decomposition
pROM	Parametrized Reduced Order Model
RANS	Reynolds-Averaged Navier Stokes
ROM	Reduced Order Model
sDMD	Sparse Dynamic Mode Decomposition
SGS	Sub Grid Scales
SHERPA	Simultaneous Hybrid Exploration that is Robust, Progressive, and Adaptive
SINDY	Sparse-Identification of Nonlinear-Dynamical Systems
SVD	Singular Value Decomposition
URANS	Unsteady Reynolds-Averaged Navier Stokes
WALE	Wall Adapting Local Eddy-Viscosity

**Roman**

$A$	Surface area	$m^2$
$f$	Frequency	$1/s$
$g$	Acceleration due to gravity	$m^2/s$
$K, k$	Kinetic Energy	J
$\dot{m}$	Mass flow	$kg/s$
$P, p$	Pressure	Pa
$t$	Time	s
$\mathbf{u}, u_i$	Velocity	$m/s$
$V$	Volume	$m^3$
$x$	First Cartesian coordinate	
$y$	Second Cartesian coordinate	
$z$	Third Cartesian coordinate	

**Greek**

$\Delta$	Grid size	
$\delta$	Kronecker delta	
$\epsilon$	Dissipation rate	$J/(kg \cdot s)$



$\eta$	Kolmogorov length scale	
$\gamma$	Regularization parameter	
$\Lambda$	Eigenvalues	
$\lambda$	Variance content	
$\mu$	Dynamic viscosity	Pa · s
$\nu$	Kinematic viscosity	m <sup>2</sup> /s
$\boldsymbol{\omega}, \omega_i$	Vorticity	1/s
$\phi$	(Spatial) modes / deterministic function	
$\rho$	Density	kg/m <sup>3</sup>
T	Temporal average	

**Other**

<b>A</b>	Mapping function / best-fit operator
<i>a</i>	(Temporal) Coefficients
<i>c</i>	Generic function quantity
<i>c<sub>set</sub></i>	Criteria for captured variance/energy
<b>D</b>	Sequential set of data vectors
<i>G</i>	Filter function
$\mathcal{L}$	Characteristics length scale
<i>l</i>	Index number
$L^2$	Euclidean norm / least squares norm
<i>M</i>	Total amount of modes
<i>m</i>	(Spatial/Temporal/Amplitude) Mode
<i>M, m</i>	All available modes
<i>c</i>	Generic function quantity
<b>R, R<sub>ij</sub></b>	Correlation tensor
<i>R</i>	Pope criterion
<i>r</i>	Rank of truncation
<b>S, S<sub>ij</sub></b>	Volumetric strain rate tensor
<b>T, T<sub>ij</sub></b>	Stress tensor
u-D	1, 2, 3...u -Dimensional
<i>W</i>	Eigenvectors

**Subscripts**

$\square_F$	Frobenius norm
$\square_i$	First Cartesian coordinate

$\square_j$	Second Cartesian coordinate
$\square_k$	Third Cartesian coordinate
$\square_m$	Mode index
$\square_r$	Resolved
$\square_{SGS}$	Sub Grid Scales

**Superscripts**

$\square'$	Fluctuations
$\square^*$	Conjugate
$\square^i$	Iteration
$\check{\square}$	Dimension reduced matrix
$\overline{\square}$	Averaged
$\tilde{\square}$	Filtered quantity
$\vec{\square}$	Vector notation

**Dimensionless numbers**

Re	Reynolds number
St	Strouhall number

# 1

## Introduction

Chapter 1 serves as an introduction to the topic and provides general information about the project. The first section provides the context and motivation behind this thesis project. Section 1.2 gives a detailed overview of the problem and touches upon the first theoretical concepts underlying the problem. Thereafter, the research questions and goals are tabulated in section 1.3. Whereas, the last section provides the outline of this thesis document.

### 1.1. Motivation

Over the past two decades Computational Fluid Dynamics (CFD) has seen a significant expansion in terms of both the number of researchers active in the field and industrial users. Engineering companies have incorporated it as a valuable tool and more commercial and open-source CFD packages are becoming available [24]. CFD is a numerical tool that allows the user to perform 'numerical experiments' for: 1) learning about complicated physical processes, 2) design and 3) optimisation purposes. The growth of CFD is enabled by the enormous increase in computational power, that in the period of 1965 and 2015 increased one-trillion-fold [94]. Anno 2019, CFD is widely used for many different physical problems of varying degrees of complexities. Regardless of the enormous growth of computational power, due to the progression in scientific knowledge the forthcoming research is spend on physical problems that push the boundaries of what is computationally possible by demanding higher accuracy results for more complex geometries and physics. Simultaneously CFD is at a stage where its relatively cost-effective, compared to experimental approaches, and therefore its usage will most-likely continue to grow.

Siemens AG is a leading company in the fabrication of large land-based gas turbines and is persistently looking for new innovative methods to help push the development of the turbines. A tool that helps them in this development is CFD. It has become indispensable for learning about the complicated multi-physics phenomena occurring in gas turbines, and for design and optimisation purposes. Common simulation topics for Siemens includes mixing of fuel and air, combustion and thermoacoustics that all require high-fidelity CFD simulations for results that accurately describe the real conditions. CFD enables Siemens to improve their product and to meet the wishes and challenges given by the industry. The biggest challenge for all gas turbine manufacturers is producing more efficient and cleaner turbines, due to the increasingly stringent emission regulations. A critical step, to meet the regulations and minimising the emissions, is optimising the design of fuel nozzles. In-house studies [10, 36, 71] have shown that a 40% reduction in  $\text{NO}_x$  emission can be achieved by having an optimal design that enhances the fuel-air mixedness, where mixedness or unmixedness is defined as a measure of the homogeneity or quality of the fuel-air mixture. However, optimisation studies (e.g. optimising the design of a fuel-air nozzle) require numerous high-fidelity CFD simulations that 1) are computationally expensive, 2) time-consuming and 3) create (extremely) large quantities of data. To showcase a typical optimisation study an example is given on the next page.

**Example 1:**

The fuel nozzles of a new combustion chamber design must be optimised. A nozzle has 5 arbitrary design parameters (e.g. length and/or diameter). The required number of CFD simulations, for the study, is strongly dependent on the number of parameters and the number of values each parameter accounts for. For the sake of brevity, let's suppose the study requires 150 high-accuracy CFD simulations to obtain reliable results. The computation of all these simulations requires a computing grid of 3000 CPU's. The computational time is in total 4 weeks (672 hours) and the cost of 1 CPU per hour is 0.05 Euro, which results in a total cost of  $\pm 100.000$  Euro. The time and cost are very high, especially considering that this only includes the computational aspect and not any pre- and post-processing work. Storing the computed data from 150 high-fidelity CFD simulations is intractable and only key figures and optimisation figures are retained.

The example is kept intentionally vague but still provides a good representation of a typical design and optimisation study for a critical component. Example 1, highlights the following problems.

1. **These studies require an enormous amount of resources.** Often computational time and cost of 1 CFD simulation can already be too high, but this is amplified when dealing with a multitude of CFD simulations.
2. **The generation of intractable amounts of data.** To store the computed data is unrealistic and would result in excessive data-storage costs. Therefore, resulting from optimisation studies are not the flow-fields (i.e. computed data) but only key figures from each CFD simulation and optimisation figures (e.g. convergence, Pareto-front and correlation figures).

As a direct consequence of not storing the data, the following problems also need to be addressed.

3. **Loss of physical insight.** Where physical insight is defined as: the possibility to see how the computed flow-field evolves over time for every simulation with different design parameters.
4. **The inability to access information or make adjustments.**

Underlying these optimisation studies are certain algorithms, Example 1 describes a study that is based on a Design of Experiments (DoE) [4, 26]. But others such as SHERPA [138], Genetic Algorithms (GA) [26] and Gradient-Based algorithms are also available to Siemens. The different algorithms can reduce the total computation cost and time of optimisation studies. However they cannot overcome the other issues. Realistically only black-box type of optimisation studies are currently available, which are expensive and simultaneously our physical insight is lost. Siemens provided the possibility for this Master thesis project to look for new innovative approaches to mitigate these 4 problems simultaneously that are adjoined to optimisation studies. Section 1.2 provides a more detailed problem description.

The innovative approaches that are researched during this project are categorised under the comprehensive term of "Machine Learning" (ML), that is an umbrella term for a wide variety of algorithms that can serve many different purposes. The field of ML has seen a dramatic growth in exposure in the recent years and are increasingly being implemented in different applications. Common examples include speech applications (Siri - Apple), traffic predictions (Google Maps) but maybe mostly appreciated for its email-spam filtering capabilities (Microsoft Outlook).

To talk about ML in a general sense is inherently flawed due to the amount and large degree of variation between algorithms. Regardless, what generally can be said is that many of the mathematical working principles of ML algorithms predate the time of computers and stems from statistics. Seminal work performed by T. Bayes [9] in the 18<sup>th</sup> century, A.M. Legendre [81] in the 19<sup>th</sup> century and A. Markov [91] in the 20<sup>th</sup> century contributed significantly to the current-day ML algorithms. Since then, the ongoing evolution of ML algorithms are primarily driven [24] by the following factors: 1) the increasing quantity of data, 2) advances in computational power, data-storage and transfer, 3) growing amount of open source software and benchmark problems, and 4) increased interest and investments by industry.

This thesis focuses on analysing, comparing and implementing different ML algorithms to determine whether they are capable of tackling all 4 problems that are adjoined to optimisation studies.

## 1.2. Problem description

In the following section the central problem of the thesis is further scrutinised, however in order to provide a detailed problem description one important theoretical concept needs to be addressed first. This is the concept of exploring a design space. A design space is a multidimensional combination and interaction of design parameters. To be specific, for this project the design space amounts to all the CFD simulations with different design parameters within an optimisation study. The term design parameters represents the variables of a CFD simulation and can be divided in two distinct groups: 1) geometrical parameters (e.g. length, diameter) and, 2) flow parameters (e.g. mass flow, heat input). A 3D visualisation of an arbitrary design space is depicted in Fig.1.1.

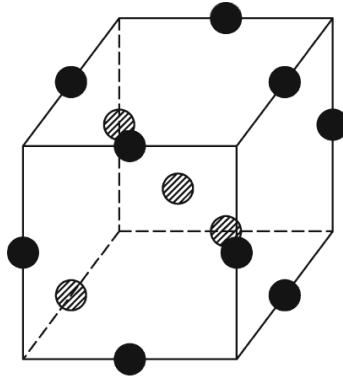


Figure 1.1: Arbitrary 3D design space. Each black and shaded dots represents a design point.

Each axis of the 3D rectangle represents a design parameter that accounts for a set of values. As mentioned above, design spaces are not limited to three dimensions (as depicted in Fig1.1) but theoretically could have infinite dimensions for infinite design parameters. Each dot is referred to as a design point and can represent 1 CFD simulation when chosen by a statistical algorithm. The details of these statistical algorithms are beyond the scope of this project, but briefly described in Appendix B.1. These statistical algorithms are designed to minimise the number of design points and simultaneously provide the best representation of the design space. For optimisation studies minimising the design points is important due to the computational cost inherent to each CFD simulation. Simultaneously, the design space cannot be too sparsely sampled with design points because it renders it impossible to guarantee that the "optimal" design is found. Lastly, the term exploration of the design space represents the chosen path between all the design points. This thesis puts a strong emphasis on the product of such an exploration. The product is defined for this thesis as: for every explored design point, its full flow-field is computed and available to the user. Therefore after this point, exploration of a design space is synonymous to having the full flow-field computed and available at every design point in the design space.

Now that the concept of design space and how to explore it is known, the detailed problem description can be formulated. The central problem where this thesis revolves around can be stated as follows:

**Problem description:** Exploration of a design space is very expensive as a whole due to the subject, number and computational cost of each design point (high-fidelity CFD simulation). Moreover, due to intractable amount of computed data it can practically not be stored and our physical insight, the ability to access information and make adjustments is lost.

With the central problem formulated above, it is possible to address the available options that researchers currently have to decrease the overall computational cost and time of optimisation studies. After each option, reasons are listed why they do not suffice to solve the central problem without severe compromises.

1. Decrease the accuracy of each individual CFD simulation. Where accuracy refers to switching between CFD solvers and/or decreasing the mesh quality. Accuracy is always a trade-off with computational cost and time. Therefore, (further) decreasing the accuracy of the CFD simulation also decreases the cost and the quality of the result.

- The optimal point, where accuracy and cost are balanced, depends on the goal and subject of the simulations. Many complex physical processes (including typical application cases of Siemens) require a high level of accuracy and therefore do not allow for a decrease in accuracy.
2. Decrease the complexity of the geometry and/or physics of each simulation subject.
    - It is common practice when setting-up CFD simulations, to keep the complexity as computationally lean as possible. This implies that when (further) decreasing the complexity the quality of your results also suffers significantly. Moreover, significantly simplifying a geometry and/or the physics is in general not possible with respect to the goal of the simulation.
  3. Reduce the overall amount of CFD simulations when performing optimisation studies.
    - As stated in Example 1, the amount of CFD simulations is strongly dependent on the design parameters and the amount of values each parameter accounts for. Therefore, the amount of total CFD simulations can be decreased by reducing the number of design parameters and the number of values each parameter can consist of. The consequence of this all is that the search for the optimal design is narrowed and can lead to an "optimal" design that is actually locally optimal but not globally optimal. Therefore, reduction of the amount of CFD simulations degrades the quality of the optimal design and cannot be therefore considered an optimal design.
  4. Implement a different optimisation algorithm.
    - Instead of using an optimisation based on a DoE, use a different optimisation algorithm (e.g. SHERPA, GA or Gradient-Based). These can reduce the computational cost but do not overcome the loss of physical insight and ability to access information or make adjustments.

The general drawback of the first 3 basic options listed above, is that every possible computational cost reduction results in a quality reduction of the CFD simulations and optimisation study. This is not surprising since quality, time and cost are correlated. Whereas the fourth option can reduce computational cost, however it still does not mitigate the other problems. These ineffective options lead to the formulation of the goal of this thesis:

**Goal:** This thesis researches the possibilities of combining CFD and Machine Learning to make optimisation studies

- computationally cheaper,
- less time-consuming,
- and more physically interpretable (by having the flow-field at each design point)
- and retain the ability to access information and make adjustments

### 1.3. Research questions

In this section a work plan in the form of research questions is formulated to reach the (aforementioned) goal of this thesis.

- Can ML-algorithms be combined with CFD simulations in such a way that they reduce the overall computational cost and time of the exploration of a design space? The product of interest are the full flow-fields for every explored design point.
  - What ML-algorithms are suited for this goal?
  - When making assumptions, what are the physical implications?
  - What speed-up times can theoretically be achieved?
  - What is the magnitude of the accuracy losses adjoint to this approach?
- What are the limitations of CFD-ML approach and can they be overcome?

Through answering these questions, this thesis aims to give a substantiated conclusion on the merit of combining CFD & Machine Learning.

## 1.4. Report outline

The outline of this thesis report is as follows. Chapter 2 provides the theoretical background of Fluid dynamics and CFD. Basically, Chapter 2 provides the information on why CFD simulations are so computationally expensive. Chapter 3 provides the literature study on Machine Learning. In this chapter potential algorithms are identified, judged and selected for the combination with CFD. In Example 1 of Chapter 1 a typical Siemens application case is described. This thesis did not use a typical Siemens case because of its high complexity that opposes the goal of this thesis. Chapter 4 compactly describes the two applications cases used throughout this project. Chapter 5 describes the first implemented ML-algorithm, which is the POD-Interpolation algorithm. This is succeeded by the second algorithm, which is the sDMD-Scaling algorithm. Lastly, Chapter 7 provides the conclusions and the recommendations.





# 2

## Theoretical background

A key problem of optimisation studies is that they require enormous amount of resources. Not uncommon is that these studies involve the computation of +150 computationally expensive CFD simulations. Chapter 2 intends to answer:

1. What makes fluid dynamics complex?
2. Why are high-fidelity CFD simulations required?
3. What makes high-fidelity CFD simulations computationally expensive?

By answering these questions, a deeper understanding of the problem is achieved and partially explains why Siemens is looking for alternative ways to predict flow-fields. Section 2.1 describes the governing equations of fluid dynamics and its complexities. Whereas, Section 2.2 describes how these equations are solved with the help of CFD. Section 2.3 provide the concluding remarks and answers the 3 aforementioned questions.

### 2.1. Fundamentals of fluid dynamics

The field of fluid dynamics aims at describing the motion of fluids and gases with the help of basic laws of mechanics and thermodynamics. Fluid dynamics provides a method for studying any "flow" in or around an subject. Some canonical examples of fluid dynamics include a flow in or around a pipe, lift and drag of a wing profile, turbulent shear mixing layers. The movement of a fluid is generally referred to as "flow" and describes how fluids behave and interact with their surrounding environment.

To reduce the complexity of the physics the following 4 reasonable assumptions are made. These hold true for the entirety of this document. For more information on these assumptions see the work of Kundu et al. [75].

- **Continuum Hypothesis.** This is an idealisation of a fluid or gas, and are treated as a continuum.
- **Newtonian fluids.** The viscosity is constant and the viscous stresses are linearly proportional to the local strain rate.
- **Incompressible flows.** A fluid or gas element is of constant density ( $\rho = const.$ ) and does not change with pressure and temperature.
- **Turbulent flows.** Only turbulent flows are researched, laminar flows are out of the scope.

Succeeding this brief introduction on fluid dynamics are 2 subsections. The first is on the conservation equations of fluid dynamics. These equations describe the physics and enable the computation of any flow-field. The second subsection is on turbulence, which is a key property of any turbulent flow. Turbulence is a reoccurring theme in all the remaining chapters.

#### 2.1.1. Conservation equations

The governing principles in fluid mechanics are based on conservation laws for mass, momentum and energy [75]. The equations of motion of any flow-field are based on these three laws.

The conservation laws are combined in the Navier-stokes equations that is a system of coupled nonlinear partial differential equations (PDE) [75]. The conservation of energy equation is neglected because for non-reacting flows, with marginal temperature differences, only the mass and momentum equation are of concern.

### Conservation of Mass

The conservation of mass equation is also referred to as the continuity equation and states that within a well defined control volume the total mass is constant. The equation can be written in several forms, but for an incompressible flow the continuity equation can be reduced to [75]

$$\nabla \cdot \vec{u} = 0. \quad (2.1)$$

### Conservation of Momentum

The conservation law of momentum is determined by directly applying Newton's law of motion ( $m \cdot a = F$ ) to an fluid element. This equation relates the acceleration of a fluid element to its inertial, surface and pressure forces and is written as [75]

$$\rho \frac{D\vec{u}}{Dt} = \rho \vec{g} + \nabla \cdot \mathbf{T}. \quad (2.2)$$

Equation 2.2 is referred to as Cauchy's momentum equation and is valid for any continuum provided its deformation is described by an Eulerian approach. The two preceding equations are not sufficient for a complete description of a flow. Because the preceding equations contain  $\rho, \vec{u}$  and  $\mathbf{T}$  that hold  $1 + 3 + 9 = 13$  unknowns. The constitutive equation that describe the relation between the stress and deformation helps in this regard. The incompressible stress tensor  $\mathbf{T}$  is defined as [75]

$$\mathbf{T} = -p\delta + 2\mu\mathbf{S} \quad (2.3)$$

where  $\mathbf{S}$  is defined as the volumetric strain rate tensor. The diagonal terms of  $\mathbf{S}$  represents elongation and contraction per unit length. Whereas the off-diagonal terms represents shear deformations. Equation 2.3 is a linear constitutive equation and is relatively accurate for common fluids such as air and water [75]. The linear constitutive relation only holds for small strain rates, or otherwise said Newtonian fluids. Non-Newtonian fluids are characterised by moderate or high rates of strain. The aforementioned equation enables for the formulation of the Navier-Stokes (NS) equations.

### The Navier-Stokes equation

The NS equations combines all previous equations into one set of coupled nonlinear partial different equations (PDE). The set of NS equations consist of the continuity equation (Eq.2.1) and the Navier-Stokes momentum equations. For an incompressible Newtonian fluid and small temperature differences the NS equations are written as [75]

$$\rho \frac{D\vec{u}}{Dt} = -\nabla p + \rho \vec{g} + \mu \nabla^2 \vec{u}. \quad (2.4)$$

When reduced to its most compact form the NS equations are quite elegant but deceptively complex. There exist only a handful of cases [120] for which Eq.2.4 can be solved analytically. These cases are however drastically simplified and have well-defined boundary and initial conditions. This is rarely ever given in nature and complex problems. The NS equations are considered so complex that they are included in the seven Millennium Prize Problems [31], that are conceived to promote solving the (current) seven most difficult problems worldwide.

The primary cause for its complexity is due to the very strong nonlinear term, the advective acceleration term  $(\vec{u} \cdot \nabla) \vec{u}$  that bolsters a quadratic non-linearity [75]. A nonlinear term does not behave in a proportional manner to a change of the given input. Nonlinear terms are the cause of any system to behave in an unpredictable and counter intuitive way. The secondary cause for the complexity is due to the strong coupling between the nonlinear equations [75]. The coupling do not allow the equations to be solved independently, because they require information from each other. These are the two predominant factors that make the NS equations so formidable, and generally can only be solved by using numerical methods. These numerical methods are extensively discussed in Section 2.2.

### 2.1.2. Turbulence

The governing NS equations, seen in the previous section, allow for the computation of the dominant flow quantities such as velocity, pressure and temperature. Turbulence however is the product of the pressure and velocity fluctuations and is a key property of any turbulent flow. Turbulence can be commonly observed and experienced in aeroplanes, near clouds, water jets and more examples are easily found. In all these physical processes, turbulence fulfills an essential role and is important for heat and mass transfer, mixing and reacting flows [103].

Aside from the crucial role turbulence fulfills in fluid dynamics, it is also an important topic in Chapter 5 and Chapter 6. These chapters provide and analyses the predicted flow-fields, which are computed using Machine Learning algorithms. To give a hint, sometimes the predicted flow-fields are inaccurate because essential turbulence phenomena (described in this section) are neglected in the computation.

#### The Reynolds number

Turbulence is categorised in two groups, laminar and turbulent. A laminar flow is characterised as layered, smooth and ordered. Whereas a turbulent flow is chaotic, fluctuating and disordered. An important parameter in turbulence is the dimensionless Reynolds number, that allows for a relatively clean distinction between a laminar and turbulent regime. The Reynolds number describes the ratio of the inertial and viscous forces and is defined as [75]

$$Re = \frac{\text{Inertial forces}}{\text{Viscous forces}} = \frac{\rho U \mathcal{L}}{\mu}. \quad (2.5)$$

In Fig.2.1 each turbulence regime can clearly be identified. From the left, where the flow starts in the laminar flow regime (where the viscous forces are dominant), to the right where it transitions to a turbulent regime (where inertial forces are dominant).

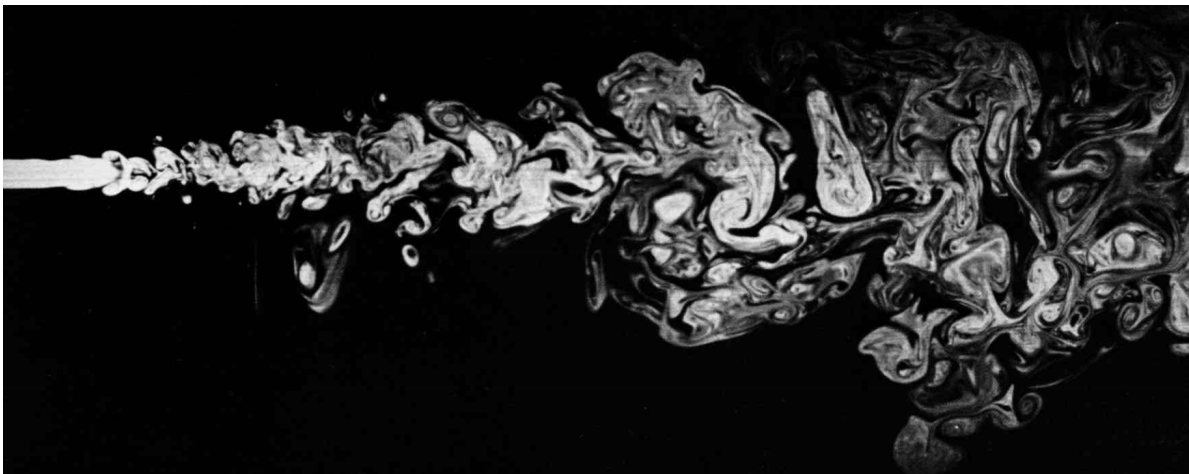


Figure 2.1: A water jet which transitions from laminar to turbulent. Obtained from the work of Van Dyke [151]

#### The Energy Cascade

Another important concept in turbulence that needs to be explored is the "Energy Cascade" which again is done with the help of Fig. 2.1. From a certain point the flow is turbulent and appears to be erratic, but if properly examined not completely random [103]. Because in the stream repeating vortex-structures can be identified, these vortices are referred to as Eddies. Eddies vary in size and the Energy Cascade describes the interaction between these Eddies. The Energy Cascade dictates that larger Eddies interact with other larger Eddies and break up into smaller Eddies. This process continues until the Eddies are so small that they are dissipated into heat through viscous friction. Turbulent structures/vortices/Eddies is again grouped in two groups: the macro- and microstructure.

The larger Eddies are grouped in the macrostructure group and are associated with a length scale  $\mathcal{L}$  and a velocity scale  $V$ . These two scales are used to express an important turbulence parameter, the kinetic energy dissipation rate [103]:

$$\epsilon \propto \frac{V^3}{\mathcal{L}}. \quad (2.6)$$

This is also referred to as the Kolmogorov relation and considered as one of the most important results of turbulence theory [103]. The dissipation of turbulent kinetic energy describes the rate at which the energy is absorbed by breaking down the Eddies into ever-smaller Eddies until it is converted into heat via viscous forces. Regarding the microstructure, it is dominated by nonlinear diffusion processes. The microstructure is not computed by the employed CFD solvers, and therefore no detailed information is given on this topic. For more information on the microstructure see the work of Nieuwstadt et al. [103]. The most important findings about the energy cascade are summarised in Table 2.1.

Table 2.1: Properties of the macro- and microstructure, obtained from the work of Nieuwstadt et al. [103]

Macrostructure	Energy cascade $\rightarrow \epsilon$	Microstructure
<ul style="list-style-type: none"> <li>• Produced by the average flow (active)</li> <li>• Frictionless</li> <li>• Effective transport of: mass, momentum and energy</li> <li>• Anisotropic</li> <li>• Scales: <math>V, L</math></li> </ul>		<ul style="list-style-type: none"> <li>• Fed by the cascade process (passive)</li> <li>• Dominated by friction</li> <li>• Molecular transport processes</li> <li>• Isotropic</li> <li>• Scales: <math>v, \eta</math></li> </ul>

Small note, the characteristics scales for the microstructure  $v, \eta$  are referred to as the Kolmogorov scales [103] for the length and velocity.

### The dynamics of vortices

The dynamics of vortices are referred to as "vorticity" and can be interpreted as the quantity of the rotation of a fluid element. Vorticity and turbulence are intertwined and turbulence can be seen as a form of "chaotic vorticity" [75, 103]. Therefore, a section on turbulence is not complete without a subsection on vorticity. Moreover, in Section 4.3 the physical implications of using a quasi 2-D CFD domain on vorticity are described. But first, vorticity is defined as the curl of the velocity field [103]

$$\vec{\omega} = \nabla \times \vec{u} \quad (2.7)$$

and the governing equation of vorticity is defined as

$$\frac{D\vec{\omega}}{Dt} = (\vec{\omega} \cdot \nabla) \vec{u} + \nu \nabla^2 \vec{\omega}. \quad (2.8)$$

Equation 2.8 is referred to as the vorticity equation and is derived from the NS equations (Eq.2.4) by applying curl to both sides of the equation. The pressure term in the equation disappears since the curl of a gradient is zero. The third term in Eq.2.8 represents the diffusion of vortices, whereas the second term describes the interaction between fluid deformation and vorticity. This second term requires additional attention because it plays a fundamental role in vorticity and therefore turbulence. The second term is rewritten<sup>1</sup> as [103]

$$\omega_j \frac{\partial u_i}{\partial x_j} = \omega_j S_{ij}, \quad (2.9)$$

where

$$S_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$$

is the volumetric strain rate tensor, previously seen in Eq.2.3. There exists two types of interactions, for an incompressible and inviscid flow, that are coupled to the diagonal and off-diagonal terms of  $S_{ij}$ .

<sup>1</sup>Using Einstein notation for referring to diagonal and off-diagonal terms.

- For diagonal terms  $i = j$ , the rate-of-strain tensor describes changes in the length of a material line segment. Where  $s_{ij} > 0$  results in a stretched line segment and  $s_{ij} < 0$  in compressed line segment. Generally speaking, for an inviscid fluid it is determined that vorticity increases when  $s_{ij} > 0$  and decreases for  $s_{ij} < 0$ . A good metaphor for vortex stretching/compression would be that of a spinning ice skater which speeds-up by reducing the rotational inertia by retracting the arm and legs. By stretching a vortex roll the rotational inertia becomes smaller which results in an increased vorticity.
- For off-diagonal terms  $i \neq j$ , the rate-of-strain tensor describes a tilting/angular displacement, from the  $j$ -direction to the  $i$ -direction, of the vorticity vector  $\omega_j(0, 0, \omega)$ .

Important to note is that an energy transfer occurs between the vortex and the deformation field  $s_{ij}$  that originates from the energy cascade. Where the larger Eddies deform and transfer energy to the smaller Eddies and therefore increase the vorticity of the smaller Eddies. Vortex stretching is an essential part of turbulence since it plays a key role in transfer of energy between Eddies. Vortex stretching however is only possible in three-dimensional flows, since in two-dimensions the vortex lines cannot be deformed.

Lastly, Section 2.1.2 is a brief overview of turbulence, only the relevant (w.r.t this thesis) turbulence aspects are described. For more information on turbulence see the work of Nieuwstadt et al.[103].

## 2.2. Fundamentals of Computational Fluid Dynamics

Computational fluid dynamics is a branch of fluid mechanics that is used to solve the NS equations. In the past decade, CFD usage and research field has grown tremendously alongside the growth of computing power. CFD offers the possibility to solve problems with complex geometries and flow physics. CFD has multiple approaches to solve the NS and to deal with turbulence. In total there are four different approaches available to CFD users: Reynolds-Averaged Navier-Stokes (RANS), Unsteady-Reynolds-Averaged Navier-Stokes (URANS), Large-Eddy Simulations (LES) and Direct Navier Stokes (DNS). Hybrid solvers are emerging but not frequently used and therefore neglected in this list.

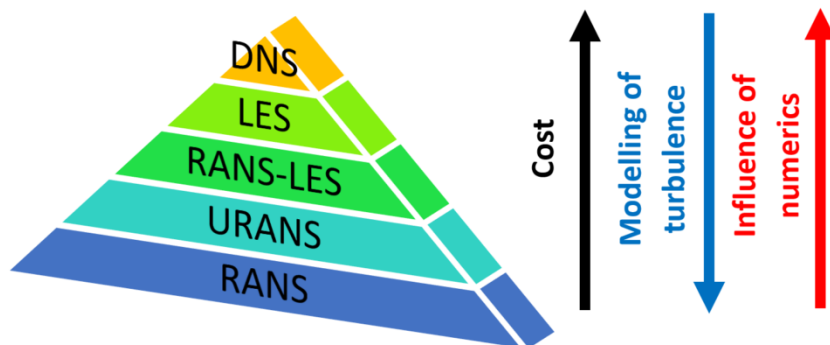


Figure 2.2: Computational cost and accuracy are ranked for different CFD solvers. Reworked from the work of Sagaut et al. [125]

Figure 2.2 depicts the accuracy scale and computational cost of the CFD solvers. In this thesis essentially only a LES solver is used, RANS is merely used as an initialisation tool for the succeeding LES solver. It is common practice at Siemens to initialise the LES solver with the computed mean flow-fields from RANS. Numerical stability and convergence properties significantly improve from this practice [139]. LES solvers are typically used because of the desired level of accuracy. Optimisation studies at Siemens are only performed for critical components and these are governed by very complex (transient) physics. Furthermore, the geometries of these critical components are extremely complicated. In-house studies [117] showed that RANS based optimisation studies did not meet the desired degree accuracy. For similar reasons are URANS solvers typically not used for optimisation studies.

Since all simulations in this work are performed with an LES solver, it is described in Section 2.2.1. For information on RANS (see Section A.1), URANS and DNS solvers see the work of Ferziger [45], Pope [110] or Wilcox [158].

### 2.2.1. Large Eddy Simulation

Unlike RANS, that averages over all the scales of motion present in a turbulent flow, a LES resolves the large scales of motion in a turbulent flow. RANS turbulence models apply relatively well to small scales of the flow, however less to the larger scales. Moreover, these larger scales contain more energy than the small scales and their size and strength make them more effective transporters of fluid properties. The LES makes a distinction between the larger scales that are resolved, and the smaller scales that are modelled with the help of turbulence models. To separate the larger and smaller scales a spatial low pass filter is used, which is defined as [110]

$$\tilde{c}(x_1, x_2, x_3) = \int \int \int G(\vec{x}' - \vec{x}) c(\vec{x}) dx'_1 dx'_2 dx'_3. \quad (2.10)$$

An arbitrary quantity ( $c$ ) is filtered with the function ( $G$ ) that has a length scale ( $\Delta$ ) associated to it. Eddies of a size larger than the length scale ( $\Delta$ ) are resolved and the smaller Eddies are modelled. The length scale ( $\Delta$ ) is generally attributed to the grid density. Therefore, the larger scales are typically referred to as the grid or resolved scale whereas the smaller scales are known as the subgrid scale (SGS). Different filter functions ( $G$ ) do exist and are used in LES research, however in essence they have the same purpose in making a distinction between the grid and subgrid scales [158]. The filtered Navier-Stokes are written as [103]

$$\frac{\partial \tilde{u}_j}{\partial t} + \frac{\partial \tilde{u}_i \tilde{u}_j}{\partial x_i} = -\frac{1}{\rho_0} \frac{\partial \tilde{p}}{\partial x_j} - \frac{\partial}{\partial x_j} \nu \left( \frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right) + \frac{1}{\rho_0} \frac{\partial \mathbf{T}_{SGS}}{\partial x_j}. \quad (2.11)$$

for LES computations. Equation 2.11 uses the symbol  $\mathbf{T}_{SGS}$  that is referred to as the subgrid stress tensor and defined as

$$\mathbf{T}_{SGS} = \rho_0 (\overline{u_i u_j} - \tilde{u}_i \tilde{u}_j). \quad (2.12)$$

The subgrid stress tensor represents how the smaller (modelled) Eddies exerts a stress on the larger resolved Eddies. Figure 2.3 provides an illustration how a LES typically resolves a turbulent jet, the large scale turbulent motions are captured by the LES.

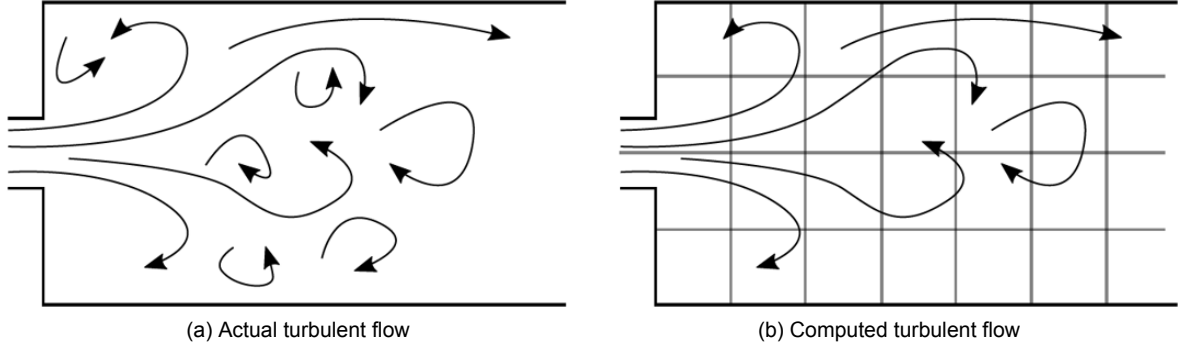


Figure 2.3: Comparison between the actual turbulent flow and the approximation computed by a LES CFD solver. Obtained from the work of Durst [41].

LES faces problems with unknown terms in the subgrid stress tensor. Turbulence models are required to find equations for the unknown components based on the known flow properties. The oldest and best known model is the Smagorinsky model that postulated the stress term as follows [136]

$$\mathbf{T}_{SGS} = \nu_t \left( \frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right) \quad (2.13)$$

where  $\nu_t$  describes the diffusive behaviour of the smaller scales and goes by the name of the subgrid Eddy Viscosity. Its definition is found in the work of Nieuwstadt et al. [103]. The Smagorinsky model was the first turbulence model for LES and multiple derivations are based on this model. Most notably, the Wall Adapting Local Eddy-Viscosity (WALE) model that is typically used in the CFD software [139] (StarCCM+) used by Siemens. For this reasons, the WALE subgrid model is used throughout the project.

The WALE model is designed to obtain better results in the vicinity of walls [48] (compared to the Smagorinsky model) by taking the strain and rotation rate of the smallest resolved turbulent fluctuations [15] into consideration. The 2 main benefits compared to Smagorinsky's model is that the viscosity goes to zero near the wall and that it does not require information of the position and direction of the wall [48].

Lastly, an important parameter in CFD and LES simulations is the ratio of resolved turbulent kinetic energy and the total turbulent kinetic energy in the flow. It can be seen as a measure of the turbulence resolution and the ratio is defined by Pope as [111]

$$R(\mathbf{x}, t) \equiv \frac{k_r(\vec{x}, t)}{K(\vec{x}, t) + k_r(\vec{x}, t)}. \quad (2.14)$$

A value of  $R = 0$  corresponds to a DNS (the highest possible accuracy), and  $R = 1$  to RANS (the lowest possible accuracy). The ratio however requires the evaluation of the turbulent kinetic energy of the resolved motions ( $K(\vec{x}, t) \equiv \frac{1}{2} \langle (\tilde{u}_i - \langle \tilde{u}_i \rangle_T) (\tilde{u}_i - \langle \tilde{u}_i \rangle_T) \rangle_T$ ) and the modelled turbulent kinetic energy ( $k_r$ ). Where T is defined as the temporal average of that property. Pope [111] suggest not to exceed a ratio for 0.2 for a LES to be considered well-resolved.

### 2.2.2. Discretization

An important step in solving the governing equations is using discretization methods to approximate the set of PDE's. The discretized equations can subsequently be solved on a computer. The approximations are applied to the spatial and temporal domain to obtain numerical results at discrete points. The accuracy of the CFD solvers are to a (large) extent dependent on the quality of the used discretization schemes. There are three common methods for discretization, Finite Volumes (FV), Finite Differences and Finite Elements. In StarCCM+, the used CFD software, FV is used and therefore only discussed below. As the name suggest, the FV method divides the domain into a finite number of small non-overlapping control volumes, called mesh-elements.

#### Spatial and temporal discretization schemes

The discretization schemes, used for all CFD computations, are based on the results of an in-house study performed by Indelkofer [65]. The study assessed the numerical accuracy of all the spatial and temporal discretization schemes available in StarCCM+. This is investigated by determining the numerical dissipation and dispersion of an initialised vortex in a two-dimensional rectangular domain. A flow without viscosity and thermal conductivity is used, therefore the only way for the vortex to dissipate is through numerical dissipation. In total eight different combinations of numerical settings were tested and analysed. Based on the results and the recommendation made by Indlekofer (on the trade-off between numerical accuracy and computational cost), the following settings are used:

Table 2.2: Discretization schemes used throughout this work

<b>LES</b>	
Algorithm	Implicit-Unsteady
Coupled/segregated	Segregated
Spatial discretization	Second-Order Upwind
Temporal discretization	Second-Order Backward Difference

The LES uses a segregated flow solver that solves the integral conservation of mass and momentum in a sequential manner. It employs a pressure-velocity coupling algorithm where the mass conservation constraint on a velocity field is achieved by solving a pressure-correction equation. The pressure-correction equation is derived from the continuity and momentum equation such that the predicted velocity field fulfills the continuity equation, by correcting the pressure accordingly. Segregated solvers are suitable and computationally inexpensive for incompressible flows [139].

To describe the flux and the interaction between control volumes a (spatial) Second-Order Upwind scheme is used. This is sufficiently accurate and computationally efficient [139] for the relative straight-forward mesh employed during the CFD problems. The Second-Order Backward Difference scheme is relatively efficient (compared to higher order temporal schemes) and is based on two previous time-steps. The scheme is formulated as [139]

$$\frac{d}{dt}c = \left(\frac{3}{2}c_{n+1} - 2c + \frac{1}{2}c_{n-1}\right)\frac{1}{\Delta t}. \quad (2.15)$$

To determine the correct time-interval ( $\Delta t$ ) the Courant-Friedrichs-Lewy (CFL) number [82] is used. It expresses that the distance of a fluid-element travelled during a time-step must be smaller than the distance between mesh elements. Information from an arbitrary mesh-element must only propagate to its adjacent mesh-element and should not be allowed to skip mesh-elements for numerical stability. The CFL number is defined as [82]

$$\text{CFL} = \frac{U\Delta t}{\Delta x} < 0.7 \quad (2.16)$$

Common practice at Siemens, is not to exceed a CFL number of 0.7 [45, 139]. With Eq.2.16 an appropriate time-interval  $\Delta t$  is determined, that is subsequently used in Eq.2.15.

### 2.3. Conclusions theoretical background

Chapter 2 started with three posted questions. The first question is formulated as: what makes fluid dynamics so complex? It is found that the complexities stem from two factors, the nonlinearity and the coupling between the NS equations. The advective acceleration term in Eq.2.4 bolsters a quadratic nonlinearity and is largely the cause for the system to behave unpredictably. Furthermore, the coupling between the equations dictates that these need to be solved simultaneously and cannot be adressed separately.

To solve the NS equations analytically is only possible for a very limited number of cases. Practically, only a numerical tool like CFD is capable of solving these equations. To answer the second question, only LES simulations (generally a high-fidelity CFD solver) are used in this work, because in-house studies showed that RANS and URANS did not meet the desired degree of accuracy.

The third question is: what makes high-fidelity CFD so computationally expensive? A LES uses a spatial filter to make a distinction between larger and smaller scales. Eddies of a size larger than the length scale ( $\Delta$ ) are resolved and those that are smaller are modelled. The length scale ( $\Delta$ ) is typically attributed to the grid density. An LES is significantly more computationally expensive than a RANS or URANS because it resolves more length scales. The ratio of resolved versus modelled length scales is notably higher for an LES than other solvers (excluding DNS). Resolving more length scales significantly increases the computational costs.



# 3

## Literature review

Where Chapter 2 largely revolved around the problem, Chapter 3 explores the research field of Machine Learning (ML) for possible solutions. Chapter 3 intends to answer or address the following points:

1. What is ML and what does it encapsulates?
2. Which unique challenges does fluid dynamics pose for ML?
3. Identify active research fields, within the fluid dynamics domain, that implement ML.
4. Identify, judge and select potential ML algorithms for the prediction of flow-fields.
5. Provide the mathematical working principles of the selected ML algorithm.

As these points indicate, the knowledge on ML is gradually build-up. Equivalent to the order of Chapter 3. First A general definition of ML is given in Section 3.1. This is succeeded by Section 3.2, that describes the current state of ML for fluid dynamics purposes. Section 3.3 converges to the main goal of the literature study by identifying, judging and selecting ML algorithms. This is followed by an in-depth description of the selected ML algorithms in Section 3.4. Lastly a discussion is written on the findings of the literature study in Section 3.6.

### 3.1. What is Machine Learning?

ML has grown dramatically in the recent decades and its use in current-day applications are widespread [92]. A strong indication for the growth of ML is the number of patents that have increased tenfold between 2000 and 2018 [42]. Figure 3.1 (a) shows the exponential growth of patents for three groups, that in this document are all categorised under the ML umbrella term. A universally accepted definition of ML is lacking and different documents tend to make a distinction between ML and (Deep) Neural Networks.

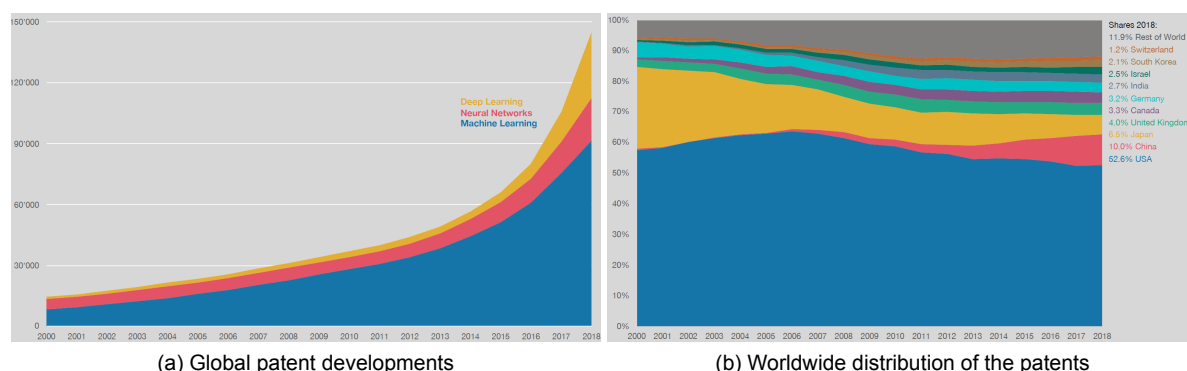


Figure 3.1: Overall development of ML patents, obtained from the work of EconSight [42].

Figure 3.1 (b) shows the worldwide distribution for all the world-class patents. A view on individual companies, ranked according to their world class patents, show a similar global distribution. 6 Out of the top 10 companies are from the US, 2 from China and 2 from Japan. Phillips is the first European company (ranked 14<sup>th</sup>) and followed by Siemens that is ranked 16<sup>th</sup>. Interestingly, in the list formulated by EconSight [42] there is not one single company that is specialised in ML. All companies are active in other areas and try to solidify their position by implementing ML algorithms [42].

As stated before, a universally accepted definition of ML is absent and to exactly define what ML encompasses is not trivial [30, 42]. Every month new algorithms or variations on existing algorithms are formulated and join the field of ML. Moreover, it is not uncommon that:

- an algorithm is referred to by multiple names depending on the field of use,
- and multiple derivations exist from the original algorithm, which again goes by a different name or names.

To name one example, a dimension reduction algorithm that goes by the name of Proper-Orthogonal Decomposition (POD). But also Principal-Component Analysis (PCA), Quasi-Harmonic Modes, Empirical-Component Analysis and Hotelling Analysis depending on the field of use [145]. The amount of variations of this one algorithm is large and easily exceeds eight variations [87, 127, 134, 145, 153]. It should come as no surprise, as why a universal definition of ML and generally an overview of ML algorithms is still lacking [30, 42]. Nevertheless, the following definition of ML is considered to be the most complete by the author.

Machine Learning is using data to create a model and then using that model to make predictions.

With this definition of ML, it is possible to address the large variety of ML algorithms. ML algorithms are categorised in three groups: 1) supervised, 2) semi-supervised and 3) un-supervised. The supervised group creates (trains) a model by being fed specific labelled examples. Whereas, un-supervised algorithms create a model by discovering structures within given data. The semi-supervised group is a hybrid and uses predominantly unlabelled training data augmented with labelled training data. The three ML groups can again be subdivided in seven groups, according to their intended use-cases. Figure 3.2 provides an overview of all the different ML categories and list a few common algorithms.

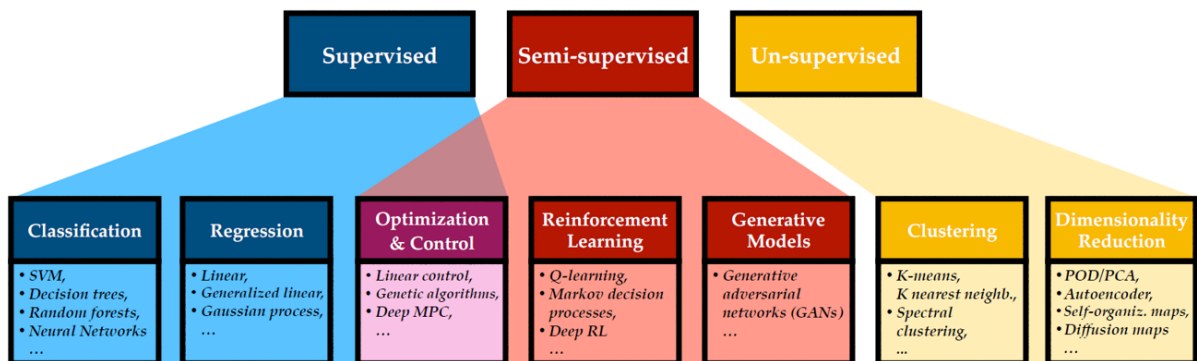


Figure 3.2: Overview of various ML groups and algorithms. Obtained from the work of Brunton [24]

There exists no official number or list of ML algorithms but to serve as an indication, a document [61] from the software *R* provides more than 100 unique ML algorithms (thus excluding the many variations on each algorithm). The seven ML categories from Fig.3.2 account for all these algorithms.

### 3.2. Machine Learning for fluid dynamics

In this section ML algorithms are discussed that have been implemented for fluid dynamics purposes. The field of fluid dynamics has traditionally dealt with large quantities of data. Historically more from experiments and measurements but present-day more from numerical CFD simulations [109]. The analysis of data relies still mostly on expertise, statistical analysis and heuristic approaches [24].

These are thus areas for further improvements, by possibly augmenting or to an extent replace, these approaches with ML algorithms. Ideally, ML offers a modular and agile framework that can be customised to different problems. Success-stories of ML in different fields are plentiful [92] and ML is progressively making inroads in the fluid dynamics field [24]. However, fluid dynamics poses new and more complex challenges. The cause for these challenges is already to some extent described by Section 2.1, with the introduction of the NS equations and its complexities. For the sake of clarity, a list is formulated with the challenges unique to fluid dynamics.

- Fluid dynamics are governed by non-linear multi-scale phenomena that are still, to a degree, not fully understood [24, 75, 103].
- In fluid dynamics, more than other fields, a relative high accuracy of the approximation (e.g. flow-field properties and/or underlying mechanisms) is needed to be able to analyse them [24].
- ML solutions must produce interpretable and explainable results in order to learn about the original fluid dynamics problem. Only by having understandable results, that can be validated, scientific knowledge increases. Fluid dynamics is governed by the need to learn about the physics in contrast to purely digital applications.
- Fluid Dynamics data is characterised by being:
  - Dense in the spatial domain and simultaneously sparse in the temporal domain. For example, for an LES the spatial density is directly linked to the turbulence resolution (see Section 2.2.1). Thus it has relatively many grid points. Whereas the computational cost of solving a LES results in sparse data in the temporal domain.
  - Often heterogeneous (data with high variability of data types and formats) and requires special attention for processing [24].
  - Lacking in terms of generalizability and transferability. The data is almost always very case specific and therefore dependent on design parameters, spatial and temporal resolutions. Other research fields have very large database that can easily be found and accessed online [157]. These are databases on a wide-variety of subjects and purely intended for ML research. For fluid dynamics there is a lack of very large and easily accessible databases [24, 109] and these are essential for the most powerful of ML algorithms (DNN, CNN and Reinforcement learning) to learn about the complicated phenomena for different design parameters.

Following this list of challenges, two research directions are identified and discussed where currently the most attention is directed towards. The cited articles in these sections prove that the aforementioned challenges are not unassailable, and that significant progress is made regarding the implementation of ML for fluid dynamics. The two research directions are briefly discussed to highlight the current state.

### 3.2.1. Flow control

Flow control modulates the behaviour of a flow-field with the help of actuators that are informed by sensors. Flow control is used to stabilise unstable systems, dampen sensor noise and external influences in a flow [24]. ML implemented for general control purposes has been done for quite some time [20]. Specifically flow control poses different challenges, such as: 1) high-dimensional states, 2) strong non-linearity's, 3) latent variables, and 4) time delays [24]. The vision for flow control is to real-time identify instabilities and simultaneously suppress them with an actuator. However, this requires very fast ML algorithms that can identify, learn and modulate information from typically a limited set of measurement points [24]. Therefore, real-time flow control is still a vision and not a reality [24].

The following examples showcase ML algorithms that fulfilled an important role for flow control. Steinfurth et al.[141] used a Spectral-POD method for flow identification that provided information to the actuator (i.e. pulsed blowing) to prevent turbulent flow separation on the trailing-edge of a wing-profile. Steinfurth et al. has shown that a recirculation zone can be intermittently suppressed or completely suppress flow separation depending on the actuation intensity. Lee et al. [80] proposed in 1997 a control network including a NN for turbulence control, that demonstrated a 20% drag reduction for a turbulent channel flow.

Erichson et al. [43] proposes a shallow-NN that allows full flow-field reconstruction from the data of a few measurement points. Identifying and predicting the flow-field from a few measurements for active flow control [43]. The shallow-NN is demonstrated on three examples and show superior performance to traditional methods. One example, a typical Von Karman Vortex Street, is shown in Fig. 3.3. For more information on ML for flow control see the work of Brunton et al. [21, 24].

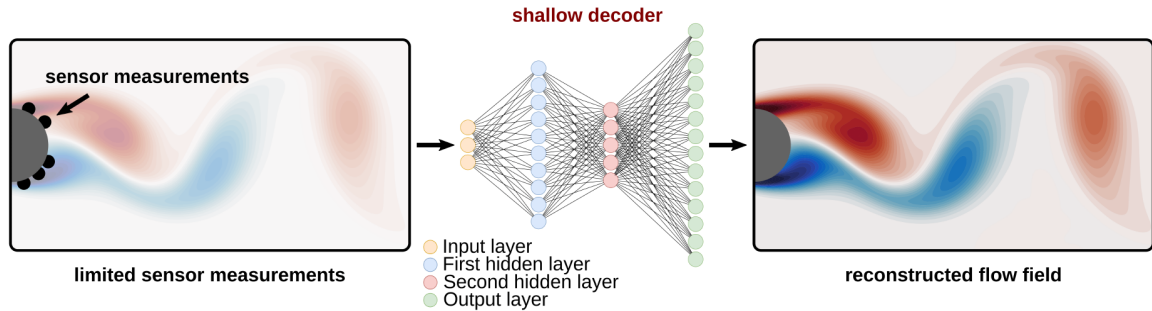


Figure 3.3: Illustration of the shallow-NN decoder, that uses a few sensor points to estimate the flow-field. Obtained from the work of Ericson et al. [43]

### 3.2.2. Identification of essential dynamical features

Identifying essential dynamical features from large data-sets is one of the core strengths of ML [24, 67]. The methods that identify essential dynamics are categorised in two groups based on their output.

1. Methods that identify and extract key features.
2. Methods that identify and represents the key features with a set of equations.

The difference between the two appears to be small, but the difference in their output is essential for their intended use cases. Group 1 represents two use cases (i.e. flow analysis and dimension reduction) in which learning about the original problem is key[144]. Whereas group 2 represents one use case, that is to express the key dynamics with a set of equations to make future-state predictions [22].

Group 1 are being dominated by modal decomposition methods (MDM), which is a common well-established group of algorithms to extract key dynamical features and/or obtain a dimension-reduction [24, 144, 145]. The MDM group is extensively discussed in Section 3.4 and therefore not further discussed in this section. Another ML technique that identifies and extract key features is the *k - means* algorithm. Kaiser et al. [70] proposed a strategy based on this *k - means* algorithm for identifying physical mechanisms in a unsupervised way. This strategy is subsequently applied to 3 cases: Lorentz attractor [20], a 2-D incompressible mixing layer and a 3-D incompressible turbulent wake. The strategy has shown to be able to capture key-features and their temporal evolution for all the three cases [70]. Another example is using a NN, proposed by Colvert et al. [33], that successfully identifies and classifies different vorticity wakes behind an airfoil.

Group 2 expresses the key features as equations (algebraic, ODEs, PDEs) that can make future-state predictions. Discovering unknown governing equations from data that are beyond the ability of humans to comprehend, is the vision for the following proposed methods. Brunton et al. [22] proposed the Sparse-Identification of Nonlinear-Dynamical-Systems (SINDY) method, that is a novel algorithm to discover the working principles underlying the data. SINDY has demonstrated to be powerful [34] in capturing key dynamical features in the data. For example, being able to capture and to predict a Hopf-bifurcation [22]. Brunton et al. suggest that SINDY is the first real significant step towards the long-term vision of sophisticated, unsupervised identification of underlying dynamical systems. Other promising work is performed by Raissi et al. [114, 115] that implements NN and DNN to discover physical laws expressed by PDEs from data. The methods are implemented on multiple canonical problems (i.e. Burgers, nonlinear Schrödinger and the Navier-Stokes equations) and shows potential to learn from (noisy) data. This topic of extraction of essential dynamical features is continued in Section 3.4, where MDM are extensively discussed.

### 3.3. Machine Learning for prediction of flow-fields

The upcoming section converges to ML algorithms that have potential to achieve the goal of this thesis. The goal is reiterated below.

This thesis researches the possibilities of combining CFD and Machine Learning to make optimisation studies:

- computationally cheaper,
- less time-consuming,
- and more physically interpretable (by having the flow-field at each design point)
- and retain the ability to access information and make quick adjustments

Algorithms with potential therefore need to facilitate all these points in the exploration of a design space. The outline of this important section is as follows, first 4 different approaches for predicting a flow-field are presented in Section 3.3.1 to Section 3.3.4. It is possible that an algorithm is listed in multiple approaches, due to differences in implementation and results of the algorithm. This all is succeeded by Section 3.3.5, that lists and selects the ML-algorithm that is going to be implemented.

#### 3.3.1. Reduced order models

Reduced order modelling (ROM) represents a large research field that covers multiple physics areas [126]. There are multiple accepted definitions of ROM and it predominantly depends on the goal of the user [126]. The general idea behind reduced order modelling is to approximate high-order problems by a low-order model (referred to as the Reduced Order Model) that yields less accurate results but can be solved for considerably less computational cost. The goal of a ROM is to provide to provide insights about the fundamental mechanisms and to be able to make future-state predictions [20, 124, 126]. The value of ROM is evident, there are significant scientific and engineering benefits in describing, solving and analysing complex problems in a low-order way.

In recent years a large number of studies are done on ROM, resulting in a large variety of ROM methods [126]. Some methods are designed for specific applications and others are more general. What follows are multiple examples of different ROM applications based on different methods (e.g. MDM, Galerkin, NN). Xia et al. [161] postulates a POD-Galerkin ROM to predict turbulent air flows for urban environments. Wolf et al. [85] proposes a ROM constructed by a DNN and applied to multiple canonical fluid-flow problems. One example of this is given in Fig. 3.4 that shows a 3-D flow-field prediction, for a time-step beyond the training window [85]. Furthermore, ROMs are postulated for reacting flow applications [62, 102], combustor instabilities [164], Electro-Thermal systems [126], low-dimensional modelling of turbulence [137] or flow control [105].

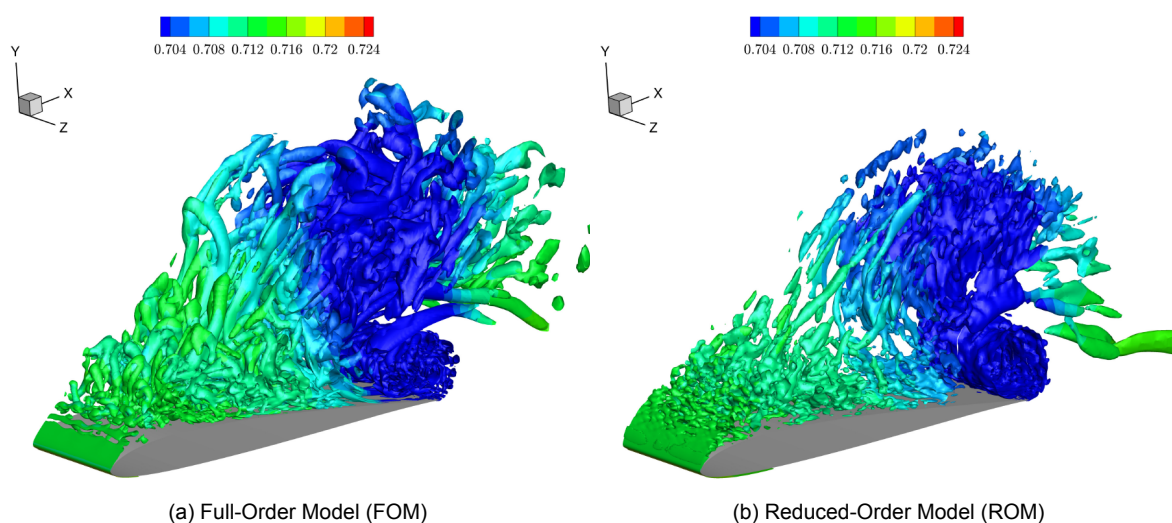


Figure 3.4: 3-D flow-field prediction: representing the iso-surfaces of the Q-criterion. A comparison is drawn between a FOM and ROM for a time-step beyond the training window. Obtained from the work of Wolf et al. [85].



The field of ROM also faces significant challenges, one of the biggest is that ROMs are notoriously sensitive to parameter changes (e.g. boundary and operating conditions) [3, 23, 37, 79, 124, 131]. There is a general lack of robustness with respect to parameter changes and ordinarily the ROM must be rebuilt for new parameter settings [3, 79]. Amsallem et al. [3] provides a fitting summary with respect to the field of ROM:

”There are three important facts to keep in mind when considering a ROM: 1) it is typically constructed at a given operating point that can depend on multiple physical or modelling parameters, 2) its construction can be CPU intensive, 3) once constructed, it often lacks robustness with respect to changes in the operating point.”

Especially point 3 is of significance for this thesis, because design parameter changes (i.e different operating points) are a part of exploring a design space and optimisation studies.

### 3.3.2. Parameterized Reduced Order Models

The shortcomings, with respect the goal of this thesis, adjoint to ROMs can be overcome with so-called Parameterized ROMs (pROM). This concept is similar to a ROM, but a pROM specialises approximating high-order problems with a low-order model that can account for parameter changes [11]. A pROM cannot make future-state predictions, thus cannot predict a flow-field beyond the timesteps from the training data.

pROMs are proposed by Amsallem et al. [3], Brunton et al. [23], Shah et al. [131] and a review paper is written by Benner et al. [11]. The cited papers are very mathematical in nature and there are several approaches (e.g. Subspace angle, Riemann or Grassmannian manifold interpolation) to construct a pROM [131]. So far, not 1 approach is identified that truly excels, as the articles [3, 23, 37, 79, 131] make contradicting statements on the validity of each others pROMs. Moreover, from the aforementioned articles postulating a pROM, none has proven to be able to predict a flow-field for new design points. A fitting example of what current pROMs can predict is the lift-coefficient of a fighter-jet for new Mach numbers with relative high accuracy. Amsallem et al.[3] uses a pROM (which uses a Grassmann manifold) for this.

### POD-based Emulation

In this paragraph a framework is described that is postulated by Mak et al. [88], from Georgia Institute of Technology (Georgia Tech.) in 2018. In essence, this framework is a type of pROM but the author refrains from using that term and refers to it as a POD-based Emulation. This framework incorporates multiple algorithms and combines it into a high-fidelity design methodology [87]. In Fig.3.5 a comparison is drawn between a LES flow-field (as validation) and two predicted flow-fields for new design points using this framework. The flow-fields are turbulent flows in a swirl injector with varying geometries.

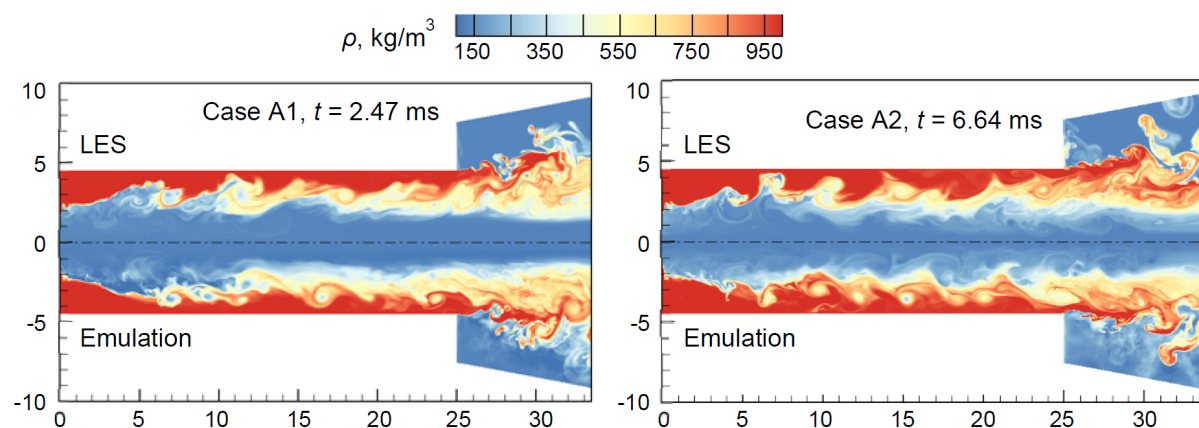


Figure 3.5: A predicted (density) flow-field for new design points (i.e. A1 and A2) using a POD-based framework and compared/validated with the LES flow-field.

In total 6 different articles [27, 28, 87, 88, 154, 165] are published by the research group from Georgia tech. on this topic. The articles [87, 88, 165] describe the framework based on the Common-Proper-Orthogonal Decomposition (CPOD)-Kriging algorithm. Whereas, the articles [27, 28, 154] are based on the Kernel-Smoothed-Proper-Orthogonal Decomposition (KSPOD)-Kriging algorithm. The methodology for both the algorithms/frameworks can be diluted to Fig. 3.6.

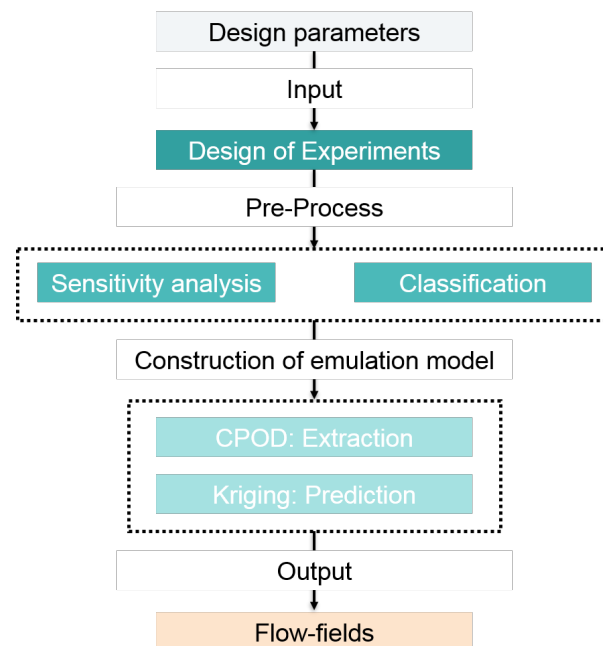


Figure 3.6: Methodology of the POD-based Emulation framework. Above is depicted the framework that incorporates the CPOD-Kriging algorithm. The methodology is the same for KSPOD-Kriging, only the extraction is performed by the KSPOD algorithm.

The framework incorporates a pre-process phase, consisting of a sensitivity analysis and classification of data, which results in additional learning benefit and higher-accuracy results [88]. The construction of the emulator model consist of two steps, the first is an extraction step with a CPOD or KSPOD algorithm. These algorithms are derived from the well-known POD [13, 86] algorithm (described in Section 3.4.1). The second is a prediction step using a Kriging algorithm [93], that is as an advanced interpolation method [88]. Typically, a Kriging algorithm is used for 3-D Response-Surface Modelling [4, 26] but Mak et al. uses the algorithm as a 2-D line interpolation method [88].

Alongside the relative good approximation capabilities by the emulator (seen in Fig.3.5. Mak et al. describes very high speed-up times that allow for the computation of hundreds new design points within a day's time [88]. In the articles [27, 28, 154] that use the KSPOD-Kriging algorithm, a speed-up factor of 42.0000 is described. Thus new design points are computed 42.0000 times faster with the trained KSPOD-Kriging model than with a LES simulation.

### 3.3.3. Neural Networks

The following potential methods are based on a type of Neural Network (NN) and are applied to flow-field prediction for varying design parameters. The most in-line with the thesis goal is the method proposed by Bhatnagar et al. [14], that proposes the use of a Convolutional Neural Network (CNN) [25, 50, 66] for the prediction of flow-fields for new design points. An approximation model based on CNNs is trained on the data of 252 RANS CFD simulations [14]. These training simulations account for the following design parameters

- three different wing profiles,
- four high turbulent Reynolds numbers,
- 21 different angles of attack of the wing profile.

With the trained approximation model, predictions of RANS flow-fields can be made for different wing-shapes, Reynolds numbers and angles of attack. The following results were obtained for a new wing-profile, where the quantity of interest is the normalised x-component of the velocity field.

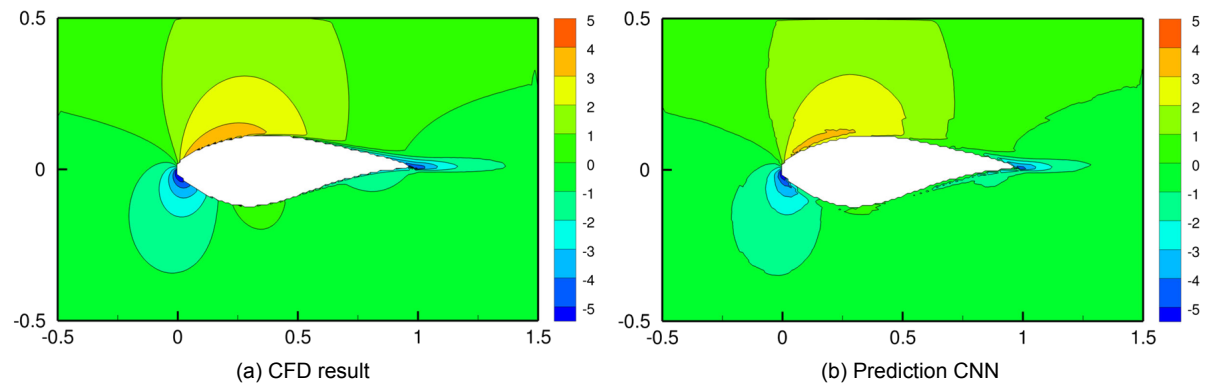


Figure 3.7: Comparison between the CFD flow-field and the CNN approximation, obtained from the work of [14].

Regarding accuracy, Bhatnagar describes errors between the 5% to 10% for different regions of the prediction. Speed-up times or computational time for training of such an approximation model are not mentioned. Bhatnagar expresses his desire to extend the algorithm to predict transient flow-fields [14].

Another work that implements a CNN is proposed by Guo et al. [57] that describes that his approximation model can estimate a velocity field two to four orders of magnitude faster than a CFD solver for a low error. Other potential methods includes the method proposed by: 1) Yu et al. [166] that combines an Artificial Neural Network (ANN) and a POD algorithm to predict flow-fields and even shock waves in the flow-field, and 2) White et al. [155] postulates a Cluster-Network (trained on a sparse-dataset) for predictions of key dynamical features of a flow-field.

### 3.3.4. Regression methods

The last group of the potential algorithms are based on regression algorithms, that is a very small group. Zhao et al. [169] constructs a ROM with a Least-Squares Support Vector Regression (LSSVR) for predicting thermal characteristics of a stratospheric airship. The ROM is able to predict the temperature distribution of the airship for six varying design parameters [169]. Zhao et al. compares the prediction quality for varying sizes of training data-sets and states that large data-sets (exceeding 700 training samples) are required for relative accurate results. Narasingam et al. [99] presents an adaption of the aforementioned SINDY algorithm [22] for identification of interpretable ROMs. Narasingam et al. implements his sparse regression on a hydraulic fracturing process to approximate the dynamics of the strongly non-linear problem [99]. Lastly, Gross et al. [56] presents the use of a regression forest for data-driven fluid simulations. High speed-up times were reported by Gross et al. accompanied by relatively high accuracy errors.



### 3.3.5. Selection

In the following section the selection process is described for the potential algorithms. On the next page, Table 3.1 lists and categorises all the potential algorithms in 3 groups. The algorithms are categorised according to their fundamental working principle. Thus, the algorithms based on a type of: 1) Modal-Decomposition Method (MDM), 2) Neural network (NN) and 3) Regression method. To narrow down the algorithms the following 4 criteria are used.

- **Complexity.** Of both the theory and implementation of the ML algorithm.
- **Accuracy.** Of the ML predicted flow-field compared to a LES flow-field.
- **Speed-up potential.** Of the computational time of a ML predicted flow-field compared to a LES flow-field. This includes the computational steps that were performed in advance: gathering training data, training, validation and basically any computational step before an accurate prediction can be made.
- **(Additional) Learning benefit.** Provided by the ML algorithm.

The speed-up potential is largely defined by the required quantity of training data. Computation of LES simulations are computationally expensive and typically trumps the succeeding computational steps. The algorithms that require less training data/LES simulations have high speed-up potential. To make separate distinctions for each computational step, for every ML-algorithm, is not possible because articles sparsely provide these details.

The fourth criteria can be seen as an incentive to choose algorithms that provide additional learning benefit of the original problem. A part of the objective of this thesis is to (at least) retain physical insight when exploring a design space. Thus algorithms that provide additional learning benefit are favourable. As an example, a POD algorithm decomposes a flow-field into 3 interpretable components [24], whereas a typical NN is more a black-box type of algorithm.

A disclaimer is appropriate for Table 3.1. It is not uncommon that important information is absent in articles that are needed to judge one or more criteria. For these cases, attempts have been made to become more familiar with the algorithms to still judge the criteria. Nonetheless, some criteria are left blank when this is not possible and/or realistic. The score for the 4 criteria are judged as objectively as possible.

Notes on Table 3.1:

- No weighting factors are used.
- The table includes algorithms that are not discussed in this document, see the cited articles for more information.
- Algorithm number 10 account for all pROMs based on different types of Manifold Interpolation (see Section 3.3.2).

Table 3.1: Table of potential algorithms

Number	Methods	Articles	Complexity	Speed-up	Accuracy	Learning benefit	Score
Modal decomposition methods							
1	CPOD - Kriging	[87, 88, 165]	3	5	3	5	16
2	KSPOD - Kriging	[27, 28, 154]	2	4	4	5	15
3	SPOD - Interpolation	[132, 148]	2	3	4	5	14
4	POD - DEIM	[29, 35, 102]	3	3	4	5	15
5	POD - Galerkin	[6, 8, 20, 118, 162]	3	4	2	5	14
6	POD - Gappy	[20, 126]	2	3	3	5	13
7	POD - Manifold interpolation	[3, 131, 170, 171]	2	4	4	5	15
8	Resolvent analysis	[124, 143]	3	N/A	3	3	9
Neural Networks							
9	POD - ANN	[166]	2	N/A	4	5	11
10	POD - DNN	[85]	2	3	4	5	14
11	Cluster-network	[155]	2	3	3	2	10
12	CNN	[14, 57, 147]	3	2	4	2	11
13	ANN	[2, 113]	2	N/A	4	2	8
14	DNN	[116]	2	N/A	5	2	9
15	Hidden physics models	[114, 115]	3	3	3	3	12
16	Super resolution	[49, 163]	2	1	4	2	9
Regression methods							
17	Sparse regression	[99, 145]	3	4	2	3	12
18	LSSVR	[169]	3	-	3	2	8
19	Regression forest	[56]	3	4	2	2	11
Legend	5 = Positive		Low	High	High	High	High
	1 = Negative		High	High	Low	Low	Low

### Discussion algorithm selection

The following general statements are made based on Table 3.1.

- MDM excel in offering good speed-up potential and additional learning benefit.
- Neural Networks have potential, but are relatively complex and offer mediocre speed-up potential due to their requirement of large amount of training data.
- Potential regression algorithms are sparse and hardly provide any additional learning benefit.
- Other types of algorithms (e.g. Genetic Algorithms, Field-Inversion, *k - means*) are identified as not suitable for the goal of this thesis

Based on Table 3.1 and the underlying literature, it is concluded that the chosen ML-algorithm must be based on a MDM. A MDM (e.g. POD, DMD) excel in speed-up potential due to their ability to decompose and express a flow-field with limited training data. Furthermore, because of the decomposition the flow-field data is expressed into 3 interpretable components which aids in the learning process.

The first chosen algorithm is from the POD-based Emulation framework, that produced the CPOD-Kriging (No.1) and KSPOD-Kriging (No.2) algorithm. The shown results (i.e. Fig.3.5) and described speed-up potential in these articles are remarkable. Thus, this framework provided a good starting point for a first implemented algorithm. Under certain conditions, that are explained in Chapter 5, can the CPOD-Kriging be simplified to a POD-Interpolation algorithm. The methodology for the POD-Interpolation, CPOD-Kriging and KSPOD-Kriging is practically identical and theoretically should provide nearly the same accuracy. Because the POD-Interpolation provides nearly the same accuracy and is the easiest to implement, it is chosen as the first algorithm. The POD-Interpolation approach is fundamentally not different as the CPOD-Kriging, nor the KSPOD-Kriging. As most of the algorithms in Table 3.1, the POD-Interpolation approach consist of two combined algorithms (i.e. POD and Interpolation). The mathematical background of the POD algorithm is described in Section 3.4.1, the methodology is described in Section 5.1 and results are provided throughout Chapter 5.

For reasons that will become clear in Chapter 5 also a second ML algorithm is implemented during this thesis. The second chosen algorithm is the proposed sDMD-Scaling algorithm. The reasons for choosing this algorithm are discussed in Section 5.4. The mathematical background for the sDMD algorithm is described in Section 3.4.2, the methodology in Section 6.1 and results are provided throughout Chapter 6.

### 3.4. Modal Decomposition Methods

What follows is the theoretical backbone of the POD and DMD algorithm. The POD and DMD algorithm make up the bulk of the computational steps of the POD-Interpolation and DMD-Scaling approach. Therefore, each is given a dedicated section to their working principles. Section 5.1 and Section 6.1 provide respectively an overhead view of the POD-Interpolation algorithm and DMD-Scaling algorithm.

#### 3.4.1. Proper-orthogonal decomposition

The POD method was first introduced in Section 3.1 as an example for an algorithm that has many different names (e.g. PCA, Karhunen-Loève expansion, Hotelling analysis) and variations (e.g. CPOD, KSPOD, BPOD). This method was first introduced to the fluid mechanics community by Lumley in 1967 [86], that enables someone to decompose almost any flow into a infinite set of eigenfunctions [109]. Since its introduction the method has been become very popular and is applied to a wide variety of problems including digital image compression [119], signal processing [150], bioinformatics [152], flow in stirred tanks [40, 84], multiphase heat transfer [18] and review papers are widely available [13, 60]. The POD method as proposed by Lumley expresses the original flow-field into a linear set of coefficients ( $a_m$ ) and eigenvectors ( $\vec{\phi}_i(\vec{x}, t)$ ) that are mathematically written as [86].

$$\vec{u}(\vec{x}, t) - \bar{\vec{u}}(\vec{x}) = \sum_m^{\infty} a_m \phi_m(\vec{x}, t) \quad (3.1)$$

A defining feature of Eq.3.1 are the eigenvectors that are functions of space and time. Since the seminal article of Sirovich in 1987 [134], and his "Snapshot" POD method has further sought to decompose space and time and thereby only obtaining spatial eigenvectors. In that context, Eq.3.1 is written as [134]

$$\vec{u}(\vec{x}, t) - \bar{\vec{u}}(\vec{x}) = \sum_m^{\infty} a_m(t) \phi_m(\vec{x}) \quad (3.2)$$

where the coefficients  $a_m(t)$  are time dependent and the modes are time independent. The use of the two forms depends on the problem at hand, more specifically the flow properties and the information wished to extract [145]. Irrespective of the Lumley approach or Snapshot approach, the key features of a POD method are listed below.

- Optimally: the POD modes are optimal in the  $L^2$  sense [86]. This has the following implications, the POD decomposition minimises the mean-square error between the flow-field and its truncated representation. Furthermore, it minimises the number of modes required to describe the original problem [72]. Lumley states that among all other linear decomposition methods this is the most efficient in terms of representing flow-field for a given number of modes [86].
- Ranking: the POD modes are ranked according to each mode energy content in the  $L^2$  sense [145]. As an example, if a POD decomposition is performed on a velocity or vorticity flow-field than the modes are ranked in decreasing order in terms of their kinetic energy or enstrophy [86]. Ranking is important for two things: 1) identification of key dynamical features in a flow-field, and 2) dimension reduction. The energy content is used as a measure for dimension reduction, where a truncated amount of  $r$  modes expresses the flow of  $m$  modes to a large degree. The following criteria to determine  $r$  is typically used [87, 145]

$$r = \frac{\sum_{l=1}^r \lambda_l}{\sum_{l=1}^m \lambda_l} > c_{set}, \quad (3.3)$$

where  $c_{set}$  is defined as the set criteria for captured energy/variance. With the reduction in modes, the flow field is represented by the truncated series.

- Orthogonality: is a useful property in the construction of ROMs [105, 145] because it simplifies the projection step (which is a typical step in a ROM procedure) [124].

Figure 3.8 captures the essence of POD and how it decomposes a flow-field into a set of modes.

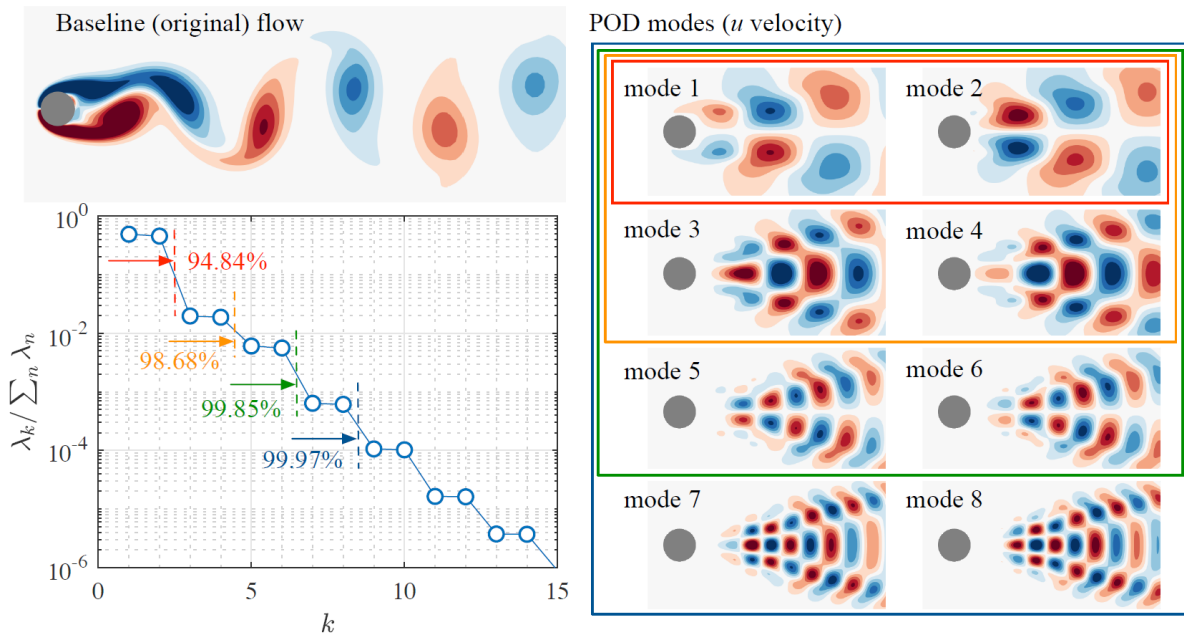


Figure 3.8: POD analysis of a von Karman Vortex Street. Top left depicts the original vorticity flow field. Bottom left depicts the ratio of variance captures by modes ( $k$ ) and on the right shows the first 8 POD modes. Obtained from the work of Taira et al.[146]

In the previous text, general information is given on the POD method and two governing equations are given that produce spatio-temporal modes and spatial modes. The two equations stem from two different POD methods, the Classic "Lumley" POD and the "Snapshot" POD. POD is one of the central theme of this research and therefore both of these methods are described in the two paragraphs below.

### Classic "Lumley" POD

The following section describes the original POD method proposed by Lumley in 1967 [86]. Let  $u_i(\vec{x}, t)$  represents a zero-mean stochastic process (flow-field) in space and time and  $\phi(\vec{x}, t)$  a deterministic function that is to approximate the stochastic function in a  $L^2$  sense [13, 109]. Subsequently the square of the inner product [109]

$$\langle |u_i(\vec{x}, t) \phi_i(\vec{x}, t)|^2 \rangle = \langle |\alpha|^2 \rangle \quad (3.4)$$

needs to be maximised and where

$$u_i(\vec{x}, t) \phi_i(\vec{x}, t) \equiv \int_t \int_{\vec{x}} u_i(\vec{x}, t) \phi_i(\vec{x}, t) d\vec{x} dt \quad (3.5)$$

defines the inner product in both space and time [109]. Squaring the integrals, averaging and maximising yields to the Lumley integral equation [109]

$$\int_t \int_{\vec{x}} \mathbf{R}_{ij}(\vec{x}, \vec{x}', t, t') \phi_j(\vec{x}', t') d\vec{x}' dt' = \lambda \phi_i(\vec{x}, t) \quad (3.6)$$

where  $\mathbf{R}_{ij}(\vec{x}, \vec{x}', t, t') = \langle u_i(\vec{x}, t) u_j(\vec{x}', t') \rangle$  is the two-point, two-time correlation tensor. Equation 3.6 reduces only under certain circumstances to the well-known POD equation (Eq.3.1), but more often it does not [109]. Only when the finite total energy property is satisfied, in all directions and time, does the integral equation yield the classically known POD solutions. The finite total energy property is fulfilled by any flow that is bounded in space or die off rapidly enough in all directions and time [109]. The property is mathematically expressed as

$$\int_t \int_{\vec{x}} \langle u_i(\vec{x}, t) u_i(\vec{x}, t) \rangle d\vec{x} dt < \infty. \quad (3.7)$$

"Finiteness" is per definition satisfied if a field is of finite extent or time, which is the case for all CFD simulations. However, "finiteness by truncation" is artificial and can result in POD solutions that have nothing to do with the original flow and more with the imposed boundary conditions [109]. Pollard et al. [109] advises always to be wary of the results and to assess the validity of the basis functions. For the case that the flow is of finite total energy and the energy integral converges, only then the solutions are referred to as "classical POD solutions" as proposed by Lumley et al.[86]. The aforementioned correlation tensor ( $\mathbf{R}_{ij}$ ) is reconstructed to [109]

$$\langle u_i(\vec{x}, t) u_j(\vec{x}', t') \rangle = \sum_{m=1}^{\infty} \lambda_m \phi_i^m(\vec{x}, t) \phi_j^m(\vec{x}', t'). \quad (3.8)$$

And with this, the flow field is represented by the POD decomposition [109]

$$u_i(\vec{x}, t) = \sum_{m=1}^{\infty} a_m \phi_i^m(\vec{x}, t) \quad (3.9)$$

using the coefficients  $a_m$  that are determined by

$$a_m = \int_t \int_{\vec{x}} \phi_i^m(\vec{x}, t) u_i(\vec{x}, t) d\vec{x} dt. \quad (3.10)$$

For more background information see the work of Lumley [86], Pollard et al. [109] and George [52].

### Space-only "Snapshot" POD method

In this paragraph the most commonly employed POD method [109, 145] is discussed, the Snapshot POD that is introduced by Sirovich [134] in 1987. First the mathematical background is given and secondly the reasons why this is the most used POD method. The Snapshot method takes a collection of flow-field snapshots at discrete time-steps. Similar to the Lumley POD approach, an inner product needs to be maximised (basically the space-only form of Eq.3.5) that allows for the formulation of the following equation [134]

$$\int_{\vec{x}} \mathbf{R}_{ij}(\vec{x}, \vec{x}') \phi_j(\vec{x}') d\vec{x}' = \lambda \phi_i(\vec{x}) \quad (3.11)$$

where the two point correlation  $\mathbf{R}_{ij}$  tensor is computed as [134]

$$\mathbf{R}_{ij} \equiv \langle u_i(\vec{x}) u_j(\vec{x}') \rangle = \sum_{m=1}^M \langle a_m^*(t) a_m(t') \rangle \phi_i^{m*}(\vec{x}) \phi_j^m(\vec{x}'). \quad (3.12)$$

Contrary to the Lumley POD, the POD integral (Eq.3.5) is replaced by a finite difference approximation [109]. And for a statistically stationary flow, the approximation becomes a summation not over space but over snapshots [134]. Sirovich proved that the same dominant modes can be found from a snapshot formed correlation matrix than spatial-temporal correlation matrix [145]. By doing so, the computational cost become much more manageable because the number of snapshots are far less than the number of spatial points [109].

Sirovich basically proposed a separation of variables technique, that makes it much more computationally tractable. Simultaneously, it raises doubts about the physical validity [109] as fluid dynamics is governed by spatio-temporal physics. Regardless, from Eq.3.12 the governing equation of the Snapshot POD is derived (see [134]) and defined as

$$\vec{u}(\vec{x}, t) - \bar{\vec{u}}(\vec{x}) = \sum_m^M a_m(t) \phi_m(\vec{x}). \quad (3.13)$$

Now lets address why the Snapshot POD became so popular, as described above the same dominant modes can be found but from a much smaller correlation tensor. Sirovich reduced the size of the correlation matrix to (Number of snapshots ( $s$ )  $\times$   $s$ ) instead of (Number of spatial points ( $n$ )  $\times$   $n$ ). Therefore, the Snapshot method is usually applied when the flow-fields/data comes from a CFD programme where spatial size of the data is large [145]. Secondly, Lumley's approach relies on solving an 4-D integral (Eq.3.10) that is computationally expensive [145].

To conclude, the Snapshot POD is computationally much more attractive but simultaneously physically less valid [109]. If the discovery of physically accurate coherent structures is key than a Lumley POD is better suited. If computationally efficiency is a priority, than a Snapshot POD is better suited [145]. Therefore, in this thesis the Snapshot POD is implemented.

### 3.4.2. Dynamic mode decomposition

Dynamic mode decomposition (DMD) is a powerful data-driven method for analysing complex dynamical systems. Using data from CFD simulations, DMD extracts vital dynamical characteristics and expresses them into modes where each mode is of a singular frequency and a decay rate. DMD has been implemented in many different fields such as finance [90], video processing [55], epidemiology [112], robotics [12] and neuroscience [19]. For an extensive overview of DMD application to fluid systems, see Table 3 in the work of Rowley et al.[124]. The use of DMD is so far mostly limited to diagnostic purposes and less for future-state predictions, optimisation and control purposes [145].

Since its introduction by Schmid [127] in 2008, DMD has quickly gained popularity due to its capability to accurately extract dynamics systems of linear systems and with the possibility to be expanded to nonlinear systems[159]. The extraction is achieved by fitting a high-degree matrix polynomial to the input data-sequence. A between snapshot linear map is then determined to approximate the system dynamics in a low-order way. This map is calculated by processing the data and represents the optimal linear operator that corresponds to the evolution of the flow of a small time interval. With the eigenvalues and eigenvectors of the linear map the original flow can be reconstructed. Now the mathematical background is presented, this algorithm is obtained from the work of Tu [76] and slightly diverges from the original provided by Schmid [127]. The difference is however minute, the original algorithm was formulated in a form that highlights its connection to the Koopman operator theory [24]. Furthermore, the method postulated by Tu uses a Singular-Value Decomposition (SVD) algorithm that is more numerically stable [76]. The CFD data is represented in a sequential set of data vectors ( $\mathbf{D}$ ).

$$\mathbf{D}_t = \{D_1, D_2, \dots, D_n\}, \quad (3.14)$$

where each  $t^{th}$  value stands for an equidistant time-step. If the data stems from a nonlinear process the linear mapping  $\mathbf{A}$  amounts to a linear tangent approximation that connects the flow field  $D_t$  to its temporal neighbour  $D_{t+1}$  [24]. The Exact DMD algorithm is given by the following steps and the code for this algorithm is given on the GitHub page[5].

1. Arrange the data  $\mathbf{D}_t$  into two matrices

$$\mathbf{X} \triangleq [D_0 \dots D_{t-1}], \quad \mathbf{Y} \triangleq [D_1 \dots D_t]. \quad (3.15)$$

Where the connection between the two matrices is determined by the best-fit operator  $\mathbf{A}$  which is defined as [24]

$$\mathbf{A} = \underset{\mathbf{A}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F. \quad (3.16)$$

For high-dimensional problems the matrix  $\mathbf{A}$  has  $n^2$  elements and its eigenvalue decomposition is often intractable [24]. Instead, the algorithm uses dimensionality reduction to determine the dominant eigenvectors and eigenvalues of  $\mathbf{A}$  without explicit computations using  $\mathbf{A}$ . This is done by projecting  $\mathbf{A}$  onto leading singular vectors and results in a significant smaller matrix  $\tilde{\mathbf{A}}$ .

2. Compute the SVD of  $\mathbf{X}$  as

$$\mathbf{X} = U\Sigma V^* \quad (3.17)$$

where  $U$  is  $n \times r$ ,  $\Sigma$  is diagonal and  $r \times r$ ,  $V$  is  $m \times r$  and  $r$  is the rank of  $\mathbf{X}$ . In practice choosing the approximate rank  $r$  is one most important and subjective steps in DMD [24]. This concept is further elaborated in Section 6.2.1.

3. Form the matrix:  $\tilde{\mathbf{A}} = U^* Y V \Sigma^{-1}$ .
4. Perform an eigenvalue decomposition of  $\tilde{\mathbf{A}}$ . The eigenvalues ( $\Lambda$ ) and eigenvectors ( $W$ ) are given by  $\tilde{\mathbf{A}} W = \Lambda W$
5. With the gained information from the second eigenvalue decomposition the following components can be computed. The spatial modes  $\phi$ , the amplitude of every mode  $b$  and the temporal modes  $\omega$ .

$$\vec{\phi} = \mathbf{Y} (V^* \Sigma^{-1}) W \quad (3.18)$$

$$b = (W \Lambda) / (\Sigma V^*) \quad (3.19)$$

$$\omega = \log(\Lambda) / \Delta t \quad (3.20)$$

6. Flow-field reconstruction

$$\mathbf{D}_t = \sum_{m=1}^r \vec{\phi}_m e^{\omega_m t} b_m \quad (3.21)$$

Similar to the POD algorithm that decomposes a flow-field into spatial modes and temporal modes, the DMD algorithm decomposes a flow-field into spatial modes, temporal modes and mode amplitudes. The DMD temporal modes can be dissected into the complex valued decay rate (real component) and frequency (imaginary component) of each mode [24]. The DMD method can be seen as combining the favourable aspects of POD and discrete Fourier transform [145]. This becomes clear when the temporal modes for the POD and the DMD are compared. A known disadvantage of the POD algorithm is that it produces temporal modes that almost always contain a mix of frequencies [24, 124, 145]. DMD on the contrary produces temporal modes of a single frequency that is a large advantage DMD has over POD. Figure 3.9 is an illustrative figure that compares result obtained from a POD and DMD. The fundamental DMD mode is similar to the dominant POD mode, the secondary modes differ from each other. The secondary DMD mode can be clearly identified to an actuation frequency of 8.8 Hz, however pinpointing POD modes to a specific frequencies is impossible [69, 145].

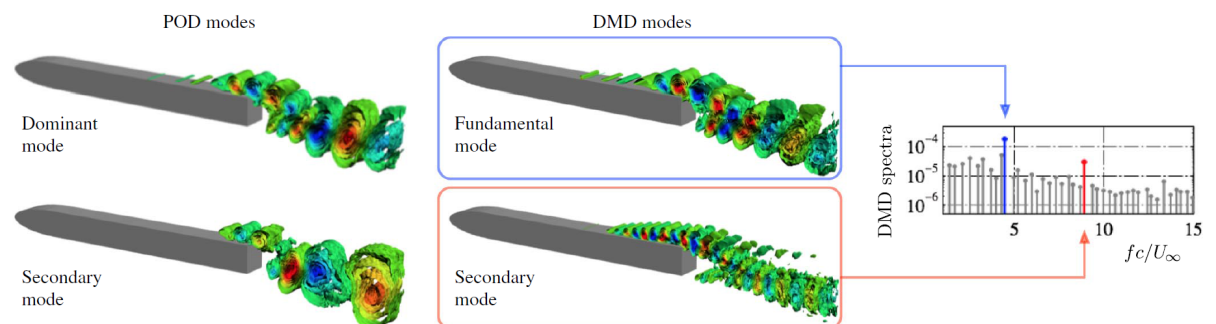


Figure 3.9: Comparison between two dominant POD and DMD modes for a 3D turbulent flow over a flat-plate wing. Figure is from the work of Brunton et al. [145].

The DMD algorithm has become quite popular and multiple variations of this approach exist (e.g. Extended DMD, Randomized DMD) each formulated for different and specific purposes. For the sake of brevity not all are discussed in this section. One variation must however be briefly described which is the Sparsity-Promoting DMD (sDMD) that is proposed by Jovanovich et al. [68, 69]. The sDMD solves one of the larger disadvantages of the Exact DMD algorithm. A disadvantage of the DMD algorithm is that there is no ranking of the modes [145], the modes are actually scrambled. Basically the relevant dynamical features are scrambled between the irrelevant dynamical features. This makes the identification of key dynamical features and dimension reduction very complicated [145]. Manual selection of key dynamical features is impractical and most of the time unrealistic. The sDMD solves this problem by promoting sparse selection of the most important modes.



### Sparsity-Promoting DMD

The sDMD is designed to achieve a desirable trade-off between the accuracy of its approximation (again in the  $L^2$  sense) and the number of modes used for the approximation [69]. Jovanović extended the regular formulation of the DMD (as previously described) with a penalty function that encourages sparse mode selection. The essence of the method can be diluted to Eq.3.22, where the variables  $J(b)$  and  $\gamma$  are introduced that respectively represents the objective function ( $L^2$  minimisation equation between the data and its approximation) of the DMD algorithm and the regularisation/penalty parameter [69].

$$\underset{b}{\text{minimise}} \quad J(b) + \gamma \sum_{m=1}^r |b_m| \quad (3.22)$$

The additional term in Eq.3.22 penalises the number of non-zero entries in the vector amplitudes  $b$  and encourages to eliminate features that hardly contribute to the approximation [69]. Equation 3.22 is a convex optimisation problem (mathematically proven in [68]) for which Jovanović uses an Alternating-Direction Method of Multipliers (ADMM) algorithm. The ADMM algorithm is a well-established algorithm that is designed for these large-scale convex optimisation problems, for more information on this algorithm see Boyd et al. [16].

One of the advantages of the sDMD, compared to the Exact DMD algorithm, is that it does not require the subjective choosing and thus commitment to a certain rank of truncation (as in step 2 of the DMD-algorithm). The sDMD performs the optimisation problem (Eq.3.22) and selects the most important modes for the approximation. The only input that is required is the regularisation parameter ( $\gamma$ ), the higher the  $\gamma$ -value the higher the emphasis on sparse mode selection [69]. The optimal selection of this  $\gamma$ -value is a topic of active research [53, 69]. In this thesis first a heuristic approach was used, that is succeeded by the parameter selection method as proposed by Wohlberg [160]. The method provides an adaptive regularisation parameter based on optimisation residual (primal ( $r$ ) and dual ( $s$ ) residuals) balancing. The following regularisation parameter is used for all sDMD computations [160].

$$\gamma^{i+1} = \begin{cases} 2\gamma^i, & \text{if } \|r^i\|_2 > 10\|s^i\|_2 \\ \frac{1}{2}\gamma^i, & \text{if } \|s^i\|_2 > 10\|r^i\|_2 \\ \gamma^i, & \text{otherwise} \end{cases} \quad (3.23)$$

Where  $i$  denotes the optimisation iterations. This scheme is proven to be effective for a wide variety of optimisation problems [160]. But more importantly it resulted in quick convergence of the optimisation problems in this thesis.

### 3.4.3. Comparison algorithms: POD versus DMD

In the last section of Section 3.3 a comparison is drawn between the two governing algorithms. An emphasis is put on comparing 4 key aspects with respect to the goal of this thesis.

1. **Sparsity.** Lumley [86] proved that the POD algorithm is optimal. For a given number of modes it provides the best representation of the input data compared to any other linear decomposition methods [86]. When sparsity (sparse data-sets results in increased computational efficiency) is important, a POD algorithm is thus better suited than a DMD algorithm.
2. **Truncation.** For truncation it is important that a type of ranking of the modes is present, such as the ranking of POD modes according to their energy/variance content. DMD modes are not ranked and complicate the identification of key modes and dimension reduction.
3. **Decomposition.** Both algorithms decompose a flow-field into 3 components. The POD decomposition results in a mean field, spatial modes and (aperiodic) temporal coefficients with a distribution of multiple frequencies. The DMD decomposition results in spatial modes, mode amplitudes and (periodic) temporal coefficients which each account for a singular frequency. Especially, this periodicity of the temporal coefficients is important and is extensively described in Section 5.4 and Chapter 6.
4. **Learning benefit.** The DMD decomposes the data into complex values and thus provides more information. For example, the real component of the temporal coefficients relates to growth or decay rate, whereas the imaginary component represents the frequency of the coefficient [24]. A POD algorithm only provides real values and therefore does not provide any extra information compared to a DMD algorithm.

In Chapter 5 it becomes clear why a second approach is needed. This subsection provided, ahead of Chapter 5 and Chapter 6, a compact comparison between the two algorithms.

## 3.5. What this work adds

Shown in the previous sections is that the prediction of flow-fields, using ML algorithms, is not a new field of study. This work intends to add to the field in the following ways:

- **Providing a review of all potential ML algorithms.** The vast majority of cited articles postulate a single ML algorithm, for their respective goal. There are no review papers on the prediction of flow-fields, this can be considered a gap in the literature. This literature review intended to provide an overview of all the potential ML algorithms.
- **Comparing and selecting a ML algorithm.** A comparison is drawn between the ML algorithms using four (universal) criteria. In general, a comparison between a large variety of algorithms is currently lacking in papers. With the help of the criteria the most suitable algorithm (POD-Interpolation) is chosen.
- **Proposing the sDMD-Scaling algorithm.** In this work a new algorithm is proposed that overcomes the root-cause from the POD-Interpolation approach. Furthermore, the algorithm is shown in Chapter 6 to be capable of predicting relative accurate flow-fields.

The approach used in this work is unique in the sense that: 1) a wide-variety of algorithms are analysed, 2) 1 high-potential algorithm is selected, implemented and evaluated, 3) 1 new algorithm is proposed. And more importantly, this unique approach allows for a substantiated verdict on:

- **The merit of combining CFD and ML for the prediction of flow-fields.** A verdict is currently lacking in literature as each article promotes an algorithm, but do not provide a critical note on the added value of ML (for this purpose).

This last point is, according to the author, the real added value of this work.

### 3.6. Conclusions literature review

At the start of Chapter 3, five points/question are formulated that this section intends to answer. Section 3.1 resulted in a clear-definition of ML: "Machine Learning is using data to create a model and then using that model to make predictions". By having a clearly defined ML term: 1) a degree of clarity is provided and more crucially, 2) defined borders for the search-field for potential algorithms.

This is succeeded by Section 3.2 that answers the second question regarding the unique challenges posed by fluid dynamics for ML. It is found that besides the strong nonlinearities, also the characteristics of fluid dynamics data is challenging. For example, having tall (spatial domain) and skinny (temporal domain) heterogeneous data-sets is not ideal. Furthermore, the active research fields for implementing ML are identified as: the flow control field (Section 3.2.1), identification of essential dynamical features (Section 3.2.2) but also for turbulence modelling [168].

Section 3.3 revolves around the goal of identifying, judging and selecting potential ML algorithms. Table 3.1 lists 18 potential holding algorithms. An important finding from this section and table, is that Neural Networks require excessive amounts of training data for accurately predicted flow-fields and thus are unsuitable. Only ML-algorithms based on a type of MDM (e.g. POD, DMD) are found to be suitable. This has led to the first chosen algorithm that is the POD-Interpolation which provides high-speed up potential, learning benefit and is relatively straightforward. Furthermore, a second algorithm is implemented in this work, the proposed sDMD-Scaling approach. The reasons why a second algorithm is needed are described in Section 5.4.

Lastly, Section 3.4 provides the mathematical background of the employed algorithms. An important finding, beside the actual working principles, is that the Snapshot-POD [134] is used instead of the classic Lumley POD approach [86], due to computational efficiency. Furthermore, the sDMD algorithm is used instead of the exact DMD algorithm due to its capability to identify and select key modes.



# 4

## Application cases

Training data is essential for ML algorithms to be able to make predictions. The data can either be used for training and validation purposes. To test the efficacy of the ML algorithms two different application cases are used. Chapter 4 describes these application cases by providing:

1. Information on the governing physics.
2. Key results of each LES simulation.
3. The design space for each application case.

A representative optimisation study for Siemens (see Example 1, Chapter 1) optimises the design of fuel nozzles for a new combustion chamber design. However typical Siemens cases possesses very complex flow physics and require a relative dense meshes. The inherent complexity of these cases made them unsuitable for this research. This thesis focuses on analysing, comparing and implementing different ML algorithms to determine whether they are capable of tackling all 4 problems (see Chapter 1) that are adjoined to optimisation studies. To achieve this as **efficiently as possible** it is decided to use canonical fluid dynamics cases with relative simple domains. The two test cases are the Von Karman Vortex Street and a turbulent shear mixing layer.

The outline of this chapter is as following: Section 4.1 describes the Von Karman Vortex Street (previously seen in Fig.3.3 and Fig.3.8). Section 4.2 describes a turbulent shear mixing layer. Section 4.3 expands on the implications of two-dimensional CFD. Section 4.4 provide the concluding remarks.

### 4.1. Von Karman Vortex Street

The von Karman Vortex Street is one of the canonical problems of Fluid Mechanics [75] named after the fluid dynamicist Theodore von Karman. Examples of the Vortex Street are widespread throughout nature, for example weather patterns (Fig.4.1) and singing of telephone and electrical cables [103].

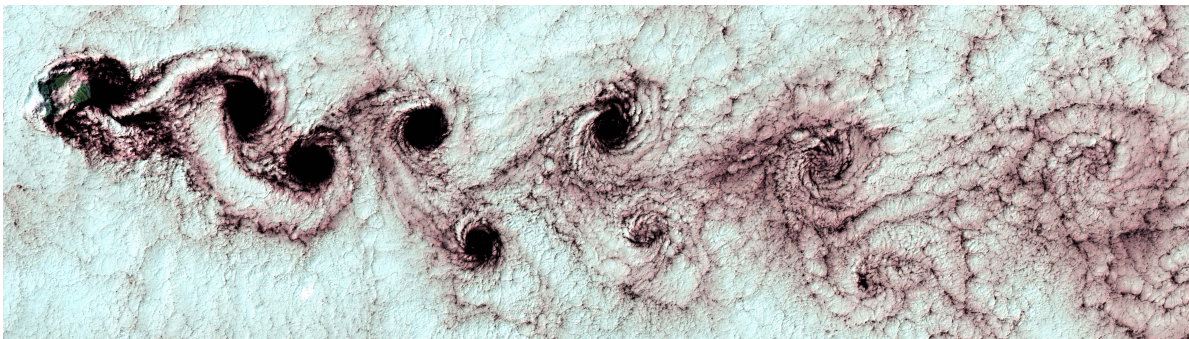


Figure 4.1: Wind-driven clouds in the shape of a Vortex Street. The vortices appeared over the Alexander Selkirk Island in the South Pacific Ocean [100].

When a fluid flows past a bluff body of a certain shape and size it will generally start to shed vortices, the exception being creeping flows ( $Re < 1$ ). When a Reynolds number of approximately 100 is reached [75], the wake becomes unstable and forms two staggered rows of vortices of opposite rotation. Furthermore, the wake slowly oscillates because the vortices separate alternately from the bluff body. Since the vortices are shed alternately, the forces experienced by the bluff body are also oscillating [75]. When the vortex shedding frequency matches the natural frequencies of the bluff body, it starts to vibrate and can lead to the singing of cables. The Strouhal dimensionless number is a key parameter for oscillating flows since it describes the relation between a length-scale of a bluff body, vortex shedding frequency and the flow velocity. The Strouhal number is defined as [142]

$$St = \frac{fL}{U}. \quad (4.1)$$

With the defined Strouhal number it is possible to describe the universal relation between the two dimensionless numbers governing a Vortex Street. The Strouhal-Reynolds describes the relation between the vortex shedding frequency and Reynolds number [142]. This relation is shown in Fig.4.2, where the cross-hatched exhibits the spread of experimental data describing this relation. The reasons for the spread in data is not discussed and the interested reader is directed towards Lienhard [83].

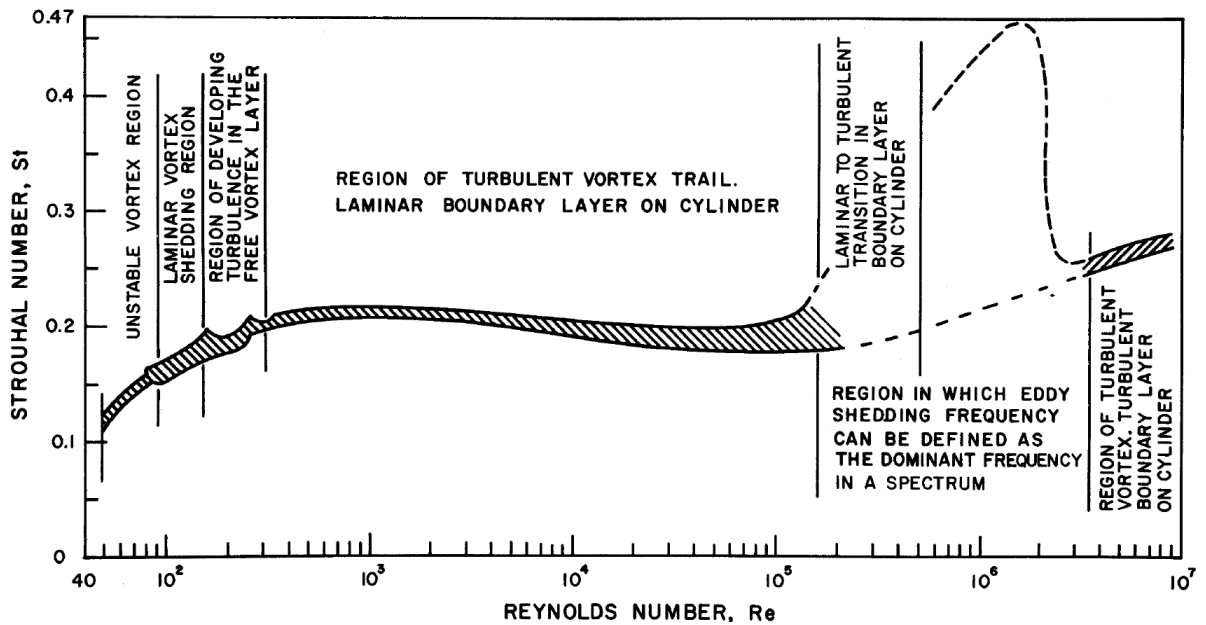


Figure 4.2: The Strouhal-Reynolds relation for rigid circular cylinders. Obtained from the work of Lienhard et al. [83]

As stated in Chapter 2, only turbulent flows are used in this thesis. Higher Reynolds numbers, through higher flow velocities, are generally attractive for practical reasons. Higher flow velocities result in lower flow-through times and higher vortex shedding frequencies. This is favourable, from a computational and post-processing point of view. These factors combined with the information from Fig.4.2 resulted in the chosen Reynolds number range,  $2500 < Re < 6000$ , for the CFD cases. For accurate vortex shedding frequency predictions the following piece-wise linear equations are used, defined by Fey et al. as [46]

$$St = \begin{cases} 0.2040 + 0.3364/\sqrt{Re}, & \text{if } 1300 < Re < 5000 \\ 0.1776 + 2.2023/\sqrt{Re}, & \text{if } 5000 < Re < 2 \cdot 10^5 \end{cases} \quad (4.2)$$

### 4.1.1. CFD set-up

The following subsection briefly describes key points of the CFD simulations, these are ordered similarly to the working procedure from StarCCM+. It should be emphasised that the focus of this thesis is directed towards ML algorithms (i.e. implementations, capabilities and drawbacks). To speed-up the whole learning process, coarse meshes are used throughout the project to obtain data that quickly can be read, manipulated and transferred. The consequence of using a coarse mesh is that for certain cases CFD criteria (e.g. Pope, CFL) are violated. Regardless of these violations, in Section 4.1.2 it is shown that the most important features for the Von Karman street are adequately captured.

This section describes the Geometry and Mesh, CFD Physics settings and CFD quality control. The used spatial and temporal discretization schemes and settings were discussed in Section 2.2.2

#### Geometry and mesh

The geometry used for the LES simulation is a quasi two-dimensional rectangle with a cylindrical cut-out on the centerline of geometry. The domain dimensions are given in Table 4.1 and key mesh statistics in Table 4.2. More information on the mesh statistics see GitHub [5].

Table 4.1: Dimensions domain

Parameter	Value (mm)
Length (X)	100
Height (Y)	25
Width (Z)	2
Radius cylinder	1.25
Origin cylinder	(40, 0, 0)

Table 4.2: Mesh statistics

Parameter	Value	Unit
Base size cell	0.60	mm
Number of cells	165412	[-]
Minimum volume change	0.06	[-]
Minimum face validity	1	[-]
Maximum skewness angle	60.97	deg.

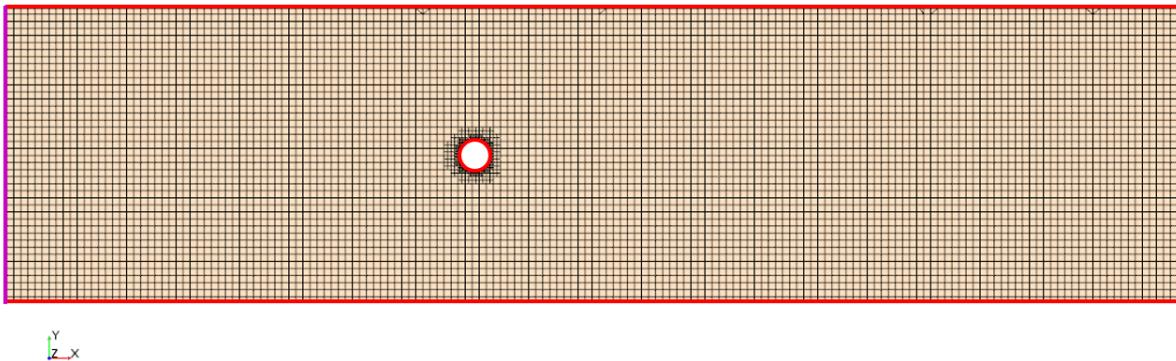


Figure 4.3: Cross-section (X,Y,0) of the mesh used for Karman Vortex Street CFD simulations. The different colours highlight the different boundary conditions that are described at the bottom of this page.

A structured mesh is used instead of a polyhedral or tetrahedral mesh, for the practical reason that this mesh resulted in fewer cells. Figure 4.3 depicts the coarse structured mesh for the quasi two-dimensional domain.

#### CFD settings

The specific StarCCM+ settings are tabulated in Table C.1. A time-step of 7.5 microseconds and a maximum of 6 inner iterations are used. There are four different boundary conditions used for the domain: 1) mass flow inlet (purple), 2) pressure outlet (green), 3) no-slip adiabatic walls (red) and 4) symmetry planes. The inlet is positioned on the left surface of the domain (perpendicular w.r.t. Fig.4.3) and the outlet is positioned on the right surface of the domain. Two symmetry planes are used at  $(X, Y, \pm 1)$  and the remainder of the surfaces are no-slip walls.

The operating conditions are: reference pressure of 1 Atmosphere, reference temperature of 300 Kelvin and a pressure outlet at -0.1 Bar (relative to the reference pressure). These are identical for all the LES simulations of the Von Karman Vortex Street.

### Design space

The mass flow inlet is the design parameter in the one-dimensional design space, and thus is basically a design sweep. In total 13 different design points are chosen with varying mass flows. These 13 design points can both be used for validation and for training of the ML algorithm.

Table 4.3: Design space of the von Karman Vortex Street application case.

No.	Deviation (%)	Mass flow (g/s)	Reynolds number	Vortex shedding freq. (Hz)
1	0	1.2500	2875	1794
2	5	1.3125	3019	1883
3	10	1.3750	3162	1970
4	15	1.4375	3306	2060
5	20	1.5000	3450	2147
6	30	1.6250	3737	2324
7	40	1.7500	4025	2500
8	50	1.8750	4312	2675
9	60	2.0000	4560	2853
10	70	2.1250	4887	3028
11	80	2.2500	5175	3198
12	90	2.3750	5462	3363
13	100	2.5000	5749	3527

The text above and Table 4.3 describe merely the boundary conditions. For accurate initial conditions a RANS simulation is performed beforehand (see Section 2.2).

### CFD quality control

This paragraph briefly describes how to assess the numerical quality of a CFD case. The first key aspect are the residuals that is the imbalance between a conserved quantity in each control volume. Low residuals are indicative for a numerically accurate solution [45]. The second aspect that is monitored, are the mean values of quantities of interest (e.g. pressure, velocity). Multiple probe points are placed in the domain that track these quantities over the total range of iterations or time-steps. Once the mean is constant or nearly constant, can the solutions be considered converged.

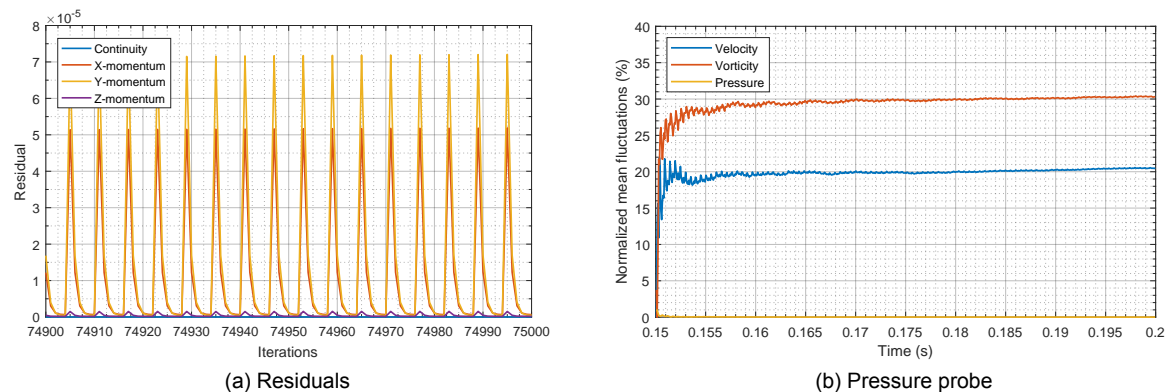


Figure 4.4: Residuals and normalised averaged values of quantities of interest

Figure 4.4 shows respectively a convergence plot and a normalised mean quantities plot. Normalised mean quantities are used to show 3 flow quantities in a single figure. Figure 4.4 (a) shows the last 100 iterations with very low residuals, whereas (b) depicts hardly any fluctuations. Lastly, the CFL-number and the Pope criterion are analysed, these are shown in Fig.C.2 and Fig.C.3. Areas can easily be identified where the Pope and CFL thresholds [82, 110] are exceeded. This is not ideal and indicates that the mesh is too coarse (as expected) from a strict CFD point-of-view. Regardless, as the next section shows the most important features are accurately captured by the CFD simulations.



### 4.1.2. Results

The following subsection highlights the pressure and vorticity flow-fields for multiple time-steps, and provides a graph of the vortex shedding frequency.

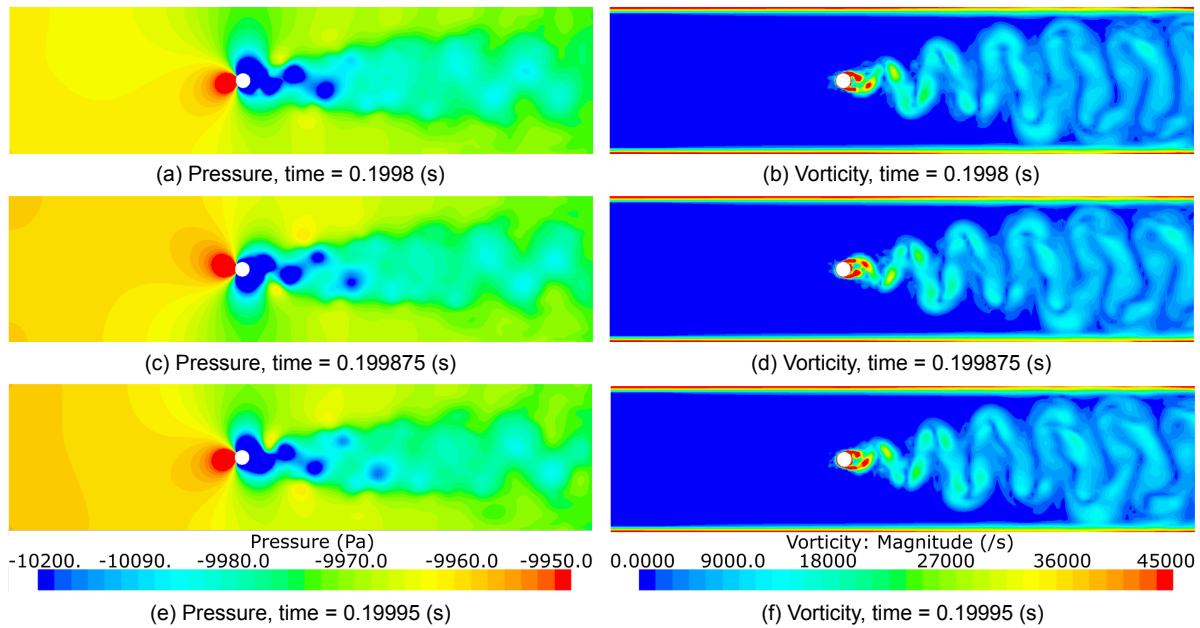


Figure 4.5: Six snapshots of a simulated von Karman flow-field for operating point number 8 (Table 4.3)

The snapshots from Fig.4.5 show the unsteady separation of the vortices and the asymmetric pressure distribution as described at the start of Section 4.1. The six snapshots represent the last three time-steps from the LES simulation of design point 1. The final figure that is provided is Fig. 4.6, that compares the simulated vortex shedding frequency with the computed vortex shedding frequency from Eq.4.2.

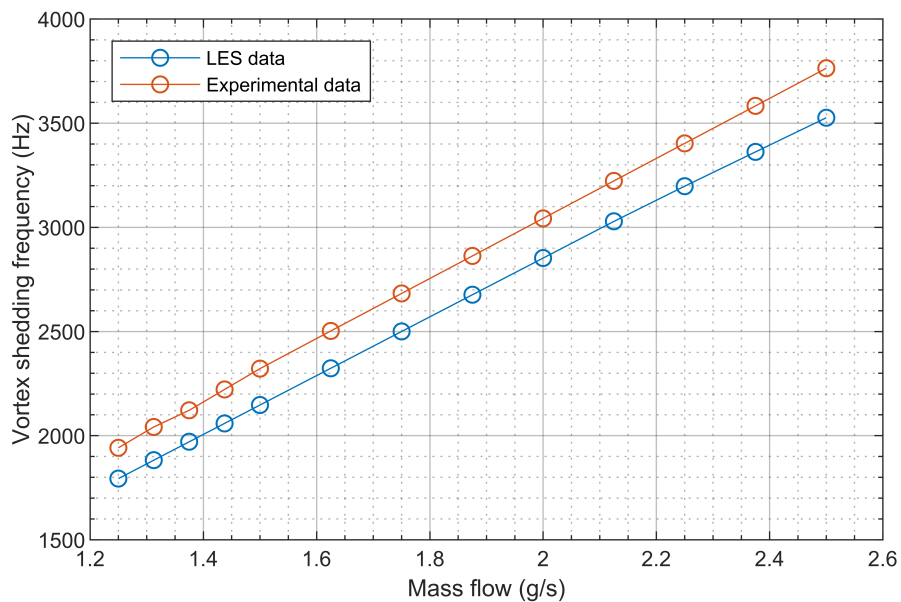


Figure 4.6: Comparison between the CFD and experimental vortex shedding frequency

### Discussion results

The linear growth of the vortex shedding frequency is relatively accurately captured. Albeit, there is a slight deviation between the simulated frequency and the experimental frequency. The observed deviation however is always well below 10%. The deviation most likely originates from the coarse mesh surrounding the cylinder, the boundary layers are not adequately resolved by the coarse mesh. This is: 1) indicated by the Pope-criterion and CFL number and 2) a mesh-dependency is observed by the author.

Typically in a high-quality CFD simulation more quantities are monitored and validated with experiments (e.g. angle of separation, drag coefficient and the length of a wake). It is decided not to evaluate and validate more quantities, since it diverges from the goal. A certain degree of inaccuracy in the CFD simulations is permitted, with respect to the goal of this thesis.

## 4.2. Turbulent Shear Mixing Layer

The second application is a turbulent shear mixing layer that is one of the canonical problems of Fluid Mechanics. The flow physics are governed by the Kelvin-Helmholtz (KH) instability [75], that occurs at the interface between two flows with different velocities. Typical examples include waves on the water surface from wind blowing over the water, and clouds forming a KH pattern of coherent structures. These typical patterns are shown in Fig.4.7.

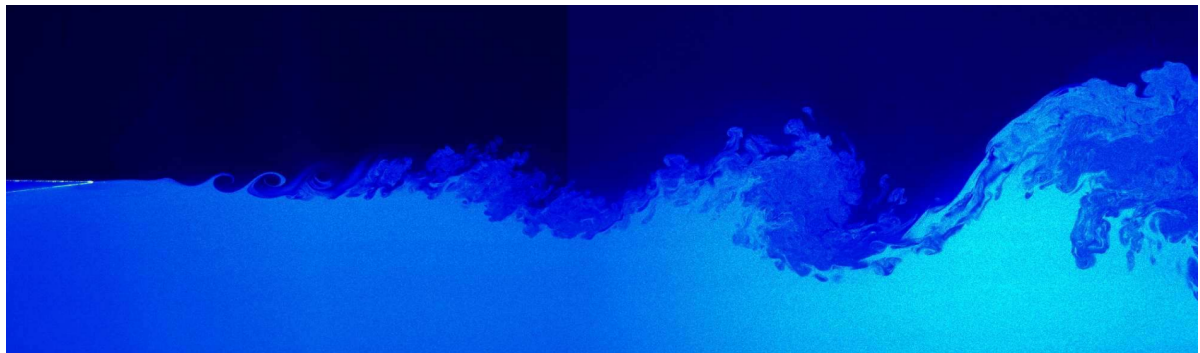


Figure 4.7: Visualization of a turbulent shear mixing layer. Obtained from the work of Parezanovic et al. [107]

The KH-instability is caused by the destabilising effect of shear that overcomes the stabilising effect of stratification [75]. This instability results in significantly improved momentum and heat transfer, it is therefore vital for many engineering applications [75].

### 4.2.1. CFD set-up

This section describes the Geometry and Mesh, CFD Physics settings and CFD quality control. The spatial and temporal discretization schemes and settings were previously discussed in Section 2.2.2

#### Geometry and mesh

This application case uses a quasi two-dimensional geometry and is based on the Mixing Layer validation case from Delville [38]. The NASA-website provides results and documents for validation of RANS turbulence models. For this research the velocity profiles are compared to assess the quality of the CFD simulations. The domain dimensions are given in Table 4.4 and key mesh statistics in Table 4.5.

Table 4.4: Dimensions domain

Parameter	Value (mm)
Length upper-duct (X)	1800
Length lower-duct (X)	1500
Height both ducts (Y)	125
Width both ducts (Z)	2
Splitter plate thickness	0.3

Table 4.5: Mesh statistics

Parameter	Value	Unit
Base size cell	100	mm
Number of cells	53666	[-]
Minimum volume change	0.0158	[-]
Minimum face validity	1	[-]
Maximum skewness angle	82.78	deg.

For this case, a small mesh-study is performed where three different meshes are compared. The results are described in Section C.3 and the mesh that is superior is shown in Fig.4.8.

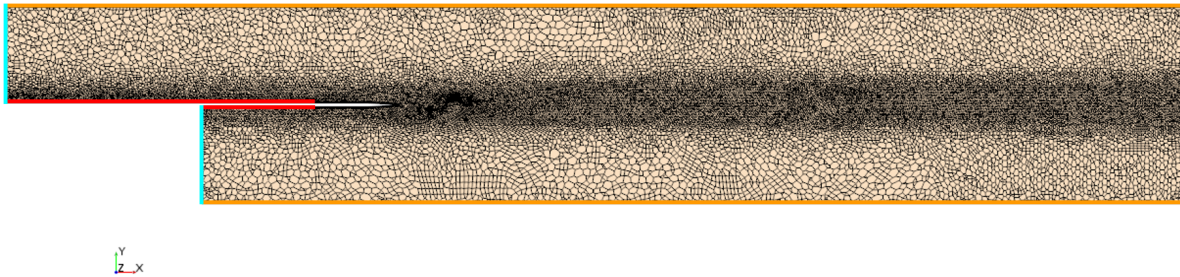


Figure 4.8: Mesh used for the Mixing Layer cases. The colors represents again the different boundary conditions which are described in the text below.

The mesh has a refinement near the centerline of the domain. This is the interface region, between the two flows and where the coherent structures form.

### CFD physics

The same physics models are used for the Mixing Layer cases as for the Von Karman Vortex Street cases, for more information on the physics models see Table C.1. For the Mixing Layer a time-step is used of  $2 \cdot 10^{-5}$  seconds and 6 inner iterations. The combination of boundary conditions (prescribed by Delville [38]) consists of: 1) velocity inlets (turquoise), 2) pressure outlet (green), 3) symmetry planes, 4) adiabatic no-slip walls (red) and, v) adiabatic slip-walls (orange). The two symmetry planes are positioned at  $(X, Y, \pm 1)$ . The operating conditions consist of a reference pressure of 1 Atmosphere, reference temperature of 300 Kelvin and the pressure outlet is at 1 Atmosphere as well, these conditions are valid for all the Mixing Layer simulations.

The design parameter for the Mixing Layer is the velocity of the flow at the inlet of the upper duct. The velocity of the inlet from the lower duct is kept constant at 22.4 m/s. A base-case is defined that uses the same operating conditions to the case prescribed by Delville [38]. The intention was to first validate the base-case, and subsequently derive and compute deviations from this base-case.

Table 4.6: Design space for the Turbulent Shear Mixing Layer application case

No.	Velocity inlet deviation (%)	Velocity inlet upper duct (m/s)	Velocity ratio (-)
1 (base-case)	0	41.54	1.85
2	-5	39.46	1.76
3	-10	37.39	1.67
4	-20	33.23	1.48
5	+10	45.69	2.04
6	+20	49.85	2.23

The CFD quality control for the Mixing Layer simulations are not discussed for the sake of brevity. All the residuals for all cases, converged to a point below  $10^{-4}$ . The CFL condition can be seen in Fig.C.6 and the Pope criterion can be seen in Fig.C.5.

**Results**

The following subsection provide the results of LES simulation number 1. Figure 4.9 shows the pressure and vorticity flow-fields for multiple time-steps.

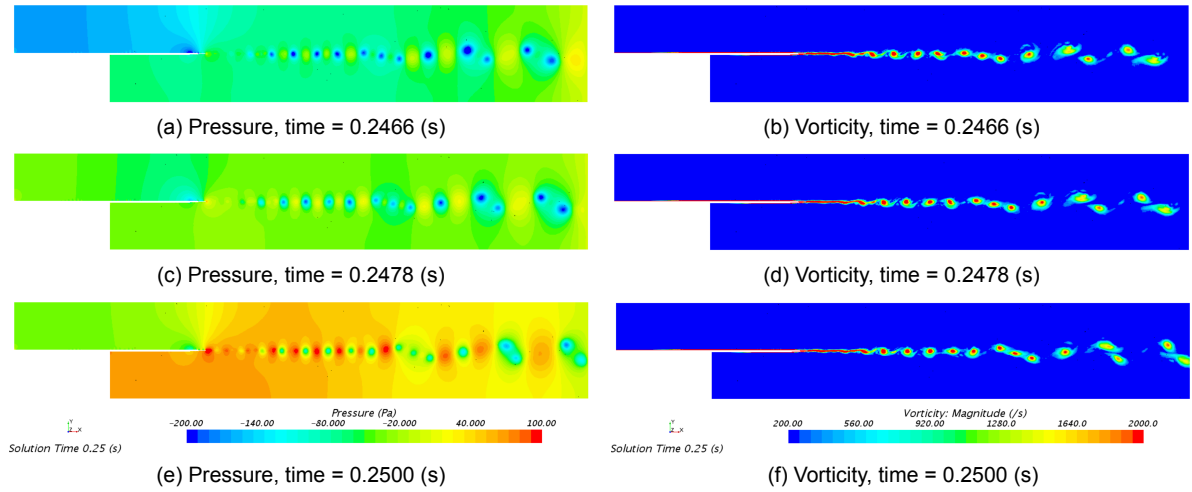


Figure 4.9: Six snapshots of the Mixing Layer base case

For the mixing layer cases the overall quality of the LES simulations are judged by how well the base-case approximates the experimental results from Delville [38]. Only the velocity profiles are compared at different line-probe positions, but all are on the centerline ( $Y=0$ ).

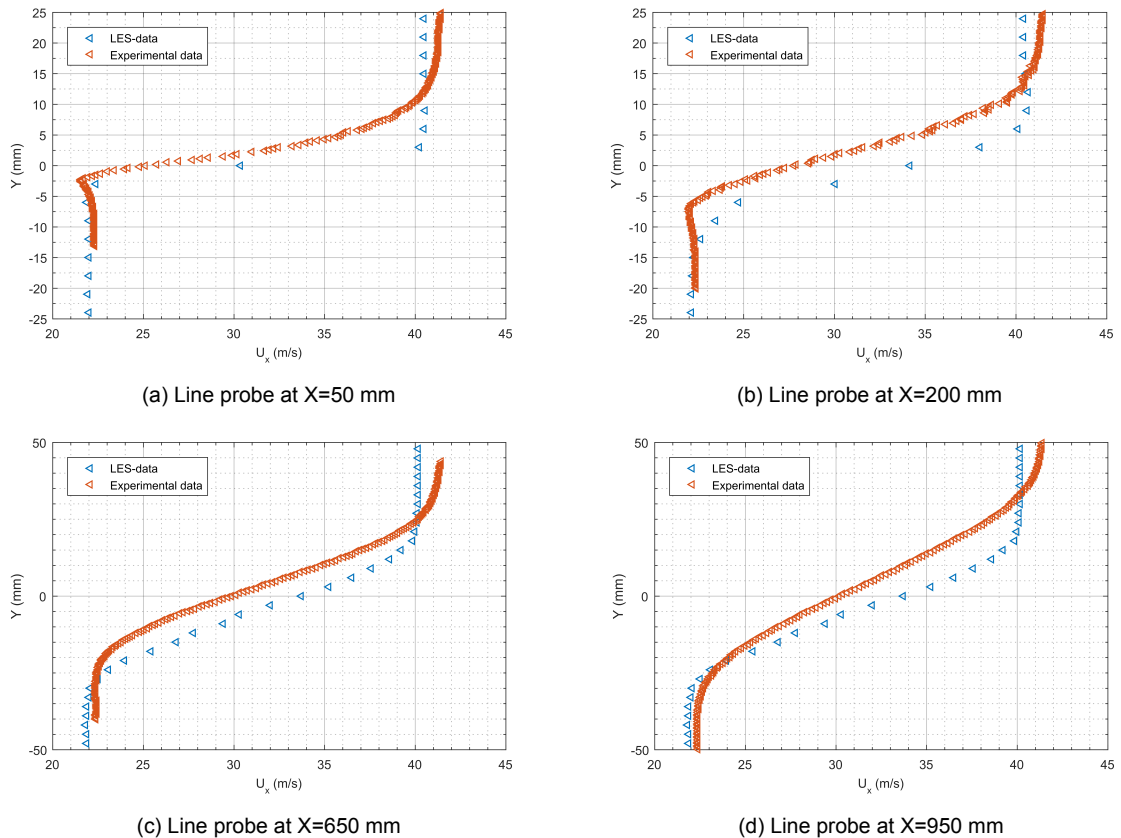


Figure 4.10: Velocity profiles of line probes at different axial positions

### Discussion results

When the velocity profiles of Fig.4.10 are analysed, a difference between the CFD and the experimental results can be seen. The dominant reason for the difference is because of an undesired CFD phenomena. It is observed that the pressure outlet behaves as a reflective wall because it is enforcing a fixed pressure value [108]. A strong indication for this is shown in Fig.4.11 where two probe points monitor the pressure over time. The blue lines represent probe points in the region of interest, thus where the coherent structures form. The red line represents probe points outside this region and where no large Eddies (fluctuations) are present. Therefore, it should have a near constant pressure.

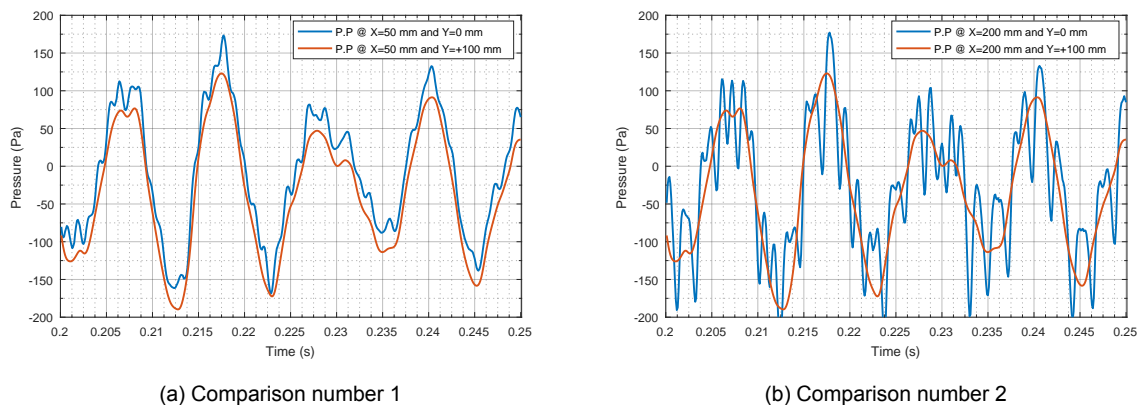


Figure 4.11: Pressure probe comparison of inside and outside the wake

From Fig.4.11 it can be seen that a large pressure fluctuation is present both inside and outside the region of interest. This is a strong indication that these are nonphysical waves. The waves most likely originate from the pressure outlet that eventually forms a feedback loop between the reflective pressure outlet and reflective velocity inlet [108]. The reflective waves can also be observed in Fig.4.9.

This phenomena has been observed before by other Siemens users. However, it seems that small and simple geometries, where the inlet and outlet are opposite to each other, experience this phenomena more forceful than other geometries. Currently, Siemens PLM (the developer of StarCCM+) is working on developing and implementing non-reflective boundary conditions, however it was not available during this research. The following attempts have been made to overcome the reflective boundary conditions:

- Used 3 different combinations (i.e. inlet and outlet) of boundary conditions.
- Used a Smagorinsky turbulence model.
- Implemented an acoustic suppression zone by implementing a sponge layer.
- Solved the cases with different versions of StarCCM+.

These attempts were all unsuccessful and due to time-constraints no further attention was given to this problem. In-house studies, performed in subsequent months, showed the difficulties of implementing an acoustic suppression zone and justified the decision to stop working on this. More importantly, it can be stated from Fig.4.10 that the velocity **trends** are captured, and from Fig.4.9 that the vorticity flow-field exhibits the classic Kelvin-Helmholtz coherent structures.

For more information on reflective boundary conditions see Poinso et al. [108], or acoustic suppression zone (sponge layer) see [32, 89].

### 4.3. Consequences of quasi two-dimensional CFD

Both the domain of the von Karman Vortex Street and the Mixing Layer are quasi 2-D. However, previously stated in Section 2.1.2, turbulence can only truly exist in a 3-D space [103, 106]. Because vortex stretching is a 3-dimensional phenomena and plays an essential role in the energy cascade. In this section the physical consequences of using a quasi 2-D domain is briefly discussed. Quasi 2-D domains are used instead of 3-D domains because of the significantly reduced computational cost of the whole process (e.g. computing, data-extraction, Machine-Learning, prediction).

In Section 2.1.2 the concept of the energy cascade is introduced, that dictates that large eddies interact with other large eddies and subsequently break into smaller eddies. This process is continued until the very small eddies dissipate into heat through viscous forces. Vortex stretching plays an important role in transferring the energy from larger eddies to smaller Eddies but is strictly a 3-D phenomena [103, 106]. When a turbulent flow is confined to a 2-D domain, the energy cascade concept still holds. Remarkably however, for (quasi) 2-D turbulence the direction of the energy cascade is reversed and thus energy is transferred to the larger scales [44, 106]. This inverse cascade theory was first postulated by Kraichnan [73] in 1967 and became a seminal contribution to the theory of turbulence [44]. In contrast to 3-D turbulence (where macro-sized eddies interact and break up into smaller eddies) 2-D turbulence has shown a form of "vortex cannibalisation" (where smaller eddies merge into larger eddies) [121]. For the mathematical working principles behind this, the reader is directed to the work of Kraichnan [73, 74].

2-D turbulence and the inverse cascade have multiple consequences that make it quite different to 3-D turbulence. Whereas, a 3-D flow tends to be extremely chaotic, unpredictable and to a degree random [103], a 2-D turbulent flow strongly tends to self-organise and form long-lived large scale coherent structures [106]. This is an important to realise when dealing with (quasi) 2-D domains, that it basically results in an unrealistic smooth flow-field compared to its 3-D counterpart.

### 4.4. Conclusions application cases

In Chapter 4 two canonical fluid dynamic cases are described. The chosen Von Karman Vortex Street case and Mixing Layer case are: 1) relatively straightforward, 2) do not require large domains with dense meshes and 3) abundant validation data is available. Furthermore, for computational efficiency only quasi 2-D domains are used in this work. As a consequence of 2-D turbulence and an inverse energy cascade, unrealistic smooth flow-field are obtained compared to its 3-D equivalent.

The physics for the Von Karman Vortex Street is governed by the alternately shedding of vortices that is mathematically described by the Strouhal-Reynolds relation (see Eq.4.2). The design space consists of 13 operating points between a Reynolds number range of  $2500 < Re < 6000$ . In this range the vortex shedding frequency increases linearly. A small deviation, 5 – 10%, is present between the simulated and experimental vortex shedding frequencies. The expected cause is due to the coarse mesh, as the boundary layers near the cylinder are not adequately resolved.

The physics from the second test case, the shear Mixing Layer, is governed by the Kelvin-Helmholtz instability. This instability is caused by the destabilising effect of shear that overcomes the stabilising effect of stratification. The design space consist of 6 operating points that account for velocity ratios between 1.48 – 2.23. Regarding the results, significant differences with experimental data are present due to an undesired CFD phenomena. It is observed that the pressure outlet acts as a reflective wall and causes large pressure fluctuations propagating through the domain. Nevertheless, the results are deemed sufficient as important velocity profile trends are captured. Furthermore, Kelvin-Helmholtz coherent structures can be observed for all the operating points.

# 5

## POD-Interpolation

At this half-way point a compact summary is appropriate. Chapter 2 revolves around fluid dynamics, its complexities and how it can numerically be solved with LES simulations. This is succeeded by the literature study, Chapter 3, that provided 19 potential holding ML-algorithms for the prediction of flow-fields. With the help of 4 criteria the algorithms are narrowed down to the first implemented algorithm, that is the POD-Interpolation algorithm. This is a simplification from the CPOD-Kriging [88] algorithm that reportedly provided very high speed-up potential, additional learning benefit and is relatively straightforward. Chapter 4 described the two application cases that are used to test the efficacy of the POD-Interpolation algorithm. Chapter 5 intends to:

1. Describe the methodology for the POD-Interpolation approach.
2. Provide the results for the Von Karman Vortex Street and Mixing Layer.
3. Give a verdict on the CFD and POD-Interpolation combination.

The outline of Chapter 5 is as follows. Section 5.1 describes the methodology and Section 5.2 provides the results of the von Karman Vortex street. This is succeeded by Section 5.3 where the results of the turbulent shear mixing layer are described. Section 5.4 provide a discussion on the results and Section 5.5 the verdict on the POD-Interpolation algorithm.

### **Simplification of the CPOD-Kriging algorithm**

The simplification to the POD-Interpolation algorithm is possible under 2 conditions. First of all, the CPOD only differs from a POD algorithm by its ability to cope with data from different meshes due to geometrical parameter variations. However, as the design space of the Von Karman Vortex Street (Table 4.3) and the Mixing Layer (Table 4.6) shows, only flow parameters are changed. Therefore, there is no incentive to use the more complex CPOD algorithm.

Secondly, instead of using an advanced interpolation method (i.e. Kriging) it is decided to use an ordinary linear interpolation method. This is justified by the following 2 points:

1. The Kriging algorithm in the methodology proposed by Mak et al. [88], is implemented as an 2-D line interpolation method instead of a 3-D Response Surface Model method.
2. A non-linear (Kriging) and a linear interpolation method are identical when interpolating between two data-points/cases.

By using the standard POD algorithm and an ordinary linear interpolation the implementation is less complex and quicker, without comprising the quality. The more advanced methods can be implemented when such a need arises. Due to the similarities between the algorithms it can be stated that by determining the potential of the POD-Interpolation algorithm, also indirectly the potential of the CPOD-Kriging (and to a large degree the KSPOD-Kriging) is determined.



## 5.1. Methodology

In this section the methodology is described approach using Fig.5.1. The starting point in this framework is the design space, for the von Karman Street and Mixing Layer these are respectively given in Table 4.3 and Table 4.6. The operating points can both be used for validation and training of the ML-algorithm. The second step is computation of the training points with a LES solver and thus creating the required training data. The training data is represented in Fig.5.1 as the instantaneous flow-field. Unlike the CPOD-Kriging methodology [88], a sensitivity analysis and a data-clustering step is not incorporated. For large design spaces with many different design parameters it is recommended to include these steps for better results [28, 88]. For this work however, these additional steps are deemed unnecessary because:

- The change in output, to a change in a flow parameters (input) is relatively well-understood.
- No large variations in the flow-fields exists over the design space.

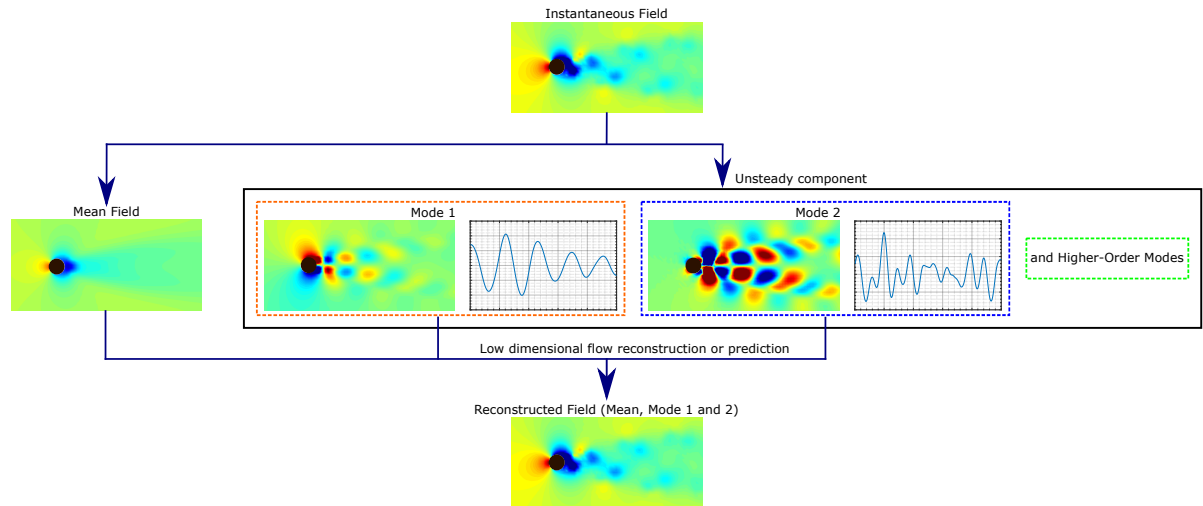


Figure 5.1: The essence of the POD-Interpolation algorithm

Step 3 is characterised by two distinct aspects: 1) the extraction of the key dynamical features and 2) prediction of flow-fields by modulating key dynamical features. Extraction of key features is performed by decomposing the flow-field data into spatial modes and temporal coefficients. This step can be seen in Fig.5.1, where the decomposition of the instantaneous flow-field results in a mean-field and unsteady components. The most important features are retained by truncating the less relevant features. This results in a reduced set of key spatial modes and temporal coefficients. Prediction of the dynamics for an objective point is performed by interpolating between temporal coefficients from training points [88]. With respect to Fig.5.1, the blue temporal signals are interpolated to obtain new temporal coefficients from the objective points. Important to note is that only the temporal coefficients and mean fields are modulated to predict flow-fields at objective points. The spatial modes are neglected, Mak et al. justifies this by assuming that the spatial modes do not change over the design space [88] and therefore do not require any attention. This assumption is extensively discussed in Section 5.4. Prediction is performed with a linear interpolation method that is defined as [24]

$$c = c_0 + (q - q_0) \frac{\Delta c}{\Delta q}. \quad (5.1)$$

Lastly, step 4 is the reconstruction of the flow-fields at objective points. Reconstruction is achieved by the addition of the modulated mean-field with the multiplication between the newly obtained temporal coefficients and spatial modes from a training case. The methodology is summarised as:

1. Define the design space with the definition of the training points and objective points.
2. Compute the training points with an LES solver.
3. (a) Decompose the flow-field data into spatial modes and temporal coefficients  
(b) Obtain a dimension reduction by truncating to a reduced set of modes.  
(c) Predict the new dynamics by interpolating the temporal coefficients of training points.
4. Reconstruct the flow-field for objective points.



## 5.2. Von Karman Vortex Street

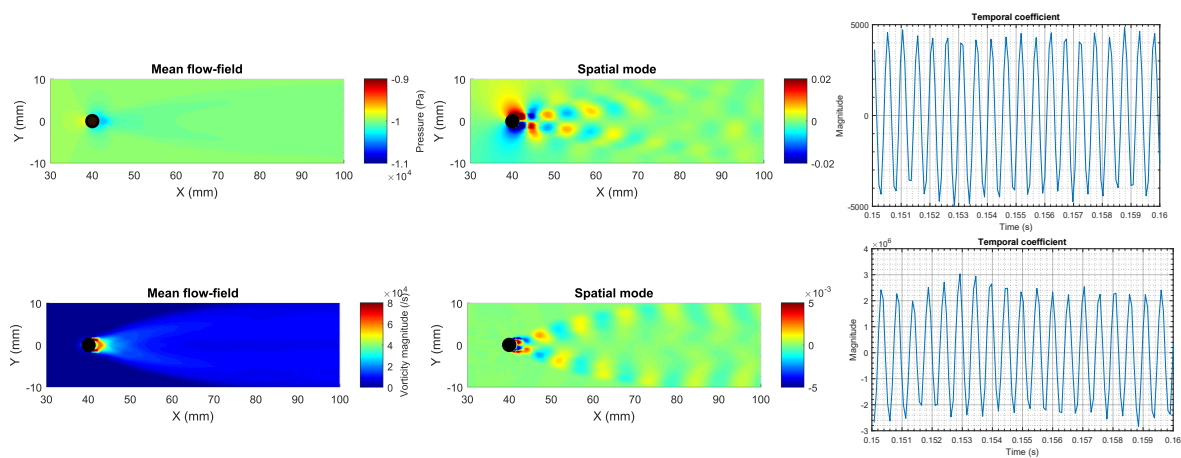
In this section the results are provided for the vortex street that is done in the order of the POD-Interpolation methodology. Step 1 and step 2 are discussed in Chapter 4.

### 5.2.1. Flow-field decomposition

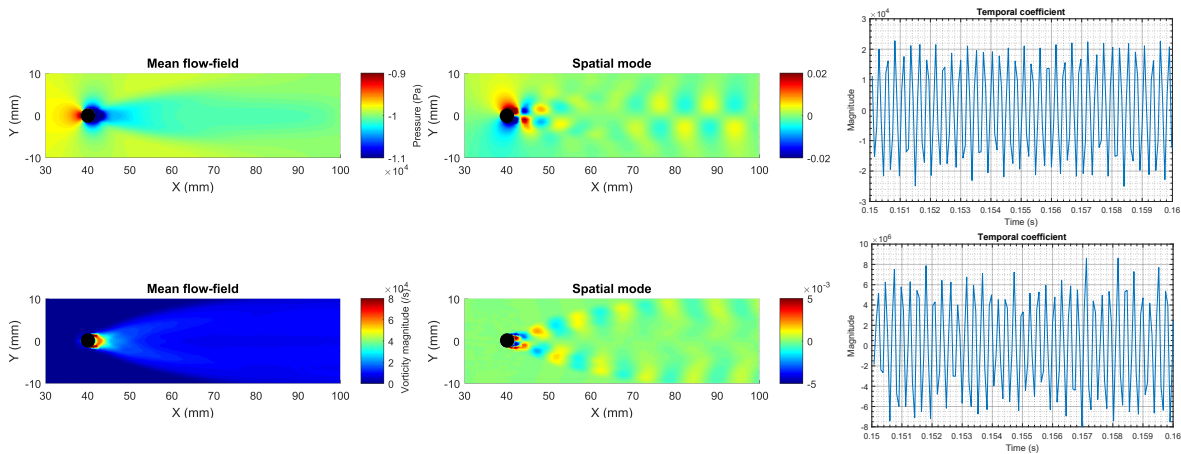
After the computation of the training points the data is loaded into MatLab and the POD algorithm decomposes the data. In this section the dominant spatial and temporal modes are shown and compared for training points 1 and 13, and objective point 8. To see the details of these design points see Table 4.3 or Table 5.1. All spatial and temporal modes for all design points are found on the GitHub page [5].

#### Mean mode + Mode 1

In this paragraph a comparison is drawn between training points 1 and 13, by showing the mean flow-field and the first mode. Mode 1 is the most dominant mode in the flow-field and has the highest energy content [86]. The upper row of each subfigure represents the pressure data whereas the lower row represents vorticity data.



(a) Training point number 1, mass flow deviation = 0%



(b) Training point number 13, mass flow deviation = 100%

Figure 5.2: The mean mode/flow-field and the first spatial mode is given with its temporal coefficient are given for 2 different operating points. The upper row represents the pressure data, whereas the lower row represents vorticity data.

The following things are observed from Fig.5.2 but generally holds true for all the design points points.

- Higher velocities results in an increase in the negative-pressure in the wake, which is clearly visible from the mean pressure flow-fields. Near the stagnation point the pressure rises.
- Differences for the vorticity mean flow-fields are less visible but present. Aside from the increased magnitude, an incline in the angle of the higher vorticity region is observed over the design space.

- The 1<sup>st</sup> spatial mode for both operating points are to a large degree similar. This generally holds true for the 1<sup>st</sup>, 2<sup>nd</sup> and 3<sup>rd</sup> spatial mode as they are to a large degree similar for all training points.
- The amplitudes of the temporal coefficients increase over the design space. For example, the coefficients from Fig.5.2 (b) are a magnitude of order higher than from (a).
- The temporal coefficients for operating point 13 are more aperiodic than those from operating point 1. The trend is observed that temporal coefficients become more aperiodic over the design space, thus with the increase in mass flow and Reynolds number.

Generally speaking, the first 3 dominant modes (i.e. spatial and temporal) are to varying degrees similar over the entire design space. From mode 3 on, dissimilarities over the design space are becoming more evident. A good example of this is shown in Fig.5.3, where the 9<sup>th</sup> mode is depicted. Lastly, the aperiodicity of temporal coefficients is a reoccurring topic from this point on. The aperiodicities of the temporal coefficients increases with the mass flow deviation, as shown in Fig.5.2. But, also with higher order modes, which is partially shown in Fig.5.3. This allows for the generalisation: the higher the mode, the higher the degree of aperiodicity of the temporal coefficients. This is a known drawback of the POD algorithm [24, 124], each coefficient accounts for multiple frequencies (Section 3.4.1).

### Mode 9

Mode 9 is showcased because it is a mode that exhibits two features that are representative for the majority of the spatial and temporal modes. Figure 5.3 shows mode 9 for training point number 1 and 13 and where each row represents the derived modes from pressure data. The following things are observed from Fig.5.3 but generally holds true for all the training points.

- The interpretability of the spatial modes is diminishing. For example, it is (very) difficult to connect the upper spatial mode from Fig.5.3(b) to a physical phenomena. Unlike the spatial modes in Fig.5.2 that shows clear vortex shedding patterns.
- The temporal coefficients are strongly aperiodic.

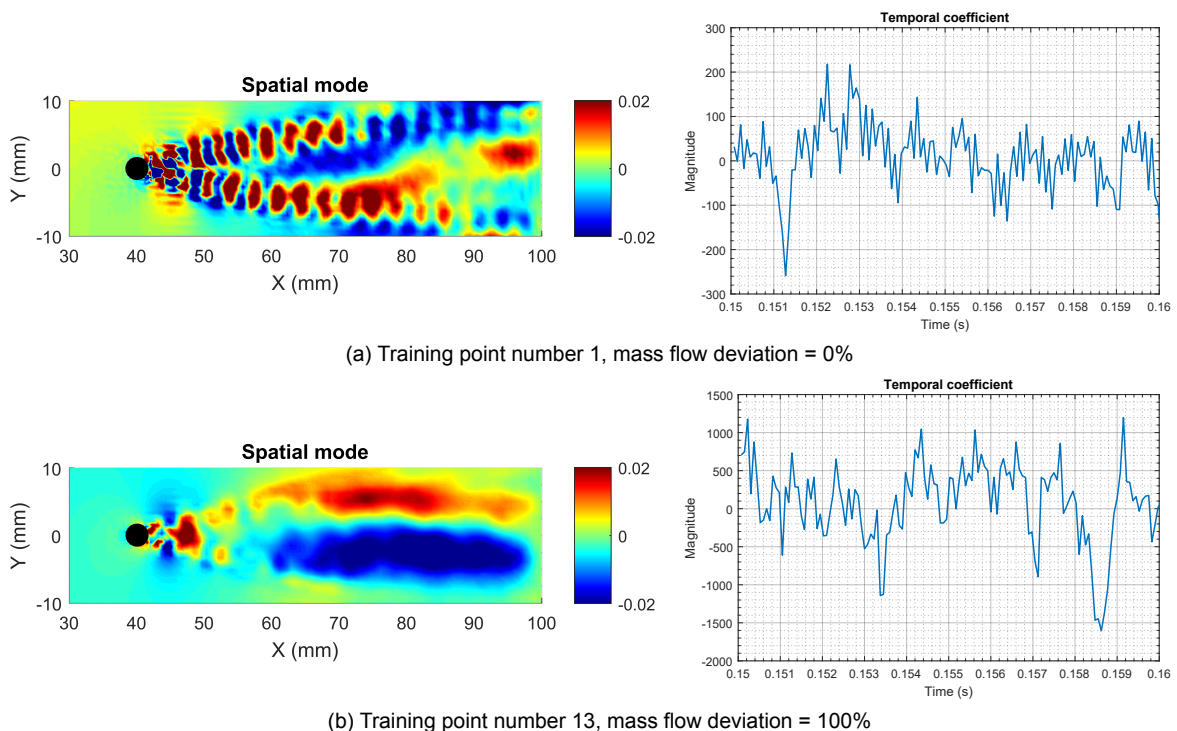


Figure 5.3: The 9<sup>th</sup> mode for operating point 8 and 13

### 5.2.2. Dimension reduction

Mode truncation is relatively uncomplex for the POD-Interpolation approach. As discussed in Section 3.4.1, the POD algorithm provides a ranking based on the variance content of each mode. This ranking and the energy content enable the possibility to truncate the modes with a variance criteria (Eq.3.3). A typical variance criteria dictates that the truncated set of modes need to capture  $> 99\%$  of variance. Two variance plots are shown in Fig.5.4. Vorticity flow-fields contains high gradients, this is the expected cause for the shallower curve with respect to Fig.5.4 (a).

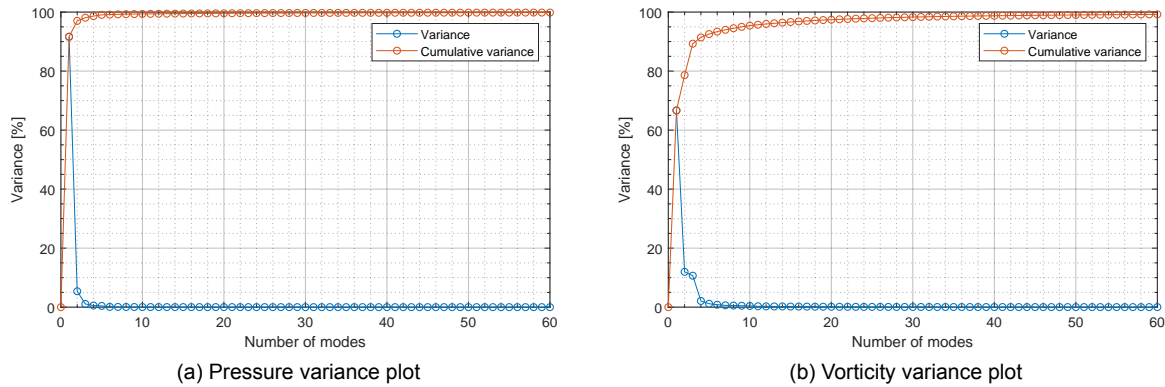


Figure 5.4: Variance plots for training point 13

Based on the variance plots 50 modes are used for all POD-Interpolation computations unless specified otherwise. 50 Modes provide: a 1) significant dimension reduction (from +1000 modes to 50), 2) exceeded the 99% variance threshold, and 3) produced visually identical flow-fields to the CFD original flow-fields. Theoretically when computational efficiency is the main priority, more modes can be truncated. In practice, the computational time of 50 modes for subsequent steps (i.e. prediction and reconstruction) are easily manageable and thereby removing the need for any further truncation.

The accuracy of 50 modes can be observed in Fig.5.5, this figure shows the time-averaged MRE (Eq.D.1) for training point 1. Important to note is that the 50 modes reconstructed the original flow-field from training point 1, this is not a predicted flow-field. The topic of error quantification is extensively discussed in Section D.2. Converging behaviour is common for a flow-field reconstruction of a training point. More modes result in a lower mean-residual error (MRE) and therefore higher accuracy. In Section 5.4.3 and Fig.5.21 it is shown that the converging behaviour is not guaranteed for predicted flow-fields.

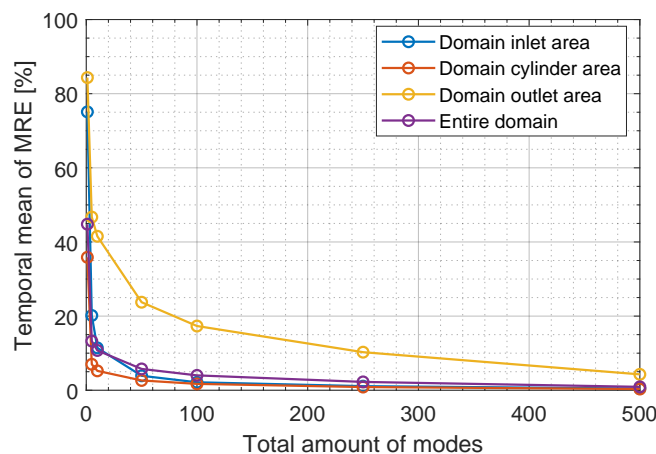


Figure 5.5: Time averaged MRE that compares the reconstruction quality to the amount of modes used for reconstruction

Seen in Fig.5.5 is that for 50 modes, the MRE is well below 10 % for the cylinder area and below 25% for the other 3 domains. The distinction between 4 different domains is done to showcase where the POD modes are working the most effectively. Lower order modes work in the area of the highest fluctuations, thus in the cylinder area. Whereas the higher order modes approximate the smaller Eddies near the outlet.

On a separate note, in literature there is debate on the topic of truncation. Taira et al. [145] states that it is not always clear how many POD modes should be kept and that many different truncation criteria exists. Taira et al. states that by only looking at the energy content of the POD modes, low-energetic modes are being truncated but these can be vital for the overall dynamics of the flow-field. Brunton et al.[24] describes the truncating method to pre-determined amount of variance as crude and suggest the use of the optimal hard threshold as proposed by Gavish et al. [51]. This method was briefly implemented during this project, it resulted in a significantly higher number of modes (+500 modes) and did not produce visually better results compared to the original CFD flow-fields. Therefore, the typical method of truncating to a pre-determined amount of variance is used.

### 5.2.3. Prediction and reconstruction

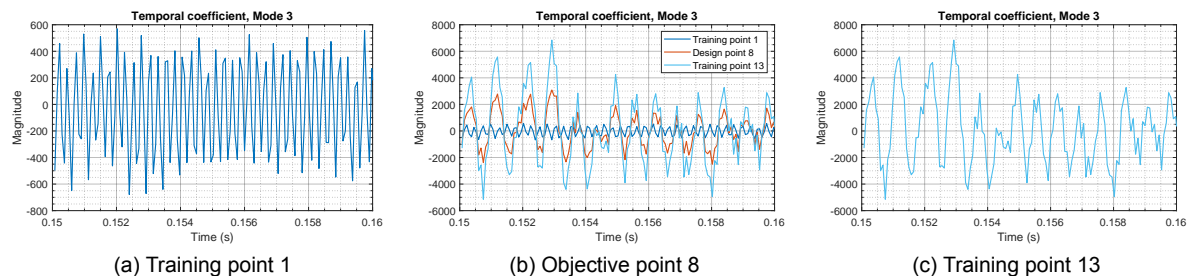
Prediction of fluctuating flow-fields is performed by linear interpolating (Eq.5.1) the temporal coefficients from the training points. Mak et al. [88] assumes that all the spatial modes are identical over the design space and therefore does not require any attention. Table 5.1 provides an overview of the objective points. The flow-field of the objective points are determined by interpolating the temporal coefficients from the training cases.

Table 5.1: Design points with their respective training points

No.	Training No.	Deviation (%)	Mass flow (g/s)	Vortex shedding freq. (Hz)
2	1 & 3	5	1.3125	1883
3	1 & 5	10	1.3750	1970
4	1 & 6	15	1.4375	2060
5	1 & 7	20	1.5000	2147
6	1 & 9	30	1.6250	2324
7	1 & 11	40	1.7500	2500
8	1 & 13	50	2675	

Important to note is that each objective point, deviation-wise, is always at the center of the 2 used training points. This is not the same as linear interpolating between training point 1 and 13 to determine the objective points. Each design point has case-specific vorticity and flow-features. Therefore, interpolating between point 1 & 3 not the same as interpolating between point 1 & 13 to reach objective point 2.

In Fig.5.5 the temporal signals from mode 3 and 11 are given for the training points 1 and 13, and objective point 8.



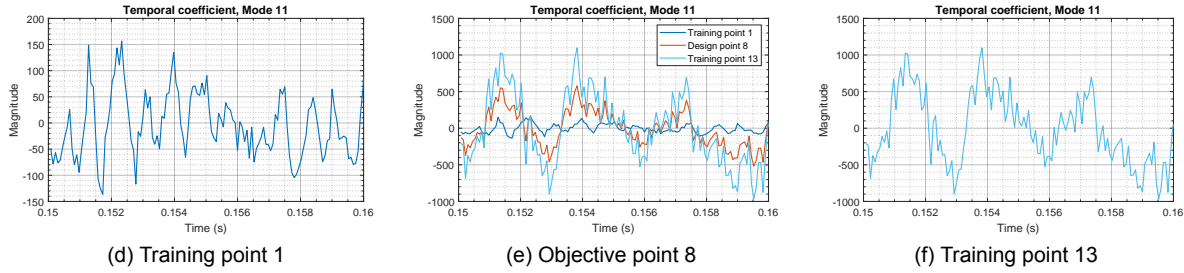


Figure 5.5: Temporal coefficients for three different operating points. The temporal coefficients from design point 8 (red) is obtained with linear interpolation of the two blue training point signals.

The following observations are made based on Fig.5.5. These observations are key and will be further discussed in Section 5.4.

- Phase differences between the temporal coefficients from the training cases exists.
- Frequency differences between the temporal coefficients from the training cases exists.
- The coefficients from the training cases do not exhibit the same aperiodic tendencies.
- The amplitude for the signals from training point 13 are significantly higher than those from training point 1
- The interpolated coefficient signal largely copies, with all its aperiodicities, the stronger signal from training point 13

The prediction of mean flow-fields is something that is not specifically addressed in the methodology of Mak et al. [88]. Since the method relies heavily on interpolation methods, it is likely that they also used it for the prediction of the mean flow-fields. Mean flow-fields are per definition steady-state, and therefore the interpolation (Eq.5.1) is performed between the spatial mean flow-fields from the training case. This diverges from the prediction of fluctuating flow-fields, where the temporal component of the training cases are interpolated. The results for the mean flow-fields interpolation are given in Section 5.2.

### Reconstruction

Reconstruction is basically the inverse of the flow-field decomposition step as described in Section 5.2.1. In the reconstruction step the 3 governing components, albeit modulated components, are reconstructed to obtain the flow-field at an objective point. The governing equation (Eq.3.2) is reiterated below, where  $\vec{u}$  denotes any flow property.

$$\vec{u}(\vec{x}, t) - \bar{\vec{u}}(\vec{x}) = \sum_m^{\infty} a_m(t) \phi_m(\vec{x}) \quad (5.2)$$

The newly obtained temporal coefficients (Fig.5.5 (b) and (e)) are imposed on the spatial modes of a training case 1, seen for example in Section 5.2.1, and added to modulated mean flow-fields.

### 5.2.4. Results

In the following subsection the results are shown and described for the Von Karman Vortex Street. The goal is to obtain accurate flow-fields for new design points for less computational time compared to a LES simulation. As a reminder, the results of LES simulations of a von Karman Vortex street are described and shown in Section 4.1.1. The results are divided in three paragraphs: the prediction of mean flow-fields, the prediction of fluctuating flow-fields and error quantification of the results. This division between the mean and the fluctuating flow-field comes naturally for anything involved with an POD algorithm, because it automatically subtracts the mean (see Eq. 5.2). To nicely highlights the stationary and transient component of each flow-field, this division is kept intact.

#### Prediction of mean flow-fields

The results for the prediction of mean flow-fields are shown in Fig.5.6, where in (a,c) the mean fields from the LES are shown and in (b,d) the interpolated mean field. Differences between the LES and interpolated flow-fields are hardly visible. The MRE, that quantifies the difference between the LES and interpolated field, never exceed the maximum value of 0.07%.



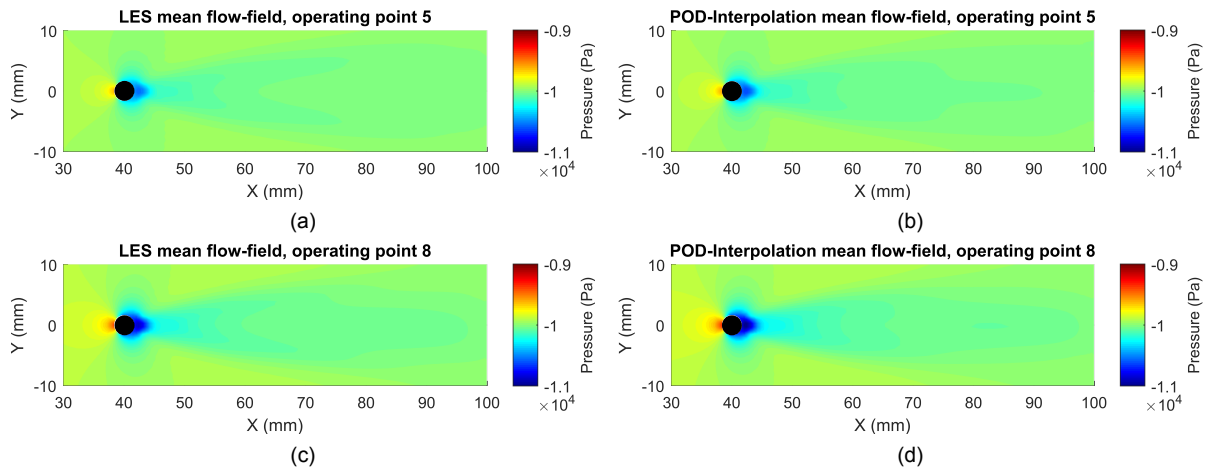


Figure 5.6: Mean pressure flow-fields for objective points 5 and 8

For this canonical test case the prediction results in negligible MRE's. However, it is doubtful that linear interpolation would work for complex cases with multiple design parameters.

### Predictions of fluctuating flow-fields

The results for this paragraph are provided as follows, first a comparison is drawn between POD-Interpolated flow-fields and its LES counterpart. Subsequently, figures of interest are shown such as pressure plots and pressure spectra. Transient flow-fields are difficult to show and compare on a paper-format, therefore GIFs are provided on the GitHub[5] page. The accuracy of the predicted flow-fields can be better analysed by using the GIFs, than looking at distinct snapshots of the flow-field. Regardless, in Fig.5.7 the predicted flow-field is compared to the CFD flow-field. The predicted flow-field from objective point 8 is achieved by interpolating the temporal coefficients from training points 1 and 13, that are imposed on the spatial modes from training point 1. Chosen are 3 time-steps of the predicted flow-field that shows relative good resemblance to the LES flow-field.

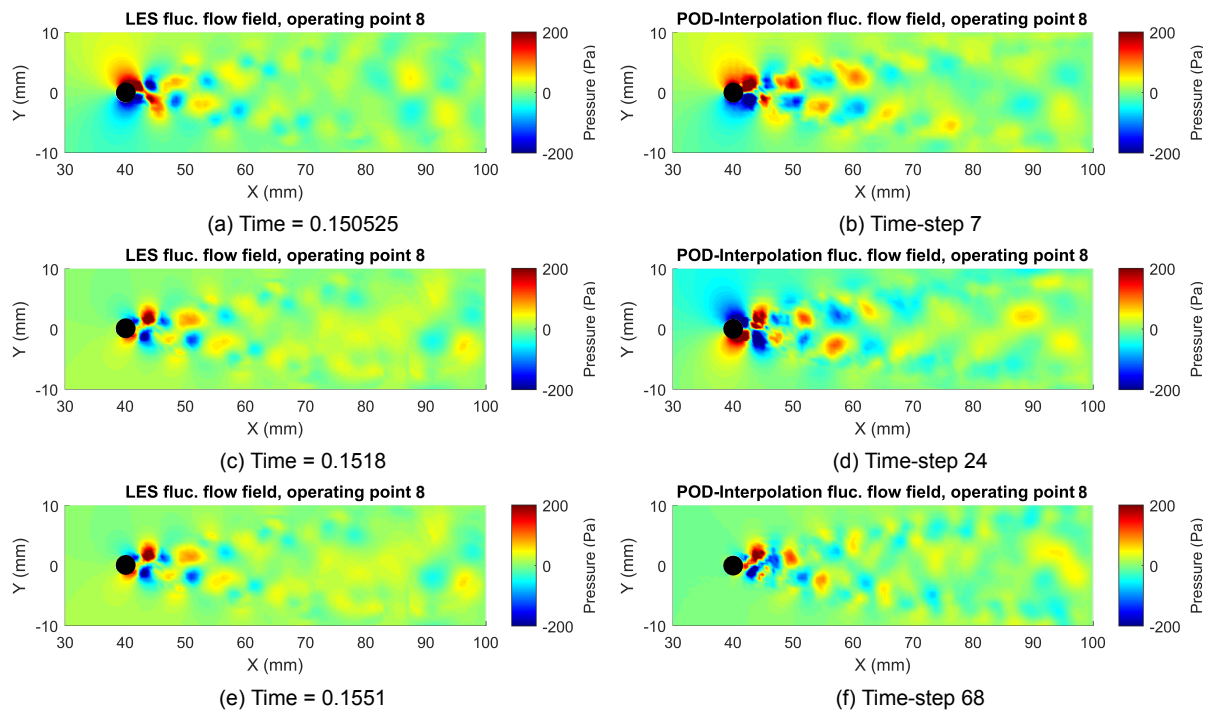


Figure 5.7: Fluctuating flow-field comparison for objective point 8

From Fig.5.7 multiple things are observed:

- The vortex shedding fluctuations are captured with reasonable accuracy.
- The LES fluctuations (coloured red and blue) that indicate large Eddies are circular and have relative well-defined boundaries. Whereas, the POD fluctuations tend to show more irregular patterns.
- A form of phase difference can be seen when comparing Fig.5.7 (c) and (d), where the first 2 fluctuations surrounding the cylinder are to a certain degree out-of-phase.
- The wake near the cylinder tends to be better predicted than the downstream wake.

These observations are further discussed in Section 5.4. Below are figures that depict the pressure for single probe points (P.P). The location of the probe points are shown in Appendix Fig.D.1. All P.P are positioned on the centerline ( $Y=0$ ) at different positions on the X-Axis. The P.P at 43 millimetres is right behind the cylinder and the P.P at 90 millimetres is near the outlet. These points monitor the pressure for objective point 8 (Table 4.3).

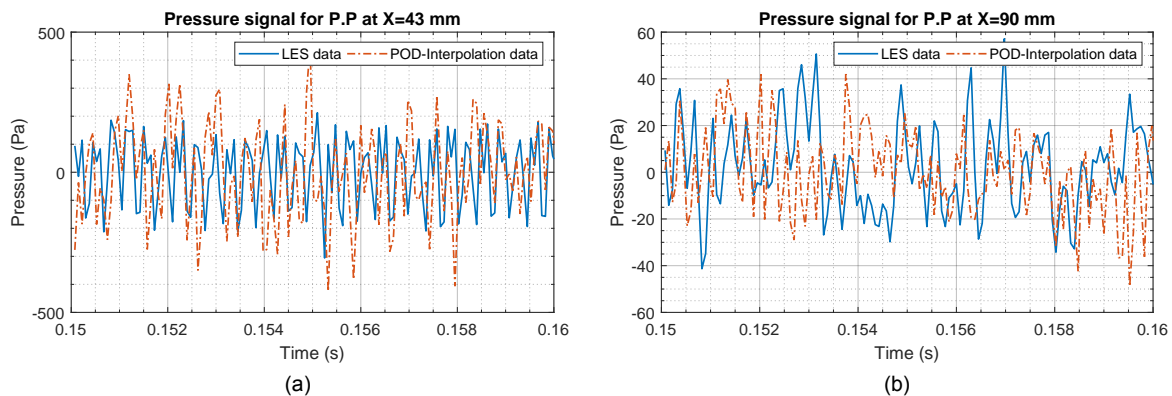


Figure 5.8: Pressure probe signals at two different position for objective point 8

The pressure probe signals depicted in Fig.5.8, show less promising results. By zooming in on certain points of interest it can be seen that the POD-Interpolation pressure signals are poorly predicted, significant discrepancies are present. These pressure signals are transformed into the frequency domain with a Fast-Fourier Transform (FFT) and the results are shown in Fig.5.9. The pressure spectra depicts the fundamental vortex shedding frequency (large blue peak) of objective point 8.

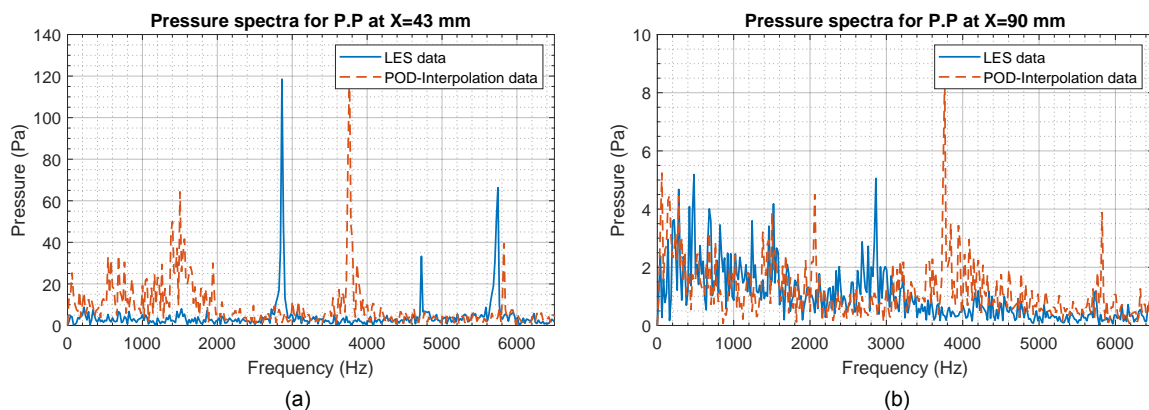


Figure 5.9: Pressure probe spectra at two different positions for objective point 8

Figure 5.9 is a descriptive figure for the POD-Interpolation method in general. The fundamental vortex shedding frequency of 2863 Hz is (very) poorly predicted, as can be seen in Fig.5.9 (a). Interestingly, two artificial peaks are predicted at 1941 Hz and 3786 Hz, which are the fundamental frequencies of training case 1 and 13. This artificial phenomena is even better captured in Fig.5.10.

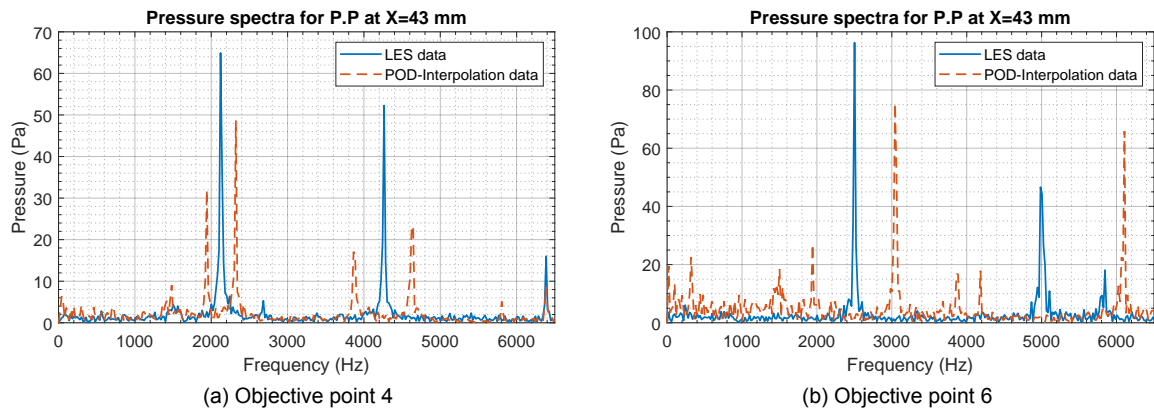


Figure 5.10: Both (a) and (b) show two red large artificial peaks at the fundamental frequencies of their respective training cases.

To predict the dynamics (e.g. vortex shedding behaviour) of a flow-field, the temporal coefficients from training coefficients are interpolated. This interpolation however results in copying the distinct features (Fig.5.10) of the training cases, instead of the prediction of the correct vortex shedding frequency.

### Error quantification of the results

In this paragraph the error between the prediction and the "exact" LES flow-fields are quantified. In Fig.5.7, six snapshots are presented that show decent resemblances to each other. Whereas, the pressure signal figures (Fig.5.8) show relative large differences, the predicted pressure signal poorly approximates the LES pressure signal. Moreover, Fig.5.9 and Fig.5.10 depict the inability of the POD-Interpolation approach to predict one of the key features of the Von Karman Vortex Street, the fundamental vortex shedding behaviour.

In this paragraph all the previous seen results and their accuracy are quantified into two types of errors, the MRE (Eq.D.1) and RMS (Eq.D.2) error. The following computational steps are implemented on the flow-field data, before any error quantification is performed.

1. **Use absolute values.** To mitigate any phase difference between the simulated LES flow-field and predicted POD-Interpolation flow-field.
2. **Transform into the frequency domain.** To highlight, compare and quantify the error between the flow-fields at frequencies of interest

To identify areas of poor prediction an RMS - spatial field is analysed, for example shown in Fig.5.11 for objective point 8. This figure shows that the POD-Interpolation algorithm struggles with the areas near the stagnation point, angle of separation and the wake. Relative high error in these areas are expected since these are governed by highly complex (nonlinear) flow physics. A linear decomposition method such as the POD [24] struggles to decompose this, let alone predicting it.

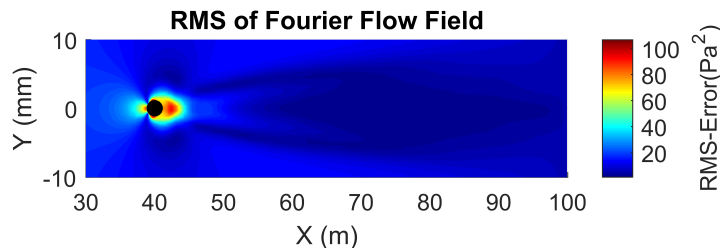


Figure 5.11: RMS error for operating point 8



The more telling figure is Fig.5.12 where the MRE is depicted against the input variable deviation as defined in Table 4.3. The MRE provides a scalar value for the difference between the LES and POD-Interpolated absolute Fourier flow-field, at the fundamental frequency.

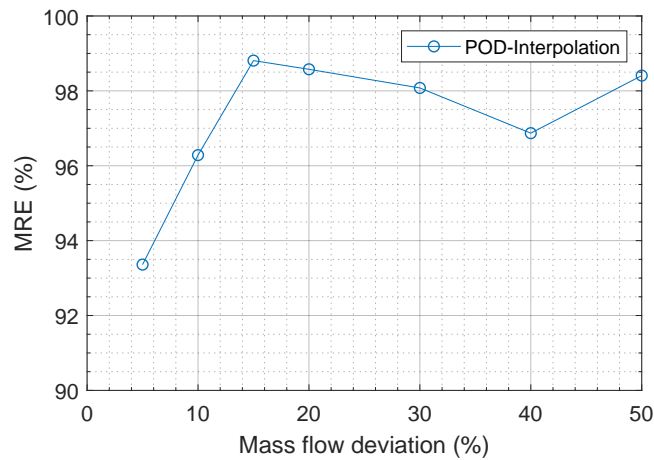


Figure 5.12: Mean Residual Error for the POD-Interpolation approach

Figure 5.12 is a critical figure for the POD-Interpolation approach. For a predicted flow-field with a small mass flow variation, it directly results in a very high MRE that exceeds 90%. Figure 5.10 already indicated the inability of the POD-Interpolation algorithm to predict the fundamental vortex shedding frequency. This is confirmed by Fig.5.12, with the very high MRE for even a small variation in the design parameter. This figure is further discussed in Section 5.4 and throughout the remainder of this document.

### 5.3. Turbulent shear mixing layer

In the following section the results for the turbulent shear mixing layer are discussed, the results are provided in the same order as Section 5.2 but written compactly. Based on the results from the previous section, especially shown in Fig.5.12, it is decided to limit the input variable deviation for the Mixing layer to a smaller deviation range. Table 5.2 provides an overview of the design points.

Table 5.2: Objective points with their respective training points

No.	Training No.	Velocity inlet deviation (%)	Velocity ratio (-)
1		0	1.85
2	1 & 3	-5	1.76
3	1 & 4	-10	1.67
4		-20	1.48
5	1 & 6	+10	2.04
6		+20	2.23

### 5.3.1. Flow-field decomposition

In Section 4.2.1 a discussion on the Mixing Layer LES simulations is given, in which information is provided on reflective boundary conditions. Shown in Fig.4.11 are large pressure fluctuations/waves that most likely stem from the reflective pressure outlet. The POD algorithm struggles with large travelling waves throughout the domain [24, 145], since real-valued POD modes cannot represent a wave with a single mode. Taira et al. [145] states that travelling structures are represented by a set of stationary POD modes, that are nearly similar but appear shifted in the advection direction. An example of this can be seen in Fig.5.13 where the first 2 modes are nearly identical but appear shifted in the advection direction.

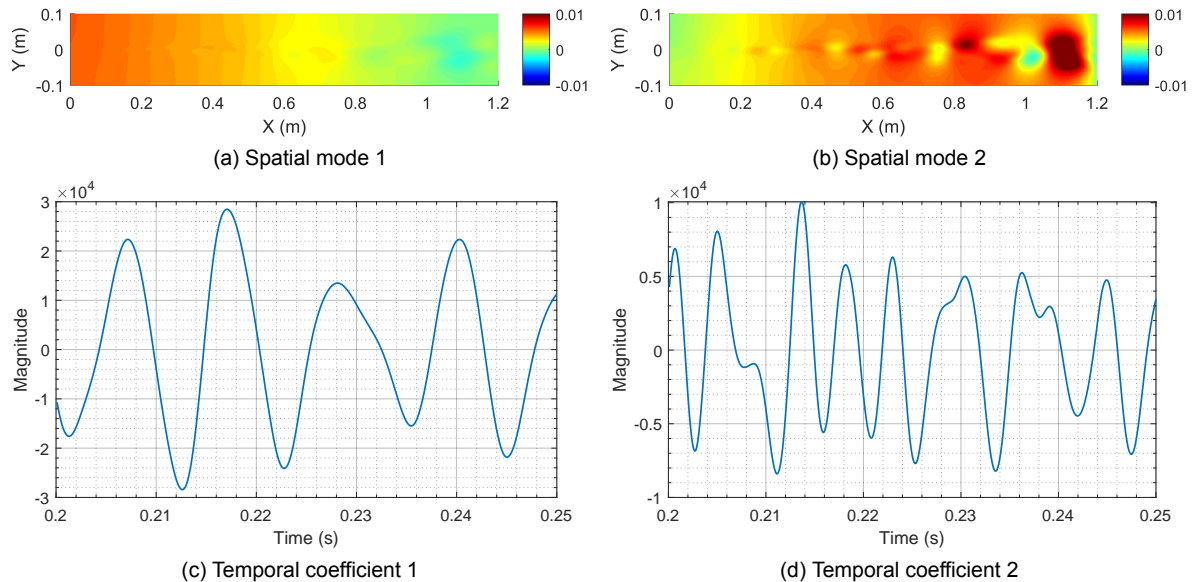
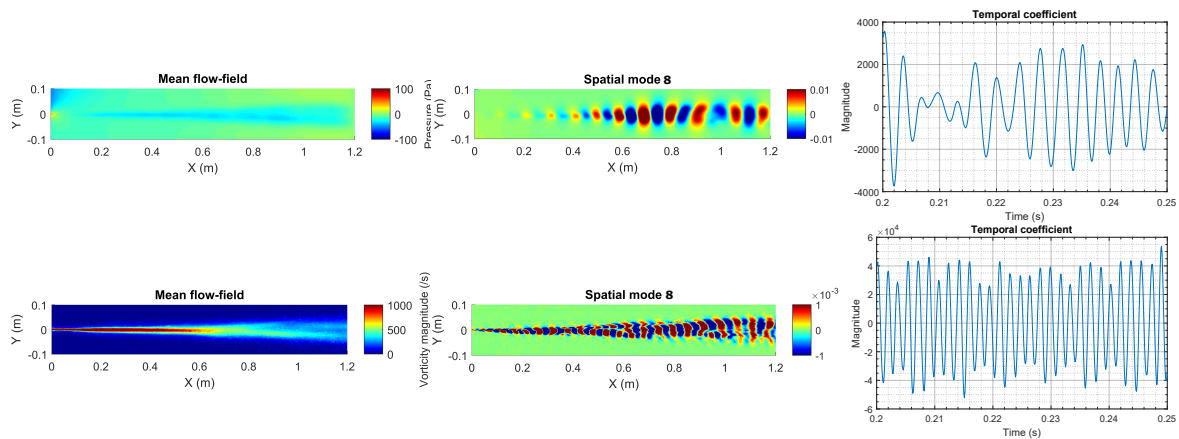


Figure 5.13: Spatial modes and temporal coefficients for operating point 1

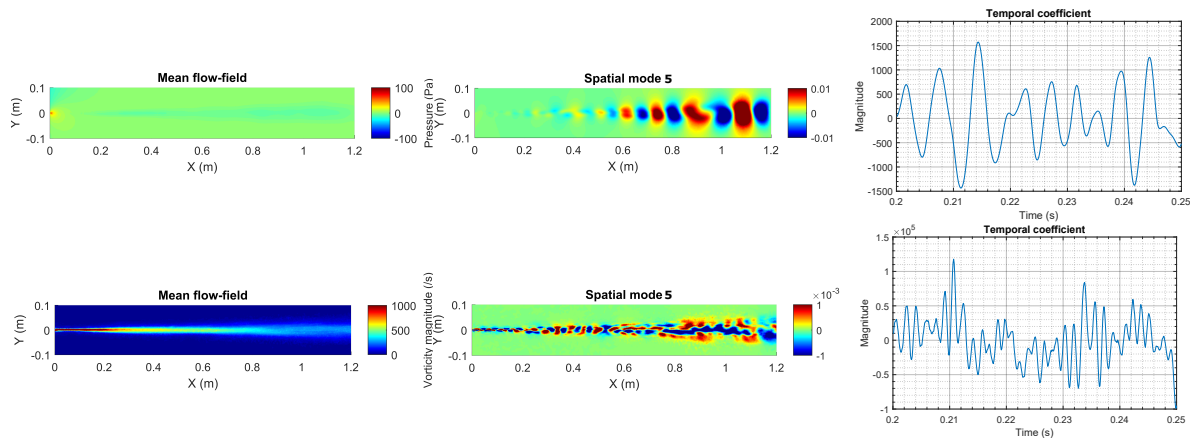
This is not ideal, basically the first few dominant modes are representing an artificial CFD byproduct instead of the actual flow physics. Simultaneously, it is a manageable problem because travelling waves merely cause rank inflation [24]. Thus, more modes are required to capture the total physics that theoretically impacts the computational efficiency or flow reconstruction quality. The impact of this is only noticeable for very sparse truncated sets, this work however retains 50 modes (as described in Section 5.2.2), which is not truly sparse. There are variants of the POD algorithm that are designed to deal with travelling structures. For more information on these algorithms see the work of Glavaski et al.[54] and Noack [104].

#### Mean mode + Mode 5 and 8

In this paragraph the mean field and two different POD-modes are shown. As discussed above, multiple modes are needed to represent the large pressure wave propagating through the domain. Different operating points require a different number of modes to represent their respective pressure wave. In Fig.5.14 the mean fields, spatial and temporal modes are visualised for operating point 1 and 4. The modes that are provided, are identified as the first dominant modes that seem unaffected by the pressure wave propagating through the domain.



(a) Operating point 1, velocity ratio = 1.85



(b) Operating point 4, velocity ratio = 1.48

Figure 5.14: Mean flow-field, and the the 1<sup>st</sup> spatial mode and temporal coefficients for operating point 1 and 4

The following things are observed from Fig.5.14 but generally holds true for all the training points.

- The mean vorticity flow-fields depict the linear growth of the vorticity thickness [38].
- Figure 5.14 provides a glimpse into a potential problem of the POD-Interpolation approach, but also in general when working with a MDM. Figure 5.14 provides two spatial and temporal modes for two operating points. The spatial modes look relatively similar (especially the spatial modes derived from the pressure data) and the amplitude of the temporal modes are the same order of magnitude. It seems that the modes describe (nearly) the same physics, even when the modes are of a different order.

This is a relevant observation, because it exposes a potential weakness. During the interpolation, the temporal coefficients from training cases are interpolated to obtain new temporal coefficients at new design points. But the interpolation is to a large degree naive, because it does not analyse the modes before for physical similarities. This approach simply interpolates the temporal coefficients from the training cases, because they have the same rank. Possibly interpolating between temporal coefficients that describe different physics. This is further discussed in Section 5.4 and Chapter 6.

Truncation, prediction and reconstruction is done in the exact same manner as for the Von Karman case (Section 5.2). To summarise, after the dimension reduction only 50 modes are retained. The temporal coefficients from the training cases are linearly interpolated to obtain the new coefficients at a new design point. These new coefficients are subsequently multiplied with spatial modes from training case 1 (base-case) and added to the interpolated mean flow-fields to obtain the new total flow-field.

### 5.3.2. Results

The results are divided in the prediction of the mean and unsteady flow-fields.

#### Prediction of mean flow-fields

The prediction of mean flow-fields is performed by interpolating between two spatial mean flow-fields. The results are shown in Fig.5.15 where the mean vorticity flow fields from the LES and the POD-Interpolation are shown.

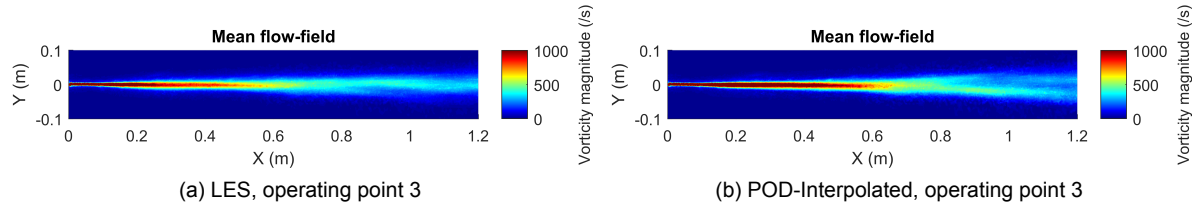


Figure 5.15: Mean flow-field comparison for operating point 3

The MRE for Fig5.15 is again very low 0.095%, that is similar to the MRE's for the mean flow-field comparison of the Von Karman street as described in Section 5.2.

#### Prediction of fluctuating flow-fields

For the fluctuating flow-fields six snapshots are shown at three different time-steps. The fluctuating vorticity flow-fields are provided in Fig.5.16. The intention of this figure is to analyse the prediction of the vorticity dynamics.

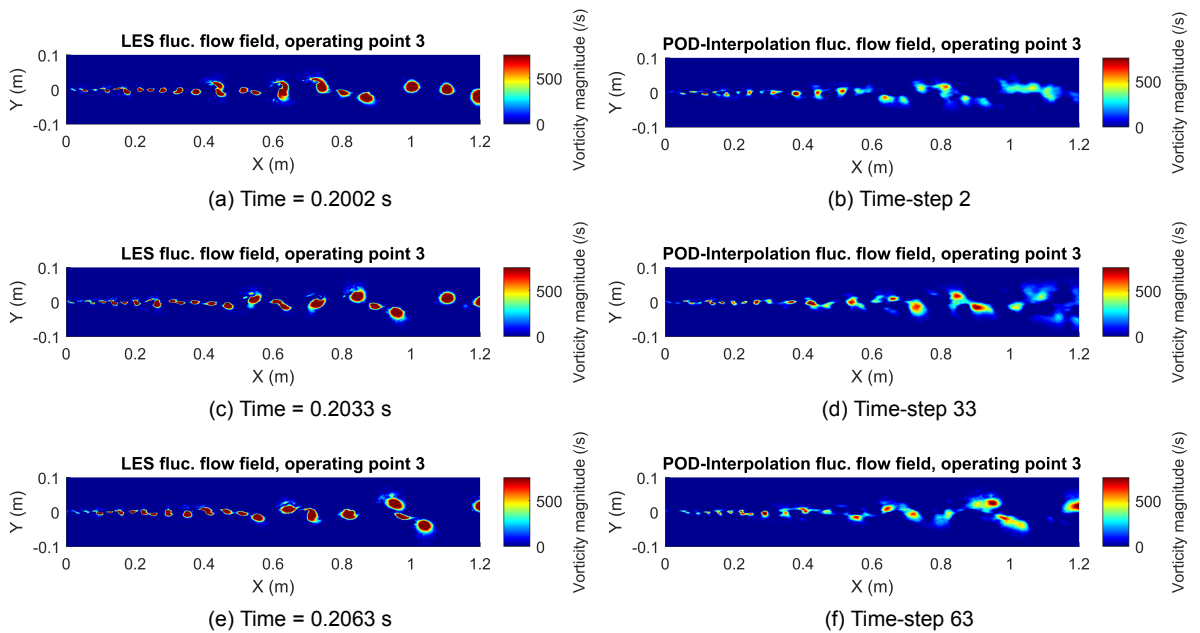


Figure 5.16: Predicted fluctuating flow-field compared against the LES fluctuating flow field, operating point 3

The following things are observed from Fig.5.16. The GIFs from the flow-fields can be found in the GitHub repository [5].

- The LES flow-field depict well-defined, nearly circular Eddies. The predicted flow-field depict smeared-out Eddies.
- Singular Eddies become difficult to identify, when compared to the LES field.
- The amplitude for the predicted Eddies are significantly smaller than the LES Eddies.
- The vorticity dynamics (i.e. roll-up, pairing, cannibalisation) are hardly visible in Fig.5.16, but even less so when analysing the GIFs.

- The GIFs show that the vorticity dynamics are poorly predicted and it looks very artificial. It becomes clear that the POD algorithm separates the spatial and temporal component. Whereas, actual physical vorticity evolves over space and time. The GIFs depict a vorticity that is clearly unphysical and inaccurate.
- The Eddies only translate, there is no or hardly rotation visible of the Eddies.
- The interaction between Eddies looks highly artificial.

The snapshots (Fig.5.7 and Fig.5.16) show that a prediction might be relatively decent. But if all the snapshots/figures are combined into a GIF, and the actual temporal evolution of the fluctuating flow-field is analysed the results are significantly worse. This trend is confirmed in the pressure signals Fig.5.8 but also below in Fig.5.17. The location for the probe points (P.P) are depicted in the Fig.D.3.

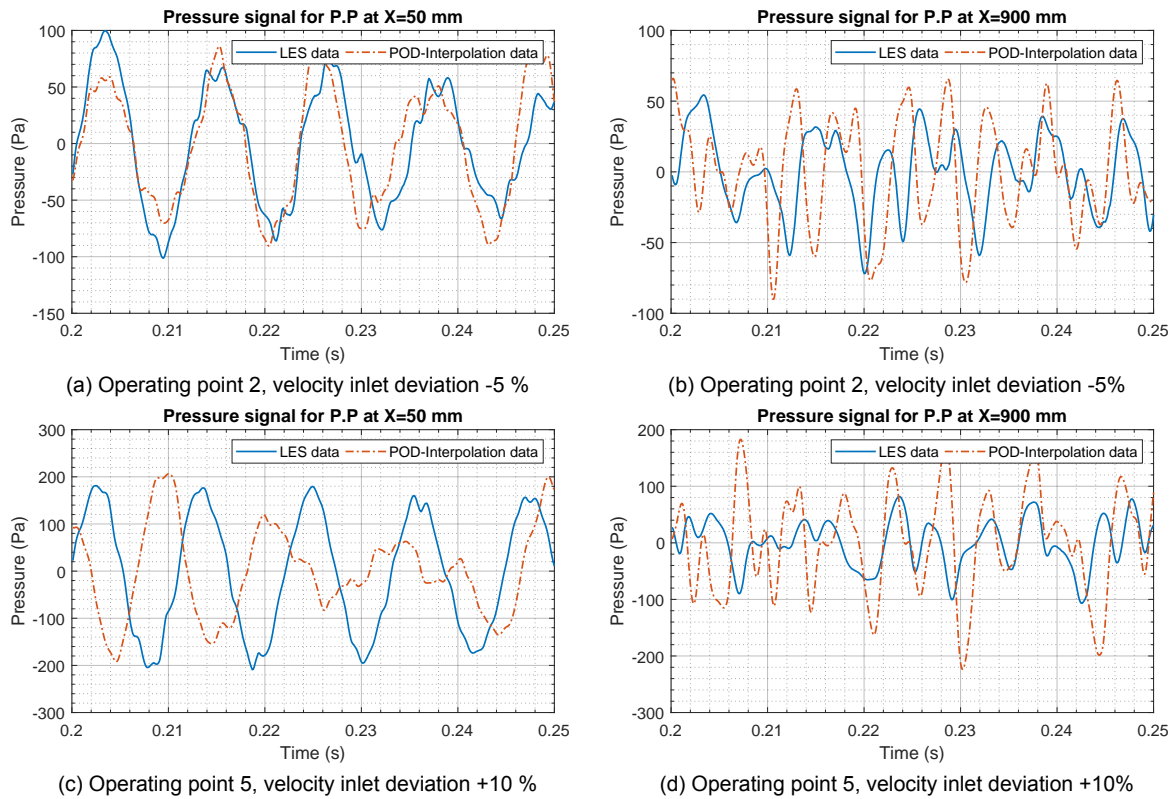


Figure 5.17: Pressure probe figures for operating point 2 and 5

Figure 5.17 depict the large pressure fluctuations propagating through the domain. The fluctuations that stem from the vortices are small in comparison to the large pressure wave. The predicted pressure signals are relatively of decent accuracy when the input variable deviation is (very) small, as is the case for operating point 2 that is shown in Fig.5.17 (a). Simultaneously, in Fig.5.17 (a) also a growing phase difference is observed for the two signals. When the input variable deviation is slightly larger, the accuracy of the predicted signal is significantly worse as shown in Fig.5.17 (c). Furthermore, correctly predicting the pressure at the outlet remains a challenge as can be seen from (b) and (d).

## 5.4. Discussion results

This section uses all the previous results to explain why this algorithm/framework fails to predict physically accurate flow-fields. The inaccurate results from previous sections are summarised as:

- Not being able to predict the key flow-feature of a von Karman Street that is the vortex shedding behaviour and frequency, see Fig.5.9 and Fig.5.10. This inability results in very high MRE (+95%) for the Fourier flow-fields compared at the fundamental frequency, see Fig.5.12.
- Being unable to accurately predict the flow quantities at points of interest, see Fig.5.8 and Fig.4.11.
- The predicted fluctuating flow-field tend to be more disorganised and having smeared-out Eddies, as seen in Fig.4.5 and Fig.4.9.
- The predicted vorticity fields can show reasonable agreement at certain time-steps, as Fig.4.9 shows. But once compiled into a GIF it can be seen that the vorticity field is poorly predicted.
- Key dynamical features (i.e. roll-up, pairing and cannibalisation) of vortices are not predicted and hardly any interaction between Eddies is observed.
- Flow-fields are very poorly predicted downstream: see Fig.5.7, Fig.5.8 and Fig.5.17.

To explain why this method fails to predict these things: 3 succeeding subsections dissect a key problem with the method for predicting flow-fields at different design parameters. Furthermore, Subsection 5.4.4 describes multiple attempts to overcome these key problems.

### 5.4.1. A (lack of a) common set of spatial modes

As stated before, Mak et al.[88] assumes that there is a common set of spatial modes. Thus over the entire design space the spatial modes should be similar to each other, for there to exist a common set of spatial modes. In this subsection this assumption is analysed and its implications are discussed.

To analyse this assumption, three different spatial modes from three operating points are visually compared. If the assumption by Mak et al. holds true the spatial modes should hardly differ for the varying operating points.

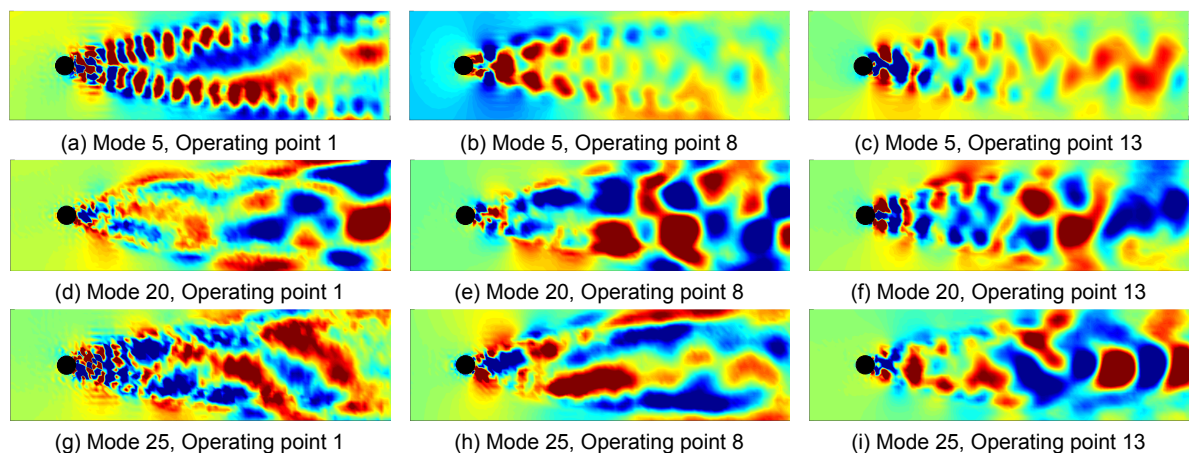


Figure 5.18: Spatial modes comparison for the von Karman case

The modes from Fig.5.18 are chosen to showcase large differences between the operating points. The first 3 dominant spatial modes, for every training point, tend to look rather similar (which partially can be seen in Fig.5.2). But higher order modes start to diverge from each other after a certain rank of the modes. Based on the spatial modes for the Von Karman case and the Mixing Layer case, it must be concluded that the assumption about a common set of spatial modes is **not** valid.



With this concluded, let's address the implications of this invalid assumption. The first complication occurs in the prediction step where the temporal coefficients are interpolated. The lack of similarity of the spatial modes does not directly impact the quality of the interpolation, however it is an indication that modes that describe different physics are interpolated. This was already briefly mentioned in Section 5.3, the POD-Interpolation approach is naive in the sense that modes are interpolated that capture different physics.

The second complication occurs in the reconstruction step, where newly obtained temporal coefficients are imposed on the spatial modes from the common set of spatial modes. Without a common set of spatial modes, the user has to decide which set of spatial modes is used. The impact of imposing it on different sets was investigated, but no large quality differences of the predicted flow-fields were found. The fluctuating flow-fields looked only marginally different and both resulted in inaccurately predicted flow-fields. Furthermore, the MRE (shown in Fig.5.12) was not significantly impacted by imposing the coefficients on different sets of spatial modes and remained very high. During this project the following guideline was used: impose the newly obtained on the set of spatial modes from the operating point with the lowest Reynolds number.

### 5.4.2. Interpolation

This subsection expands on the observations made in Section 5.2.4. The most relevant observation is that a phase and frequency differences exists for temporal coefficients from training cases, which are interpolated. To showcase the implications of this phase and frequency difference, a simplified example is presented of two arbitrary signals ( $S_1$  and  $S_2$ ) that are linearly interpolated to obtain a new coefficient signal  $S_3$ .

$$\begin{aligned} S_1 &= \cos(2\pi \cdot 80t), \\ S_2 &= \sin(2\pi \cdot 100t), \\ S_3 &= 0.5(S_1 + S_2). \end{aligned} \quad (5.3)$$

These signals, along with their representation in the frequency domain, are shown in Fig.5.19.

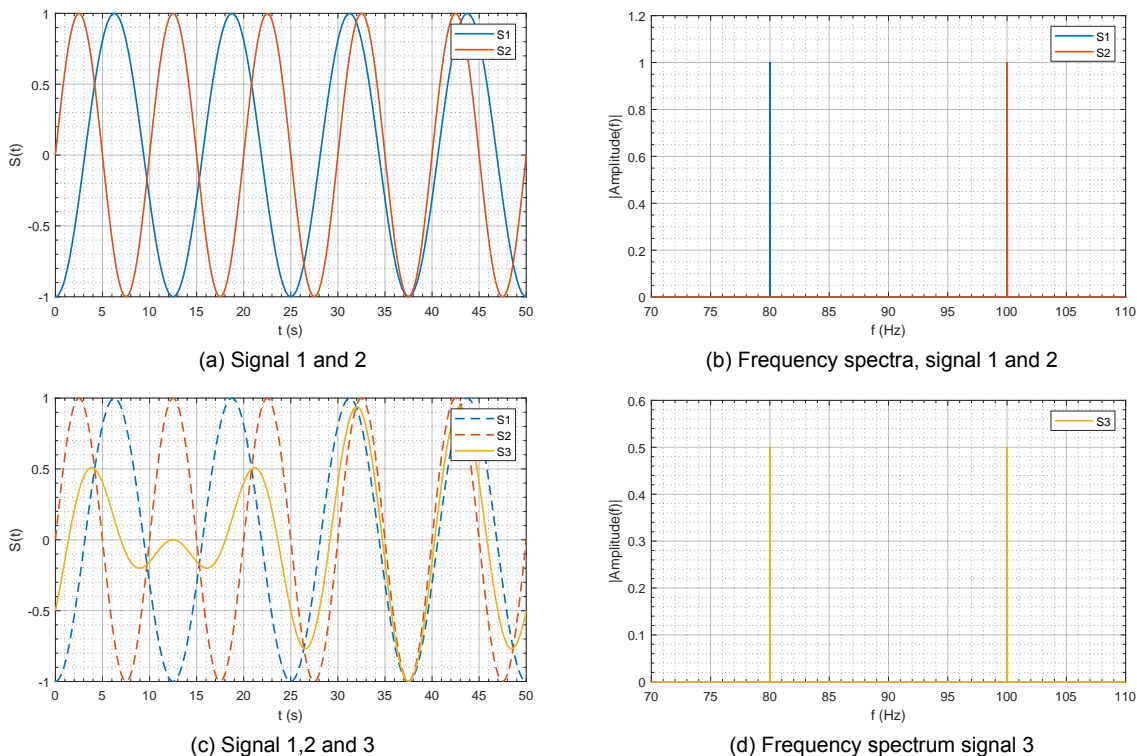


Figure 5.19: Simplified example of interpolation of temporal coefficients

From Fig.5.19 two things become clear. The first is that constructive and destructive interference plays a critical role in the interpolation of temporal coefficients. As can be seen in Fig.5.19 (c), where the interpolated signal  $S_3$  dances around an amplitude of zero between 7 and 17 seconds, as a result of destructive interference. Moreover, this phenomenon is observed in the prediction of fluctuating flow-fields where fluctuations disappears at certain time-steps. An example of this is given in Fig.5.20. The vortex shedding fluctuations at time-step 37 and 206 are merely 2 examples, the GIFs [5] exhibit this behaviour as well. Disappearing fluctuating flow fields are unphysical and severely impact the quality of the prediction.

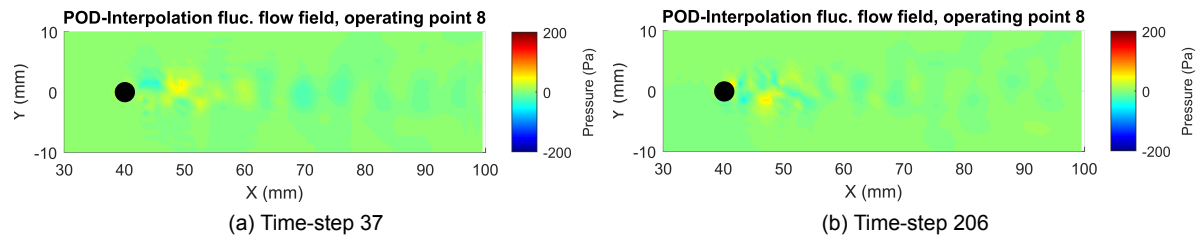


Figure 5.20: Two time-steps for the predicted flow-field, the fluctuating flow-field completely disappears at these instances.

The second thing that becomes clear from Fig.5.19, is that by interpolating the desired frequency of the interpolated signal is not predicted. The hypothetical vortex shedding frequency for example signal  $S_3$  is 90 Hertz, whereas  $S_1$  represents a shedding frequency of 80 Hertz and  $S_2$  100 Hertz. In Fig.5.19 (d) the inability of interpolation to predict the fundamental frequency is shown. This effect is also visible in Fig.5.10, where the fundamental frequency of the desired operating point is not predicted and the fundamental frequencies of the training cases are clearly visible. Interpolation results in signals/flow-fields that inherit the governing features from the training cases, instead of predicting the new features for new design points. This can be mathematically proven by using basic trigonometry rules, Simpsons [133] proved in the 18<sup>th</sup> century that

$$\sin(a) + \sin(b) = 2 \sin\left(\frac{a+b}{2}\right) \cos\left(\frac{a-b}{2}\right). \quad (5.4)$$

And thus when interpolating  $S_1$  and  $S_2$  the resulting  $S_3$  does not represent a new frequency, but merely the frequency from the training cases.

### Flows with dominant frequencies

The von Karman Vortex Street has dominant vortex shedding frequencies. Proven in this thesis, is that the POD-Interpolation method can never accurately predict these dominant vortex shedding frequencies for different Reynolds numbers. For applications cases with high dominant vortex shedding frequencies the consequences of interpolating become obvious, with the inability to predict the fundamental frequencies.

The Mixing Layer case does have vortex shedding frequencies but of significantly lower frequencies, ranging between 0 and 500 Hertz. For unphysical features (e.g. disappearing flow-field) to be visible for low frequencies it require significantly longer run-times before destructive or constructive interference plays a role. Sometimes inaccurate features are not visible because the available data-range is too limited. For these cases, the absence of these features is positive but it does not imply that the POD-Interpolation is not inherently flawed for predicting dominant frequencies. This can partially be seen in Fig.5.17 (a). As time progresses the prediction becomes less accurate and the phase difference grows. Longer run-times and the the phase difference eventually leads to inaccurate behaviour, such as flow-fields completely disappearing.

It remains to be seen whether the POD-Interpolation approach would be suitable for flows without dominant frequencies. Flows for example with a turbulence noise profile. Possibly, the inaccuracy of the approach is somewhat hidden because no dominant frequencies has to be predicted. According to the author however, even for these flows the merit of combining CFD and the POD-Interpolation algorithm is doubtful.



### 5.4.3. Prediction of the downstream wake

Flow-fields and properties are in general poorly predicted at the outlet. For example, see Fig.5.7, Fig.5.8 and Fig.5.17. However, this is not surprising as 2 factors play an essential role in this. Sirovich [135] has proven that the POD eigenvalues correspond to the turbulence energy spectrum. Sirovich related the  $n^{\text{th}}$  wave number ( $k_n$ ) to the  $n^{\text{th}}$  Eigenmode, with the relation  $n \propto k_n^3$  [84, 135]. This finding is of significant for dimension reductions. Because dimension reductions are equivalent to drastically cutting in the turbulence energy spectrum. The first factor is therefore: dimension reductions can be seen as a spatial filter that only retains the large scaled Eddies.

The second factor that plays an essential role, in the poor prediction of the downstream wake, is the separation of space and time. The Snapshot-POD enables a decomposition that results in spatial only modes and temporal coefficients. However, turbulence and vorticity is a spatial-temporal phenomena (as described in Section 2.1.2). As a result of this decomposition: 1) the vorticity dynamics in general are poorly predicted and, 2) the cascading of vorticity energy is eliminated. The 2 aforementioned factors ensures that the prediction of the downstream wake is rather challenging. Due to the direct relation between modes and the wave number, a logical assumption would be that by adding more modes the prediction of the downstream wake improves. Similar to the converging behaviour shown in Fig.5.5. Predicted flow-fields however exhibit a diverging trend, shown in Fig.5.21. The x-axis

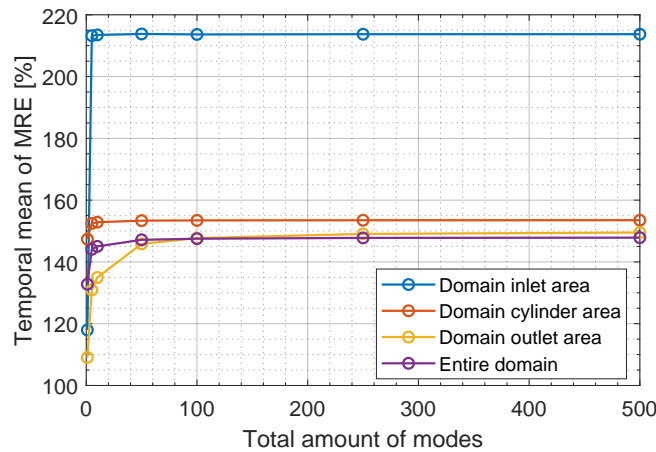


Figure 5.21: Time averaged MRE that compares the prediction quality to the amount of modes used for prediction. The 7 circles respectively represents 1,5,10,50,100,250 and 500 modes. See Fig.D.2 for the definition of the different domains.

of Fig.5.21 represents the total amounts of modes that are used for prediction of the flow-field from objective point 8. Besides that Fig.5.5 shows that the POD-Interpolation approach fails, it also highlights that it hardly matters whether 1,5,10 or 50 modes are used for the prediction. The deviation is for 1 mode already larger than the LES flow-field and only grows with more modes.

### 5.4.4. Attempts to overcome the limitations

In the project multiple attempts are made to overcome the limitations of constructive and destructive interference and the inability to predict the vortex shedding frequencies. One of these attempts is described because it partially resolved the inability to predict the fundamental frequencies. The second reason why this method is highlighted is because it led to the algorithm described in Chapter 6.

The general idea behind the POD-Scaling method is to replace the interpolation step with a scaling step. The von Karman Vortex Street case lends itself for scaling, since the scaling relation for the vortex shedding is known (Eq.4.2). Only periodic temporal coefficients can be scaled and as Fig.5.2 shows the first 3 dominant temporal modes are nearly sinusoidal. Higher order modes become increasingly aperiodic and can not be scaled. To be specific, the first 3 temporal coefficients of operating point 1 are scaled to any objective point. The results showed reasonable improvements compared to the POD-Interpolation algorithm. The fundamental frequency for the predicted design points is captured with the POD-Scaling method. This improvement initialised the concept of the sDMD-Scaling approach. As the sDMD algorithm provides purely sinusoidal temporal coefficients.

## 5.5. Conclusions POD-Interpolation

Chapter 5 described the implementation and results for the POD-Interpolation algorithm. First the methodology is described in Section 5.1 and it consists of the following steps:

1. Define the design space with the definition of the training points and objective points.
2. Compute the training points with a LES solver.
3.
  - (a) Decompose the flow-field data into spatial modes and temporal coefficients
  - (b) Obtain a dimension reduction by truncating to a reduced set of modes.
  - (c) Predict the new dynamics by interpolating the temporal coefficients of training points.
4. Reconstruct the flow-field for objective points.

The algorithm is implemented on 2 test cases, the Von Karmen Vortex Street and Mixing Layer. The design space for the both cases respectively consist of velocity and mass flow deviations. Section 5.2 and Section 5.3 provide the results for these test cases. The results for both cases are summarised as: inaccurate and nonphysical. The algorithm fails to predict key flow-features for the Von Karman Street as the vortex shedding behaviour and frequency is poorly predicted. This inability results in very high MREs. Furthermore, for the Mixing Layer the vorticity fields are poorly predicted. Key dynamical features (i.e. roll-up, pairing and cannibalisation) of vortices are hardly predicted and no interaction between Eddies is observed.

Multiple root-causes are found that explain why the POD-Interpolation algorithm fails to accurately predict flow-fields. The first root cause is a failed assumption. There exist no common set of spatial modes over the design space. This failed assumption complicates the flow-field reconstruction aspect, since it is unclear on which set of spatial modes the newly obtained temporal coefficients should be imposed upon. Furthermore, varying spatial modes (over the design space) indicates that modes from different training points describe different physics. This is problematic because these modes (that describe different physics) are interpolated to predict flow-fields at objective points. The second root-cause is interpolation of the temporal coefficients. Interpolation in the time and frequency domain results in inaccurate new temporal coefficients due to phase and frequency differences between coefficient signals. Constructive and destructive interference results in new coefficient signals that exhibit non-physical behaviour (i.e. disappearing flow-fields). It is impossible to predict new dominant frequencies with interpolation, because it merely projects the dominant frequencies of the training cases with lower amplitude.

The governing task of Chapter 5 is to provide a verdict on the POD-Interpolation algorithm. The verdict is that this framework is unable to accurately predict flow-fields. There is no merit in combining LES simulations and the POD-Interpolation algorithm to reduce the computational cost and time for optimisation studies. Multiple attempts were made to overcome the limitations of the POD-Interpolation algorithm and eventually led to the formulation of the sDMD-Scaling approach.

# 6

## Sparse DMD-Scaling

The closing remark of the previous chapter serves as a fitting introduction for Chapter 6. Multiple attempts were made to overcome the limitations of the POD-Interpolation algorithm. An attempt, that is referred to as the POD-Scaling approach, showed reasonable improvement and indicated that scaling has more potential than interpolation. A limitation of the POD-Scaling approach is that only the quasi-periodic temporal coefficients can be scaled, whereas the vast majority of coefficients are aperiodic. The sDMD algorithm mitigates this limitation because the flow decomposition results in strictly periodic temporal coefficients. Chapter 6 revolves around the sDMD-Scaling approach and intends to:

1. Describe the methodology.
2. Provide the results for the Von Karman Vortex.
3. Describe how scaling relations can be discovered and used for prediction of flow-fields.
4. Give a verdict on the CFD and sDMD-Scaling combination.

The outline is as follows, Section 6.1 describes the methodology. Section 6.2 describes the implementation of the algorithm on the Von Karman Vortex street. The Mixing Layer is not implemented due to time-constraints. Section 6.3 discusses the possibility to extend the algorithm to cases where the scaling relations are not obvious and known. Section 6.4 provides a discussion on the performance of the algorithm. Lastly, in Section 6.5 conclusions are drawn on the four aforementioned points.

### 6.1. Methodology

The DMD-Scaling methodology is to a large degree similar to the POD-Interpolation methodology. The first step, for both approaches, is to define the design space (Table 4.3) and classifying the training and objective points. The second step is the computation of the training points with an LES solver to gather the required training data. The third step is where the DMD-Scaling approach diverges from the POD-Interpolation approach, by replacing the interpolation with a scaling method. The governing steps of the methodology are listed below.

1. Define the design space with the definition of the training points and objective points.
2. Compute the training points with an LES solver
3.
  - (a) Decompose the flow-field data and extract key dynamical features.
  - (b) Obtain a dimension reduction by truncating to a reduced set of modes.
  - (c) Predict the new dynamics for objective points by **scaling** the temporal coefficients and mode amplitudes.
4. Reconstruct the flow-field for the new design parameters.

Lets expand on the scaling aspect of the methodology. Section 3.4.2 introduced the governing DMD equation, which for good measure is reiterated below in Eq.6.1. A flow-field is decomposed into spatial modes ( $\phi$ ), temporal coefficients ( $\omega$ ) and mode amplitudes ( $b$ ).

$$\mathbf{D}_t = \sum_{m=1}^r \phi_m e^{\omega_m t} b_m \quad (6.1)$$

Scaling relations are used for the temporal coefficients and mode amplitudes to predict the dynamics of objective points. The spatial modes are unaltered. The newly scaled temporal coefficients and mode amplitudes are imposed on the spatial modes from training case 1 (Table 4.3). The quality of the sDMD-Scaling method revolves to a large degree around the quality of the scaling relations, as will be shown over the next few sections.

## 6.2. Von Karman Vortex Street

Section 6.2 is ordered differently to its equivalent Section (Section 5.2.4) of the POD-Interpolation approach. Because there is no ranking of modes and the dominant modes are not obvious. This problem is mitigated by using the Sparsity Promoting DMD algorithm [68, 69] that identifies the relevant modes, but does not rank them. Subsection 6.2.1 first describes the mode selection, succeeded by the truncation and the visualisation of spatial modes. Step 1 and 2 of the sDMD methodology are discussed in Chapter 4.

### 6.2.1. Mode selection, truncation and visualisation

Mode selection and truncation for a DMD algorithm is not as trivial as for the POD-algorithm, there is no set criteria (Eq. 3.3) that has to be met. For each of the training points the typical DMD procedure (Section 3.4.2) is performed. This results in a large number of modes that are highlighted by blue circles in Fig. 6.1. These modes are scrambled in terms of their relevancy. To identify the relevant modes the DMD procedure is augmented with a penalty function (Eq. 3.22). This penalty function promotes sparsity, as the use of more modes results in a higher penalty that Eq. 3.22 tries to minimise. The relevant modes are highlighted as the red crosses in Fig. 6.1. In Fig. 6.1 the mode selection for operating points 1 and 13 are shown.

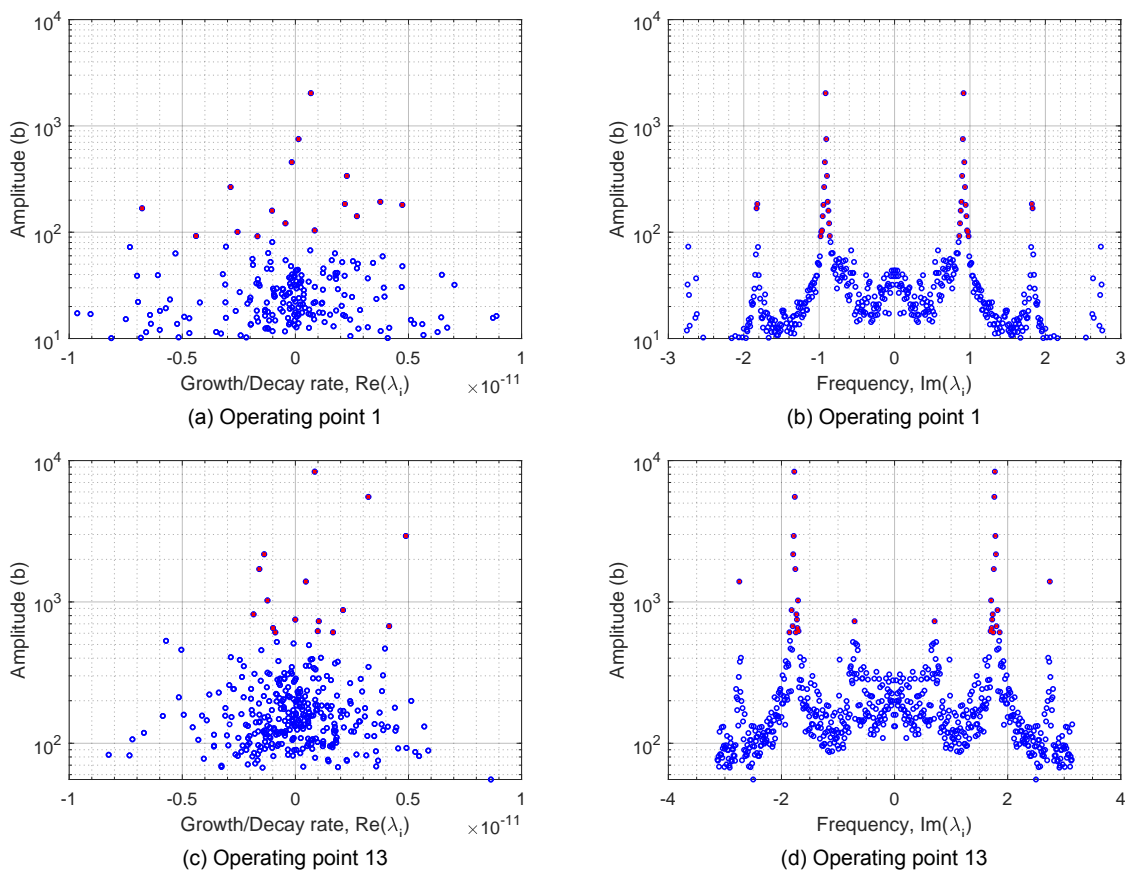


Figure 6.1: The mode selection for two operating points, between all the available modes (blue circles) and the identified relevant modes (red cross).

Figure 6.1 shows that the spectrum is symmetric with respect to the frequency. This is a consequence of processing real-valued data that yields eigenvalues that are either purely real (non-oscillatory) or come in complex conjugate pairs [128]. Only if the input is complex valued data, than the spectrum becomes asymmetric [128]. Furthermore, Fig.6.1 shows how the sDMD algorithm identifies the modes around the fundamental frequency and its harmonics as the most relevant. These can be seen as the large peaks protruding in (b) and (d). When emphasising sparsity the sDMD method selects the modes with highest amplitudes of the dominant frequencies.

Regarding the truncation, a number between 18 and 22 complex mode pairs are chosen. This varies because of the optimisation and the variability in the regularisation parameter ( $\gamma$  in Eq.3.23) that is incorporated in the sDMD algorithm. The rank of truncation of approximately 20 complex mode pairs was determined to approach the POD rank of truncation of 50 Modes. 20 Complex mode pairs provided visually identical flow-fields compared to the LES flow-fields. Quoting Brunton et al. [24]: "selecting the right rank of truncation is one of the most subjective and important steps for a DMD-algorithm". This is especially true for the standard DMD-algorithm but partially for the sDMD-algorithm.

Above the mode selection and truncation are discussed, which leaves the visualisation of relevant spatial modes in Fig.6.3. Three spatial modes, from training point 13, are chosen that represent the fundamental frequency (3764 Hz) and a relative low (1501 Hz) and high (5826 Hz) frequency mode. A single-sided spectrum is shown beforehand in Fig.6.2.

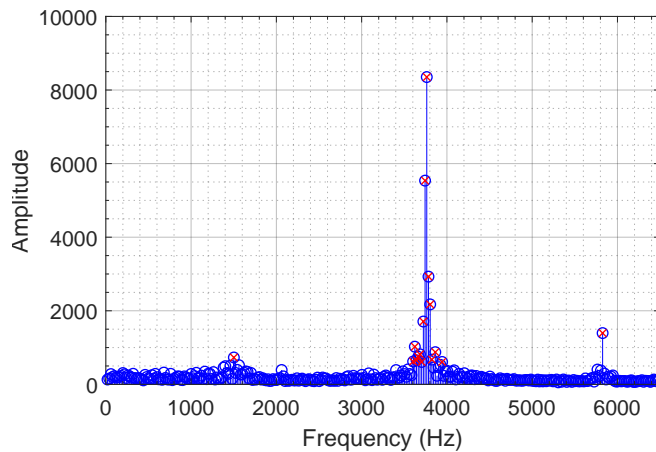


Figure 6.2: A different representation of Fig.6.1 (d) where only the single-sided spectrum is shown for operating point 13. Blue circles represents again all the available modes and the circles with red crosses represents the selected modes

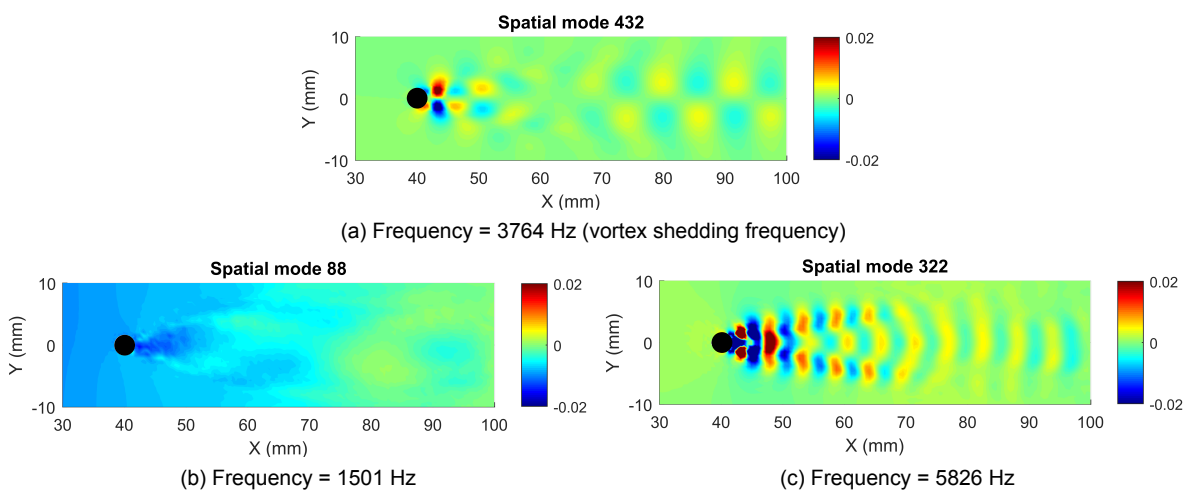


Figure 6.3: Spatial modes for operating point 13 representing 3 different dominant frequencies in the spectra of Fig.6.2.

## 6.2.2. Prediction and reconstruction

There are few fluid dynamic cases that are as well documented and well-understood as the Von Karman Street, it lends itself for scaling. In Chapter the 4 the physics is described with a focus on the Strouhal-Reynolds relation (Eq.4.2, Fig.4.2). This relation describe how the fundamental vortex shedding behaviour changes with a velocity deviation, the linear growth is captured by the CFD simulations as shown in Fig.4.6. The Strouhal-Reynolds relation is used to scale the frequencies of the temporal coefficients to a new design point.

The second relation is derived from the Bernoulli [75] equation, which correlates the pressure to the square of the velocity deviation ( $P \propto U^2$ ). Strictly speaking, using the Bernoulli equation is not correct since unsteady viscous flows are simulated. Nevertheless, these 2 relations are used as initial relations to test the efficacy of the sDMD-Scaling approach. These scaling relations are visualised in Fig.6.4 against the mass flow deviation. These scaling relations are partially revised in Section 6.3.

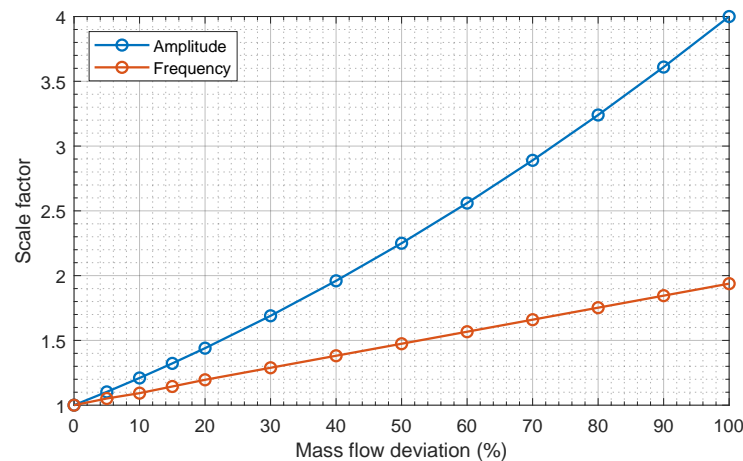


Figure 6.4: Initial scaling relations for the Von Karman Vortex Street

Regarding the implementation of these scaling relations. Each complex valued temporal coefficient consists of a decay rate and frequency. Since there are no known scaling relations for the decay rate these are unaltered. Only the frequencies of the temporal coefficients are scaled to predict the dynamics of new design points. Roughly 20 complex mode pairs account for 20 different frequencies. The initial approach is to implement all these 20 frequencies (/modes) with the same scaling relation. To be specific, all the selected modes (red crosses) from Fig.6.1 (b) are scaled with the same scaling factor. This is to a degree incorrect, since the Strouhal-Reynolds relation only describes the change of the fundamental frequency with velocity. Regardless, a single scaling factor is imposed on all the frequencies from the selected modes. In an identical manner, all the scalar mode amplitudes are scaled with the Bernoulli pressure correlation.

The spatial modes, as stated before, are not altered in any way. This is because streamwise and spanwise scaling relations are hardly available, also the implementation of such relations is rather complex. Therefore, scaling is deemed not viable for spatial modes. An alternative solution would be to interpolate between the spatial modes of 2 training cases. This is attempted and deemed not viable, because it is difficult to match spatial modes that describe the same physics. Moreover, often the dissimilarities between spatial modes is too great to be interpolated.

The reconstruction is done in an identical manner to the POD-Interpolation approach. The **unaltered** spatial modes ( $\phi$ ) are used from training point 1. The new temporal coefficients, originally from training point 1, ( $\omega$ ) are comprised of the: 1) **unaltered** decay rates, and 2) **scaled** frequencies. The mode amplitudes ( $b$ ) from training point 1 are **scaled** to new design points. Multiplying these three components (Eq.3.21) results in a new predicted flow-field.

### 6.2.3. Results

The results are given in 2 paragraphs that describe the prediction of fluctuating flow-fields and the error quantification. The mean-field is not addressed in Chapter 6. The prediction of mean flow-fields is handled in an identical manner to the POD-Interpolation approach. Interpolating mean flow-fields is successfully done and never exceeds a MRE of 0.07%. For more information on this topic the reader is directed to Section 5.2.4.

#### Prediction of fluctuating flow-fields

For the prediction of the fluctuating flow-fields the same snapshots are shown as in Chapter 5. First 6 snapshots are shown that compare the LES flow-field and the predicted sDMD flow-field. This is succeeded by 6 snapshot that compares the predicted POD-Interpolated flow-field and sDMD-Scaled flow-field. Furthermore, for the GIFs the reader is directed to the GitHub page [5]. The GIFs provide a better comparison than the snapshots.

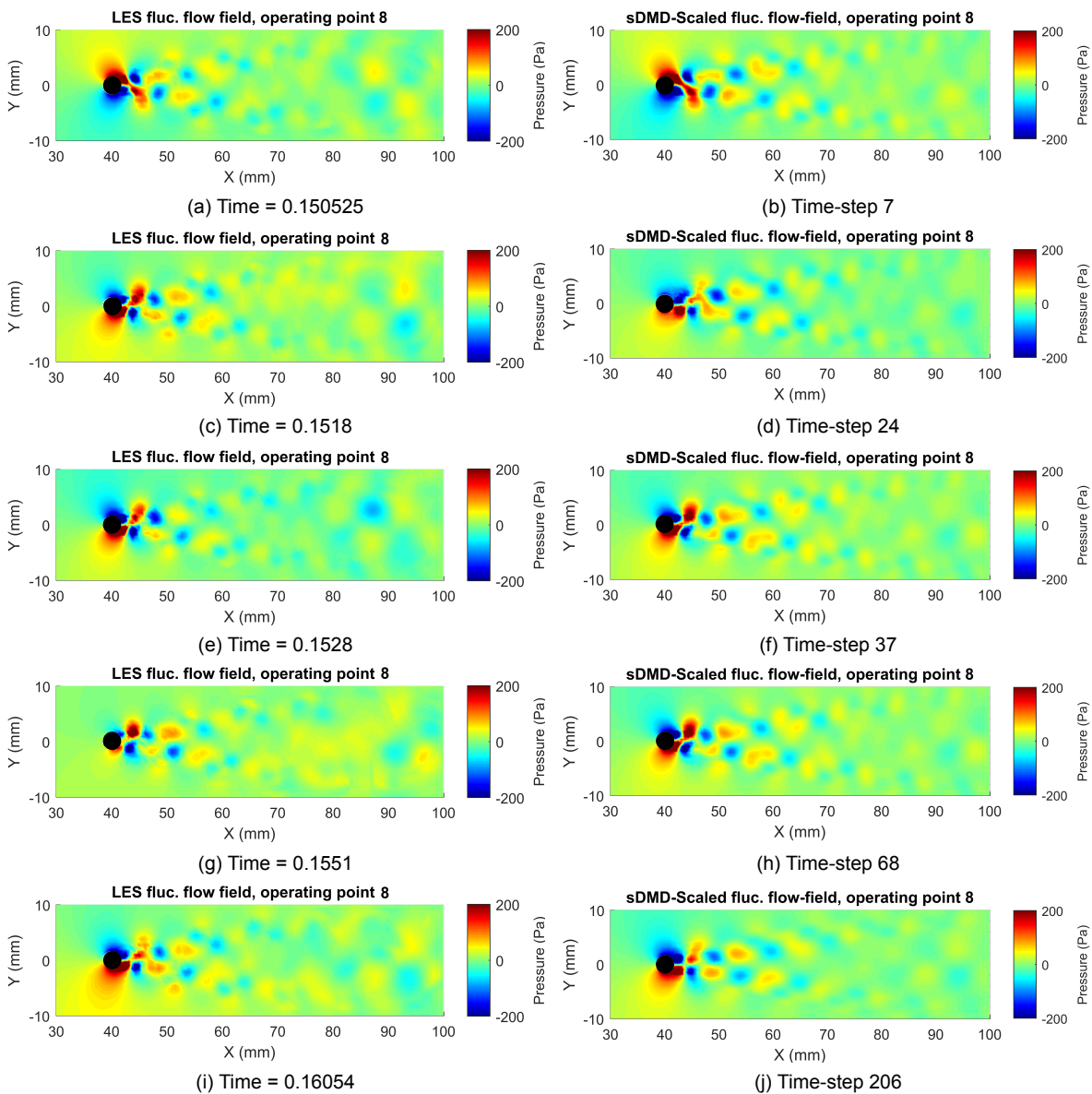


Figure 6.5: Comparison between the LES and sDMD-Scaling flow-field

The following observations are drawn by comparing the LES and sDMD-Scaled flow-fields.

- The vortex shedding fluctuations near the cylinder are captured with relative high accuracy.
- The fluctuations near than the downstream wake are poorly predicted.



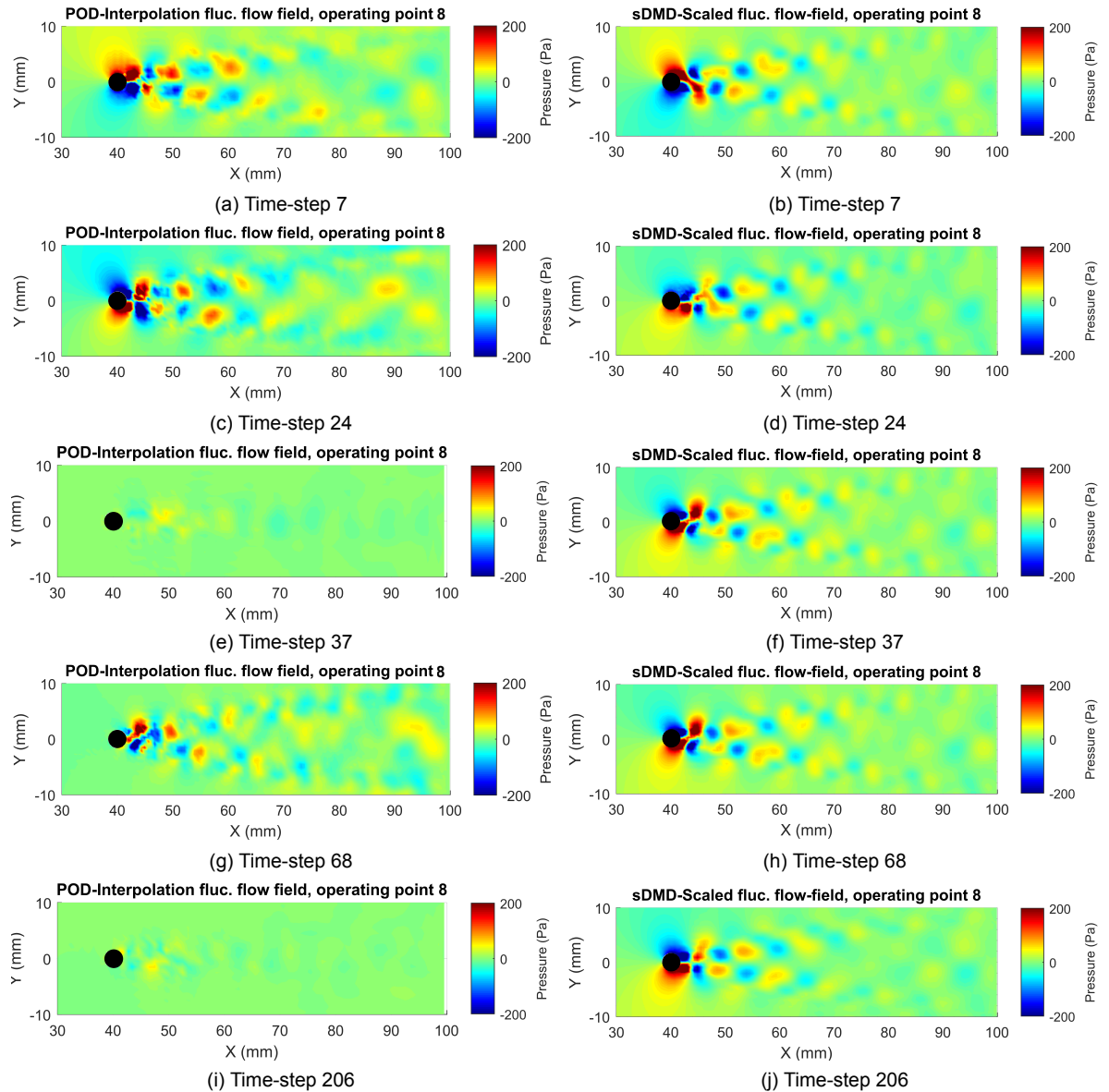


Figure 6.6: Comparison between the LES and sDMD-Scaling flow-field

The following observations are drawn by comparing the sDMD-Scaled and POD-Interpolated flow-fields.

- The sDMD-Scaled flow-fields looks cleaner with more circular Eddies and less irregular patterns.
- Both predict different downstream flow-fields. The sDMD-Scaled downstream flow-field shows more similarities to the LES downstream field, than its POD-Interpolated counterpart. The predicted downstream fields however differ significantly in their amplitude.

But maybe the most significant observation is that the flow-field does not disappear at times shown in Fig.6.6 (i,j). The mitigation of this nonphysical behaviour is considered one of the biggest improvements of the sDMD-Scaling approach, compared to the POD-Interpolation approach. These observations are succeeded by the pressure figures, from objective point 8, shown in Fig. 6.7.



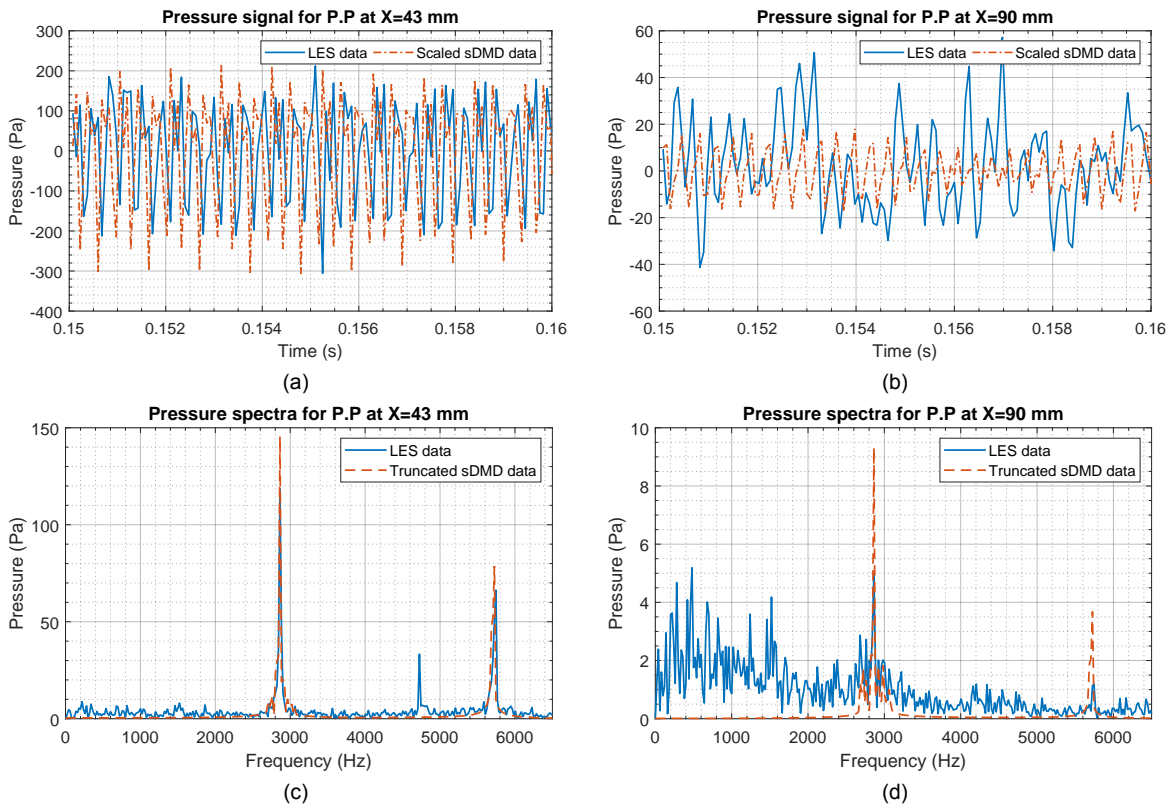


Figure 6.7: Pressure signal and spectra for two different points for objective point 8. See Fig.5.8 and Fig.5.9 for comparison.

The inability of the POD-Interpolation algorithm to predict the fundamental frequency is shown in Fig.5.10, the ability of the sDMD-Scaling algorithm is shown in Fig.6.7 and Fig.6.8.

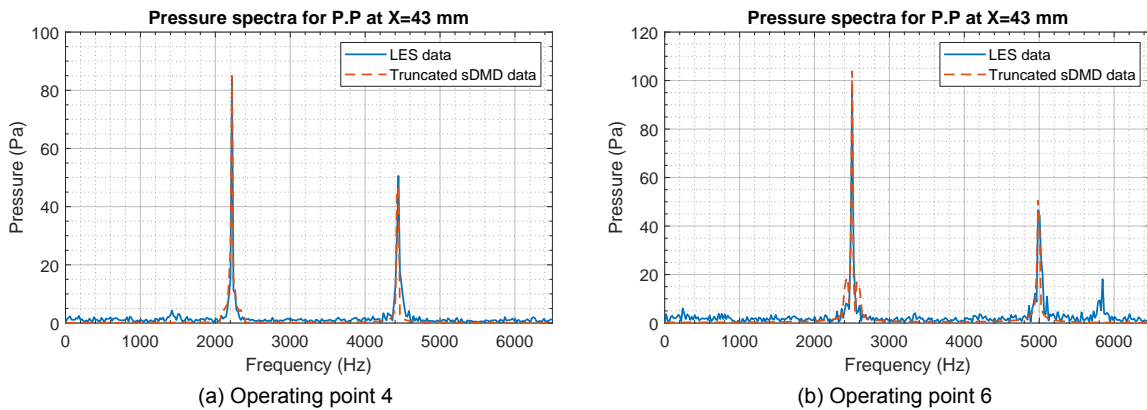


Figure 6.8: The pressure spectra for two operating points. A comparison can be drawn with Fig.5.10.

The predicted pressure signals (e.g. Fig.6.7 (a)) approximate the LES signals reasonably well. At downstream probe locations the prediction is of lesser quality. Based on the 4 spectra figures, featured on this page, it can be stated that the fundamental frequency is predicted with high accuracy.

**Error quantification of the results**

In this paragraph the error between the LES flow-field and the predicted flow-field is determined. The RMS spatial-field is not shown in this section since it is nearly identical to Fig.5.11. A linear decomposition method, like POD or DMD, struggles with regions of high nonlinearities such as stagnation point, angle of separation and boundary layers.

A more important figure, that shows the improvement over the POD-Interpolation approach, is given in Fig.6.9. The MRE shown in the Fig.6.9 describes the error between the Fourier flow-fields at the fundamental frequency for each operating point. The x-axis represents the mass flow deviation as described in Table 4.3.

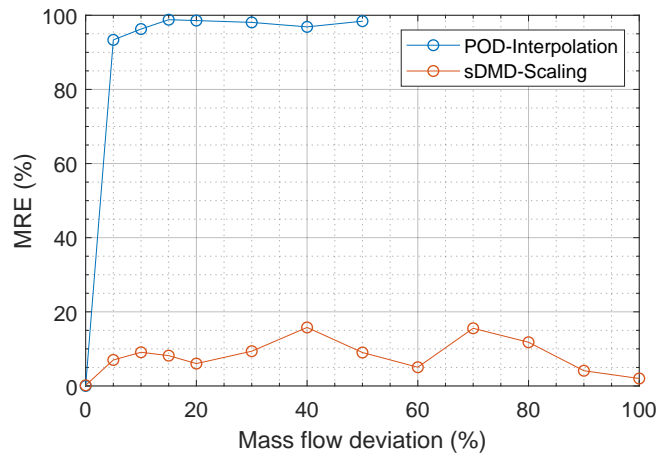


Figure 6.9: Mean Residual Error for the two governing algorithms

Figure 6.9 shows the inability of the POD-Interpolation approach to predict a physical accurate flow-field. Whereas, the proposed sDMD-Scaling algorithm can actually predict a physical flow-field with an acceptable accuracy loss. A significant improvement is made in this regard.

An additional improvement is made regarding required training data. For the POD-Interpolation approach, always 2 training points are necessary for the prediction of a objective point. For the rare case that scaling relations are known, only 1 training point is required to predict an objective points in the design space. Theoretically, any flow-field within the limits of the scaling relations. As an indication, the used Strouhal-Reynolds relation (Eq.4.2) is valid up to a Reynolds number of  $2 \cdot 10^5$ . To emphasise, with training point 1 and the available scaling relations can (theoretically) flow-fields be predicted up to a Reynolds number of  $2 \cdot 10^5$ . For unique application cases that have scaling relations, the sDMD-Scaling approach is an effective way of exploring a design space with acceptable accuracy losses.

### 6.3. Extending the algorithm

Three reasons drive the need to extend and improve the proposed sDMD-Scaling algorithm. The first and governing reason is that scaling relations are required. However, rarely does a fluid dynamic case have known scaling relations. This severely limits the use of this algorithm to application cases with available scaling relations. The second reason is that single scaling factors (see Section 6.2.3) are imposed on every mode, and this is most likely incorrect. Nevertheless, as an initial test it provided good results. The third reason is to provide additional learning benefit by discovering and formulating new scaling relations. By overcoming these challenges can the algorithm truly become a viable asset in the exploration of a design space.

The sDMD algorithm decomposes any flow-field into 3 components, but provides no ranking of modes. This complicates the discovery of any scaling relations. To overcome this, the modes are ranked in descending order of their mode amplitude. A type of ranking is necessary to compare and match modes that describe the same physics. Therefore, the first step in discovering scaling relations is to rank the modes according to their amplitude. The second step is to visualise the 3 components with respect to the design parameter. Ideally, patterns emerge by visualising the components. The third step is the extraction step of scaling relations. This step allows for creativity, different algorithms can be used to extract the relations from the data. For example, when a pattern or relation is obvious (e.g. Strouhal-Reynolds) a simple linear regression is sufficient. When relations are less obvious, more complex algorithms must be used to extract physically meaningful relations.

The methodology for discovering relations is summarised below.

1. **Order and rank.** The modes are ranked in descending order, depending on their mode amplitude.
2. **Visualise** Each mode from the 3 components (spatial, temporal, amplitude) with respect to design parameter variations.
3. **Extract scaling relations.** By using an algorithm that extracts physically meaningful relations.

The following 3 subsections implements this procedure for the mode amplitudes, temporal coefficients and spatial modes.

### 6.3.1. Mode amplitudes

The first component that is discussed are the mode amplitudes ( $b$ ). These are easily visualised as they are ranked scalar values. In Fig.6.10 the four strongest mode pairs are visualised for all mass flow deviations.

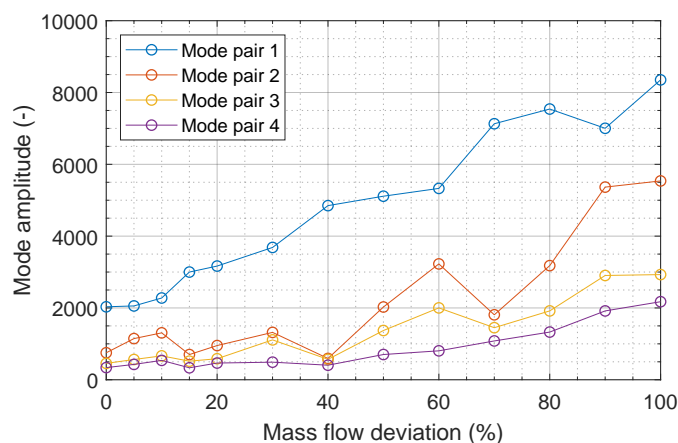


Figure 6.10: Mode amplitudes for the four leading mode pairs

The following observations are made, based on Fig.6.10.

- The Bernoulli derived scale factor ( $P \propto U^2$ ) reasonably approximates the leading mode pair. A scale factor of  $\approx 4$  can be seen between operating point 1 (0% deviation) and 13 (100% deviation), thus a doubling of the mass flow and the velocity.
- A noticeable decline is present for the lower order mode pairs. This indicates that imposing a single scaling factor to all the modes is incorrect, as is previously done in Section 6.2.3).
- For unknown reasons are 3 out of the 4 leading mode pairs converging at a mass flow deviation of 15%, 40% and 70%.

Regarding the extraction of new scaling relations different algorithms can be used. It predominantly depends on the amount of training points that are available. Visualised in Fig.6.10 are 13 different operating points, as defined in the design space. However, it is likely that real applications uses more sparse design spaces since every operating point is a computationally expensive LES computations. Regardless, the patterns in Fig.6.10 are relative clear and therefore a linear regression [24] is used. This regression occurs between training point 1 and 13. Furthermore, due to time constraints it was not possible to: 1) research the ideal balance between the number of training points and the quality of scaling relations, or 2) try different algorithms. It is important to emphasise that the extraction aspect is open to creativity, different order polyfits, regression methods or even a Neural Network can be used.

### 6.3.2. Temporal coefficients

The complex valued temporal coefficients are comprised of two values, the (real) growth/decay rate and the (imaginary) frequencies. The leading 22 frequencies for every operating point are visualised in Fig.6.11. Approximately 20 modes are required to identify reasonable differences in frequencies, 12 leading modes represent a narrow bandwidth near the fundamental frequency. This can be seen in Fig.6.1, where the majority of selected modes (red crosses) surround the fundamental frequency.

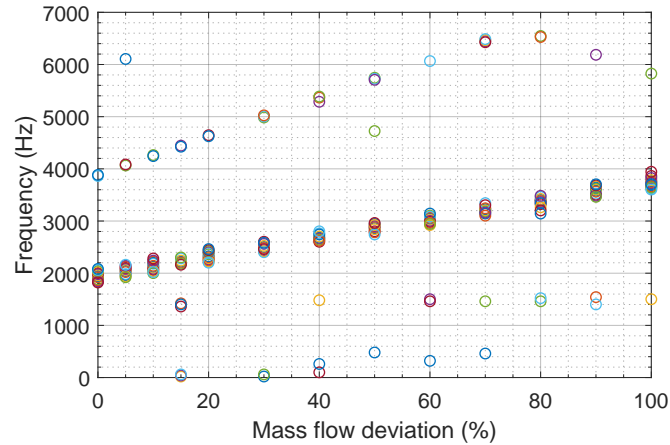


Figure 6.11: Frequency of the 20 leading modes with respect to variations in the flow parameter

Unsurprisingly, the linear growth of the fundamental frequency is accurately captured as can be seen by all the circles between the range of 2000- 4000 Hertz. A secondary batch of modes is observed between roughly 4000- 7000 Hertz. These modes presumably capture the harmonics of the fundamental frequency, that for majority also exhibit a linear growth over the design space. After 80 % it declines for unknown reasons.

Figure 6.11 offer a few possibilities to extract different scaling relations. For example, the batch of modes between 4000-7000 Hertz has a steeper incline compared to the linear growth of the Strouhal-Reynolds relation. A polyfit, or piece-wise linear regression would be able to capture this steeper incline and subsequent decline. However, **no new scaling relations are extracted**. The practical reason for this is because the dominant frequencies are already captured with (very) high accuracy, as Fig.6.7 and Fig.6.8 shows. The predicted frequencies are nearly identical and only a slight amplitude overshoot is present. Thus there is no need to use different scaling relations, and the Strouhal-Reynolds relation is imposed on all modes, as in Section 6.2.3.

For the real-valued decay rates of the temporal coefficients also **no new scaling relations are extracted**. After visualising the decay rates in Fig.6.12, no useable and visible patterns emerged. Moreover, the values are of a very small magnitude  $O(10^{-7})$  and the impact of improved scaling relations is expected to be limited.

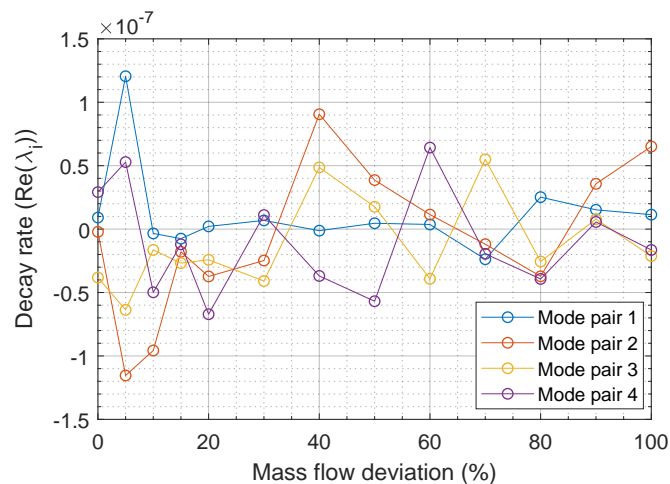


Figure 6.12: The decay rates of the leading modes

Theoretically, an advanced ML-algorithm can be used to extract patterns/scaling relations from the data. However this is not done and the possible benefit of scaling the decay rates remains unknown.

### 6.3.3. Spatial modes

The visualisation of spatial modes over a design space is previously done for POD spatial modes and shown in Fig.5.18. An important observation was that a common set of spatial modes does not exist. Thus, the spatial modes vary over the design space. The same phenomena still holds true for the sDMD algorithm. Figure 6.13 depicts the three spatial modes with the highest amplitudes for operating point 1, 8 and 13.

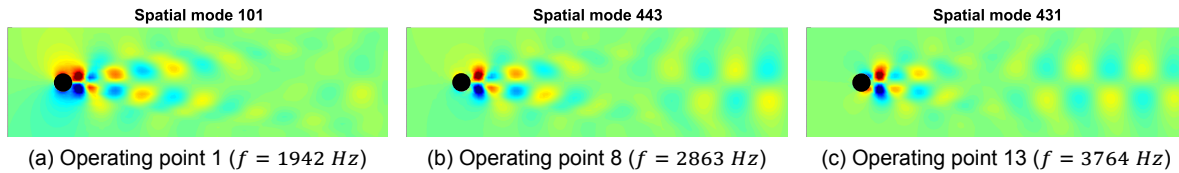


Figure 6.13: Spatial modes corresponding to the fundamental frequency

Figure 6.13 is example of similar looking spatial modes. These could hypothetically be interpolated, in a similar way to the mean-field interpolation (Section 5.2), with reasonable accuracy. However, the dissimilarities for the vast majority of spatial modes is simply too great that they be interpolated with reasonable accuracy. Moreover, it is very difficult to match modes that describe the same physics. Finding scaling relations for spatial modes remains non-trivial. Due to time-constraints and its inherent complexity the spatial modes are **unaltered**. The spatial modes from training point 1 are used for prediction of flow-fields.

### 6.3.4. Results

In this last subsection the results are shown for the sDMD-Scaling approach with partially self-made scaling relations. These are summarised as:

- **Mode amplitudes.** All modes are scaled with a separate linear regression method between training point 1 and 13.
- **Frequencies.** No new scaling relations are extracted. The frequencies of all temporal coefficients are still scaled with Strouhal-Reynolds relation.
- **Decay rates.** No scaling relation became visible from Fig.6.12. The decay rates from training point 1 are used.
- **Spatial modes.** No scaling relation could be found. The spatial modes from training point 1 are used.

The sDMD-Scaling approach with partially self-made scaling relations is denoted in Fig.6.14 with an asterisk. Figure 6.14 presents the MRE for the governing algorithms.

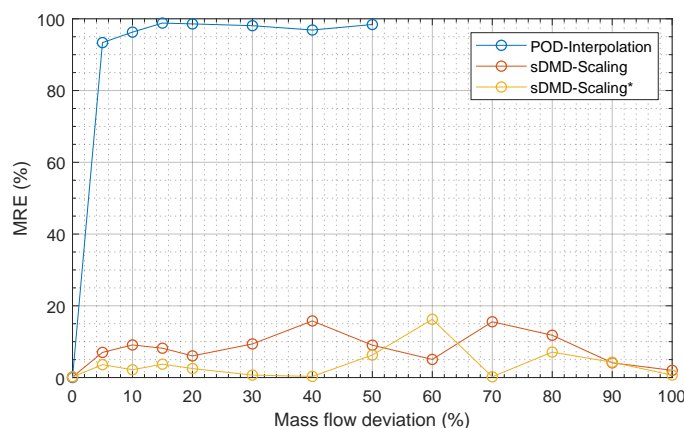


Figure 6.14: Mean Residual Error for the two algorithm and a variation with partially self-made scaling relations. The sDMD-Scaling approach with new scaling relations is denoted with an asterisk (\*).

Figure 6.14 is a key figure of the project and is extensively discussed in Section 6.4. The incremental error reduction: from a very high MRE (POD-Interpolation) to an acceptable MRE (with the proposed sDMD-Scaling approach), to even a lower MRE with self-made scaling relations, is encouraging. Below pressure figures are shown that depict the spectra of operating point 8. A comparison is drawn between the spectra of the predicted flow-fields with conventional and self-made scaling relations.

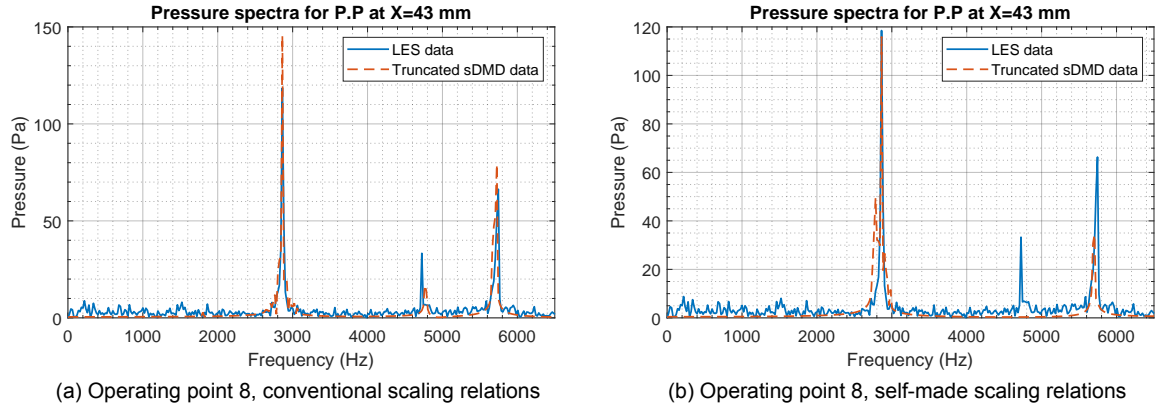


Figure 6.15: The pressure spectra for operating point 8, this point is representative for the other operating points.

From Fig.6.15 (b) it can be seen that the peak amplitudes are predicted with higher accuracy. Unlike, Fig.6.15 (a) where significant amplitude overshoots are observed. Secondly, an artificial peak is introduced left of the fundamental frequency (Fig.6.15 (b)) and two dominant frequencies between 4800-5800 Hertz are suppressed. These observations indicate that improvements are made with regards to mitigating the amplitude overshoot. Simultaneously, artificial peaks are introduced and harmonics are suppressed. These results prove the potential of the sDMD-Scaling algorithm to discover and use scaling relations. Simultaneously, the results clearly indicate that more work is required for the sDMD-Scaling algorithm to be used for complex application cases.

### 6.3.5. Potential speed-up

So speed-up times are not mentioned, instead in this document a strong emphasis is put on predicting physical accurate flow-fields. Furthermore, no speed-up times are given for the POD-Interpolation approach since it fails to predict accurate flow-fields. A comparison can be drawn between the conventional CFD route and the CFD-sDMD-Scaling combination, for the computation of the Von Karman design space. The CFD approach computes the 13 design points with LES simulations. Whereas, the CFD-sDMD-Scaling approach requires the training data of 2 LES simulations and 11 sDMD-Scaled computed design points. To predict the flow-fields of these 11 points, the partial self-made scaling relations are used. A speed-up factor of  $\approx 5.6$  is achieved when observing the total computational time. Any new design point (within the confines of the design space) can be simulated  $\approx 309$  faster with the trained sDMD-Scaled algorithm than with a LES simulation.

Table 6.1: Estimated times for every computational step and the cumulative (cum.) time for von Karman design space

	CFD			CFD-sDMD-Scaling			Unit
	Quantity	Time	Cum. time	Quantity	Time	Cum. time	
CFD	13	3	39	2	3	6	(hr)
Extraction				2	4	8	(s)
Prediction				11	1	11	(s)
Reconstruction				11	30	330	(s)
Total time			$\approx 39$			$\approx 7$	(hr)

Caution is always warranted when looking at time-estimates and speed-up factors, these times deviate for other application cases, hardware configurations and users.



## 6.4. Discussion results

In this section a discussion of the previous seen results is given. First a compact comparison is drawn between the 2 governing algorithms. Secondly, an analysis is given on the potential of the sDMD-Scaling algorithm.

### 6.4.1. POD-Interpolation vs sDMD-Scaling

This subsection augments the information that is given in Section 3.4.3 where key features of the POD and sDMD algorithm are compared. Key observations from this section are:

- **Sparsity.** Any POD algorithm outperforms a DMD algorithm, as proven by Lumley [86]. For a given number of modes a POD based approach provides the best representation of the input data compared to any other linear decomposition methods [86].
- **Truncation.** The POD algorithm allows for relative straightforward dimension reduction. The ranking of the modes enable a truncation based on the cumulative energy content. Whereas, the sDMD algorithm provides no ranking and only with the help of a penalty function (Eq.3.22) are the relevant modes identified.

Based on these observations and the reportedly spectacular results from Georgia Tech. [27, 28, 87, 88, 154, 165] was the POD-Interpolation the favourite algorithm for the prediction of flow-fields. Chapter 5 showed, however, significantly less promising results and 2 root-causes are identified (see Section 5.4). The verdict on the POD-Interpolation approach is clear: it is not suited for the prediction of flow-fields for different Reynolds numbers (see Section 5.5). Succeeding this are the advantages of the sDMD-Scaling algorithm.

- **Decomposition.** The biggest advantage of the sDMD algorithm are the periodic temporal coefficients. Periodic coefficients can be scaled and thereby avoiding any interpolation, which is a root-cause for the POD-Interpolation approach.
- **Additional learning benefit.** The DMD offers, simply said, more information. All the components are complex valued that provide for example information on the growth and decay rate.

At this point it should come as no surprise, but the sDMD-Scaling significantly outperforms the POD-Interpolation algorithm. The periodic coefficients can be scaled with relative high accuracy to predict the dynamics of objective points. Key instances where the sDMD-Scaling algorithm is better than the POD-Interpolation approach are:

- The fundamental vortex shedding frequency, the key dynamical feature of a von Karman Vortex Street is accurately predicted.
- The predicted fluctuating flow-field of the sDMD algorithm is physical, it does not disappear at time instances (i.e. Fig.6.5 (f,j)) unlike the predicted POD-Interpolation flow-field (Fig.5.20 (a,b)).
- The MRE (Fig.6.14) for the sDMD algorithm is significantly lower than the POD algorithm. An error reduction of  $\approx 5$  is achieved when using the known scaling relations. An error reduction of  $\approx 10$  is achieved when using self-made scaling relations.

Only the results, from both algorithms, can be compared for the von Karman case. Due to time-constraint it was decided to research the possibility of discovering and using new scaling relations, instead of applying the sDMD-Scaling algorithm to the Mixing Layer case. In the next section the potential of the sDMD-Scaling algorithm is analysed.

### 6.4.2. Potential

So far the sDMD-Scaling algorithm is only tested on 1 application case. If a ranking on the complexity of application cases would exist, surely a von Karman Vortex Street is near the bottom. Whereas, reactive flows in combustion chambers (every-day application cases at the Siemens R&D Combustion department) is at the ceiling of the scale.

For the relative straightforward von Karman Vortex Street it is shown that the POD-Interpolation (and thus the CPOD-Kriging and KSPOD-Kriging) approach is not able to accurately predict flow-fields for variations in flow parameters. The proposed sDMD-Scaling algorithm significantly outperforms the POD-Interpolation algorithm, and most likely the other aforementioned algorithms. Interpolation (i.e. Kriging) of temporal coefficients is a root-cause and shown to produce inaccurate results.

However these results are still limited to the von Karman case. Whereas, the vision is to use ML-algorithms for optimisation studies of critical components with complex flows and complex geometries. For example, shown in Fig.6.16 (b).

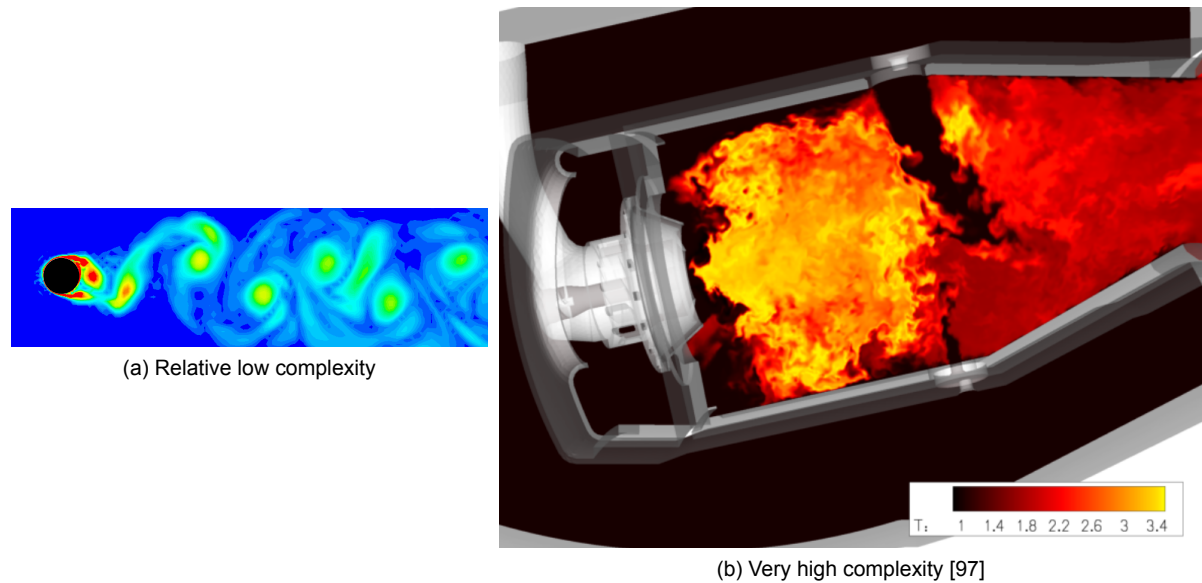


Figure 6.16: Two representative applications cases. The von Karman Vortex street (a) and a reactive flow in a combustion chamber (b).

A significant limitation of the sDMD-Scaling algorithm is that it needs scaling relations. Complex application rarely have known scaling relations and first must be discovered. In this work a methodology is proposed to discover scaling relations for predictions of flow-fields. In Section 6.3 this methodology is applied and for the von Karman street a substantial decrease in the MRE is achieved, compared with the conventional scaling relations. Simultaneously, new challenges presented themselves: 1) only new relations could be extracted for the mode amplitudes, 2) artificial peaks are introduced in the spectra and 3) dominant frequencies are suppressed. This algorithm needs to overcome plenty of challenges before it can ever be implemented on a case as Fig.6.16 (b). Chapter 6 proved that the sDMD-Scaling has potential and that it outperforms a reportedly spectacular algorithm.

The true extent of the sDMD-Scaling potential is still unknown, since it has only been implemented and tested on 1 application case. It can safely be stated that the challenge posed by optimisations studies with complex varying physics and complex varying geometries is **extremely high**. It remains to be seen whether the sDMD-Scaling algorithm, or for that matter, any ML algorithm can predict accurate flow-fields for these complexities.



## 6.5. Conclusions Sparse DMD-Scaling

Chapter 6 revolves around the sDMD-Scaling algorithm and intended to describe 4 points. The first point is regarding the methodology, which is nearly identical to the POD-Interpolation approach. The key difference is that the prediction is done by scaling the periodic temporal coefficients, instead of interpolation.

The second point of Chapter 6 provides the results. For this the Von Karman Vortex Street is used, to test the efficacy of the sDMD-Scaling algorithm. The frequencies of temporal coefficients are scaled with the Strouhal-Reynolds relation. Whereas, the mode amplitudes are scaled with a relation derived from the Bernoulli equation. These initial scaling relations proved to be effective as the vortex shedding behaviour and frequency are accurately predicted. An error reduction of  $\approx 5$  is achieved compared to the POD-Interpolation approach and the MRE ranges between 5 – 17%.

Application cases rarely have known scaling relations and thereby limiting the applicability of the sDMD-Scaling algorithm. In this work a methodology is proposed (which addresses point 3 of Chapter 6) that enables the discovery of scaling relations for the prediction of flow-fields. This is again tested on the Von Karman Vortex Street, to determine whether its possible to improve on the aforementioned scaling relations. An incremental error reduction of  $\approx 5$  is achieved compared to the conventional scaling relations. Simultaneously, new challenges presented themselves (e.g. artificial peaks) that need to be overcome before the algorithm can tackle application cases of higher complexity. Regarding speed-up potential, the computation of the von Karman design space can be performed  $\approx 5.6$  faster with the CFD-sDMD-Scaling approach than with LES simulations. New design points can be simulated  $\approx 309$  faster with the trained sDMD-Algorithm than with a LES simulation.

Lastly, the main goal of Chapter 6 is to provide a verdict on the CFD and sDMD-Scaling combination. Shown in this chapter is that it is possible to combine CFD and a ML algorithm to accurately predict flow-fields and simultaneously meet the goals of this thesis (see Section 1.2). But the true potential of the proposed sDMD-Scaling algorithm is still to a large degree unknown, as it is only tested on 1 canonical test case. Future work is required to determine the extent of the potential.



# 7

## Conclusions and recommendations

In this work, the combination of CFD simulations and Machine Learning (ML) algorithms is researched to simultaneously: 1) reduce computational cost and time, 2) retain physical insight by focusing on the prediction of flow-fields and 3) keep the ability to access information or to make adjustments. Chapter 7 provides the main conclusions based on the results, answers the research questions and provides recommendations for future work.

### 7.1. Conclusions

In Chapter 1, multiple research questions were posed that have guided the development of this project. The governing research question of this project is:

- Can ML-algorithms be combined with CFD simulations in such a way that they reduce the overall computational cost and time of the exploration of a design space? The product of interest are the full flow-fields for every explored design point.

Before this governing research question is addressed, each upcoming subsection is dedicated to a separate research question. Furthermore, an additional subsection is dedicated to a conclusion that was drawn throughout this work, but never was included in the original research questions.

#### 7.1.1. Machine Learning algorithms

- What ML-algorithms are suited for this goal?

One of the key results of the literature study (Chapter 3) is the identification, listing and judging of 19 potential ML-algorithms. Algorithms based on a type of Neural Network show promise but require excessive amounts of training data. Regression algorithms are sparse and did not show significant promise. The most suitable algorithm is chosen with the help from 4 criteria, that is the CPOD-Kriging algorithm. In studied literature it reportedly provided very high speed-up times, good accuracy, additional learning benefit and is relative simple. This work simplified this algorithm and it is referred to as the POD-Interpolation algorithm. This simplification is valid because: 1) no geometrical parameters are varied and thus no data from different meshes are used, 2) the Kriging algorithm reduces to a standard linear interpolation method when interpolating between two points.

#### **Prediction of flow-fields using the POD-Interpolation algorithm**

The POD-Interpolation algorithm (described in Chapter 5) is applied on 2 test cases: a Von Karman Vortex Street and Mixing Layer. The design parameters for the test cases are respectively the mass flow and velocity at the inlet. From these test cases, it can be concluded that the POD-Interpolation algorithm is unable to predict physically accurate flow-fields. Small variations in the design parameter results in very high MRE's of +95%. Key instances of where this approach fails are: 1) the inability to predict the fundamental frequency, 2) the disappearing of flow-fields, 3) failure to accurately predict vorticity dynamics and 4) failure to accurately predict the downstream wake.

Two root-causes are identified, the first is an invalid assumption. The approach assumes a common set of spatial modes, thus all spatial modes over the design space should be (nearly) identical. This work shows that this assumption is invalid for varying flow parameters. This failed assumption complicates the flow-field reconstruction aspect, since it is unclear on which set of spatial modes to impose the newly obtained temporal coefficients on. Furthermore, varying spatial modes over the design space indicates that modes from different training points capture different physics from the flow-field. This is problematic because these modes that describe different physics are interpolated to predict flow-fields at objective points.

The second root-cause stems from interpolation of the temporal coefficients. Interpolation in the time and frequency domain results in inaccurate new temporal coefficients due to phase and frequency differences between coefficient signals. Constructive and destructive interference results in new coefficient signals that exhibit nonphysical behaviour. With interpolation it is impossible to predict new dominant frequencies, because it merely projects the dominant frequencies of the training cases with lower amplitude.

### Prediction of flow-fields using the sDMD-Scaling algorithm

With the inability of the POD-Interpolation approach to predict physical accurate flow-fields, the proposed algorithm sDMD-Scaling algorithm is implemented (described in Chapter 6). This method relies on accurate scaling relations to predict flow-fields. The Von Karman Vortex Street case lends itself for scaling, due to the availability of the Strouhal-Reynolds relation. The sDMD-Scaling algorithm predicted relative accurate flow-fields and, more importantly, did not exhibit any inaccurate behaviour. The vortex shedding frequency and important harmonics are accurately predicted. The MRE of the predicted flow-fields is predominantly below 10%, for 3 objective points that ranged between 10 – 17%. Using the sDMD-Scaling algorithm, an error reduction of  $\approx 5$  is achieved when compared to the POD-Interpolation approach.

Application cases have rarely known scaling relations. The sDMD-Scaling algorithm provides a potential route to: 1) provide additional learning benefit, and 2) predict flow-fields by discovering and using new scaling relations. Patterns or scaling relations can become visible when visualising the modes over the design space. Subsequently, possible patterns can be extracted with algorithms (e.g. interpolation, NN, regression). This approach is implemented on the Von Karman case to determine whether the prediction of the flow-fields can be improved, with respect to the conventional scaling relations. With partially self-made scaling relations an error reduction of  $\approx 10$  is achieved when compared to the POD-Interpolation approach. The prediction of the fluctuating flow-field rarely exceeds a MRE 7%. As a concluding remark, after implementation and analysis of 2 algorithms, only the proposed sDMD-Scaling algorithm has proven its worth for predicting flow-fields.

## 7.1.2. Assumptions

- When making assumptions, what are the physical implications?

Five governing assumptions are made throughout the process of predicting flow-fields.

1. **Quasi 2-D CFD.** This results in an unrealistic smooth flow-field compared to its 3-D counterpart.
2. **Separation of variables.** Both the Snapshot-POD and the sDMD algorithms separate space and time which result in spatial only modes. Turbulence is a spatial-temporal phenomena and this raises doubts on the validity of a spatial-only approach.
3. **Common set of spatial modes.** This work shows that no common set of spatial modes exists for both the POD-Interpolation and sDMD-Scaling approach. This invalid assumption complicates flow-field prediction and reconstruction.
4. **Modulation of components.** The thesis relies on the assumption that it is possible to predict flow-fields by modulating the temporal coefficients and mode amplitudes. Strictly speaking, this is incorrect since modes vary over the design space.
5. **Scaling of modes.** Singular scaling factors are imposed on all temporal coefficients and mode amplitudes. In Section 6.3 this assumption is shown to be invalid since scaling factors change for different modes. This faulty assumption is partially mitigated with self-made scaling relations.

More assumptions were made (e.g. incompressible flows) but these were off less importance.

### 7.1.3. Potential speed-up times

- What speed-up times can theoretically be achieved?

Only the CFD-sDMD-Scaling approach allows for a speed-up time comparison since it is capable of predicting flow-fields. The computational time is compared of the exploration of the Von Karman design space with an CFD and CFD-sDMD-Scaling approach. A speed-up factor of  $\approx 5.6$  is achieved for this specific case. New design points (within the confines of the design space) can be computed  $\approx 309$  faster with the trained sDMD-Scaling algorithm than with a LES simulation.

### 7.1.4. Accuracy losses

- What is the magnitude of the accuracy losses adjoint to this approach?

The POD-Interpolation approach fails to predict physical flow-fields and results in a very high MRE, +95% (Fig.5.12), for small deviations in the flow parameter. The sDMD-Scaling algorithm significantly improves in this regard and never exceeds 17%. With self-made scaling relations the MRE is for 10 out of 11 objective points below 7%.

### 7.1.5. Limitations

- What are the limitations of the CFD and ML combination and can they be overcome?

The POD-Interpolation (but with high likelihood also the CPOD-Kriging and the KSPOD-Kriging) fails to predict accurate flow-field for varying flow parameters. These methods are inherently flawed since interpolation produces inaccurate results due to constructive and destructive inference. This is a fundamental problem in signal analysis and physics, no realistic solutions were found to overcome this.

The sDMD-Scaling has shown potential but its prediction capabilities is constrained by the range and quality of the scaling relations. More work must be performed on discovering and using self-made relations to determine the extend of this limitation.

### 7.1.6. Error quantification

This additional subsection is dedicated to a conclusion that was drawn throughout the development of this work, but never was included in the original research questions. The question is defined as follows:

- How can differences between CFD results and approximated CFD-ML results be compared and quantified?

When comparing 2 flow-fields with phase differences it is not possible to use a single (all-encompassing) error quantification method, because it simply does not exist. Multiple methods need to be combined to objectively determine the quality of the predicted flow-field. This work used the following qualitative and quantitative methods:

1. **Visual inspection.** Quickly gives a global understanding of the predicted flow-field quality.
2. **Figures of interest.** Local representative information probes are distributed throughout the domain. These determine how accurately the CFD and ML combination is able to predict the dynamics of the exact CFD result. Typical examples are temporal pressure and spectra figures.
3. **Error quantification.** Using a MRE (Eq.D.1) and RMS (Eq.D.2) method to quantify the discrepancies.

A significant portion of this research is dedicated to the CPOD-Kriging derived framework, that reportedly provided spectacular results. After close examination, it became obvious that the researchers postulating the CPOD-Kriging and KSPOD-Kriging algorithm, creatively used different error quantification methods to partially hide the discrepancies and boost their results.

### 7.1.7. Main conclusion

- Can ML-algorithms be combined with CFD simulations in such a way that they reduce the overall computational cost and time of the exploration of a design space? The product of interest is the full flow-fields for every explored design point.

This thesis proved that for the Von Karman Vortex Street design space it is possible combine to CFD and Machine Learning to:

1. **Reduce computational cost and time.** The exploration of the design space is performed  $\approx 5.6$  times faster using the CFD-sDMD-Scaling approach than with conventional LES simulations.
2. **Retain physical insight.** By being able to predict relative accurate flow-fields, the MRE never exceeds 17% and with self-made scaling relations rarely exceeds 7%
3. **Access information.** Due to the significant dimension reduction of the computed data, it becomes tractable to store, modulate and access information.
4. **Make quick adjustments.** New objective points are computed within approximately 35 seconds, thus  $\approx 309$  times faster with the trained sDMD-Scaling algorithm than with a LES simulation.

Simultaneously, a critical remark is appropriate. The sDMD-Scaling algorithm proved to be successful for a canonical fluid dynamic test case. But given the difficulties and efforts that went into this relative straightforward case, it is difficult to foresee any large-scale implementation of any CFD and ML combination for predicting flow-fields.

## 7.2. Recommendations

In this last section, the recommendations for future work are given. Section 7.2.1 is directed towards the purpose of predicting flow-fields by combining CFD and Machine Learning. Subsection 7.2.2 provides a wider discussion on combining CFD and Machine Learning algorithms for other prediction purposes.

### 7.2.1. Future work

The following recommendations are made regarding future work.

- **Extending the sDMD-Scaling.** In this thesis, the proposed sDMD-Scaling has shown enough potential with the relatively accurate predicted flow-fields that it warrants extra attention. A significant limitation is the need for the scaling relation that most applications cases do not have. More work must be performed to truly determine whether it is possible to discover scaling relations for complex application cases.
- **Parameterized Reduced-Order Models.** According to the author pROMs that employ different manifold interpolation techniques are the only category of algorithms that require extra attention. pROMs articles tend to be very mathematical and no superior pROM has emerged yet, but these algorithms are designed for a similar purpose to the goal of this thesis. These algorithms might lead to new insights and possible solutions.
- **Varying geometrical design parameters.** During this thesis, the decision was made to focus on varying flow parameters (i.e. mass-flow, velocity) to avoid the challenges posed by varying geometrical parameters. Future work must include overcoming these challenges. The interested reader is directed towards the articles proposed by Mak et al. [88] and Bhatnagar et al. [14] as these articles can provide possible solutions.
- **Programming language of software.** The current code (i.e. MatLab) needs to be converted to either Python or something equivalent. These software packages are more suitable in handling large quantities of data and complex meshes, potentially reducing computational cost and time.
- **Prioritise learning by removing constraints.** There is a direct need to reduce the computational cost and time for optimisation studies that involve numerous LES simulations. By attempting to overcome this, you commit to significantly reducing the number of LES simulations and indirectly constrain yourself to a few potential ML algorithms.

Change the priority to learning and remove the limit on training data by possibly: 1) accepting large computational cost and time, or 2) switching to RANS or URANS based optimisation studies. This way, different types of ML-algorithms can relatively quickly be tested. Aside from the direct results, the user or company becomes increasingly familiar with the potential and limits of Machine Learning. Moreover, with the increased knowledge, different use-cases become obvious and create value for their respective goal.

Machine Learning in general, provides an almost naive approach to solving or learning about very complex physical problems. Machine learning has the potential to really augment the current CFD packages. Therefore, the author would like to encourage Siemens and others to keep researching the potential of these algorithms.

### 7.2.2. Discussion on CFD and Machine Learning

Following up on the previous recommendations is a discussion on the added value of combining CFD and ML, but not towards the prediction of flow-fields. This discussion is based on a year-long research into the niche field of combining CFD and ML. The intention of this is to provide a critical and thought-provoking discussion to (possibly) steer future work.

Computing flow-fields is generally complicated. For the foreseeable future, any CFD and ML combination is still too underdeveloped to be competitive with current CFD simulations. This statement is based on the following 2 factors.

1. **Complexity of relevant application cases.** This thesis intended to test ML-algorithms on typical Siemens application cases (e.g. mixing of fuel and air, reactive flows). The complexity of these cases are however extremely high. To showcase: 1) the physics behind these cases are not always understood, 2) the CFD for these cases remains a challenge, and 3) the validation with experimental measurements do not always match. This thesis shows that even relative straightforward cases are not trivial to predict with ML. A significant amount of work needs to be done before it can be implemented to complex cases.
2. **Nonlinearity of the physics.** Flow-fields and flow phenomena can react very strongly to small variations in design parameters. Capturing complex nonlinear physics requires more than a linear approach (i.e. POD or DMD). Moreover, overcoming the challenges posed by different meshes from geometrical parameter variations are going to be challenging.

The complexity of computing flow-fields will hardly change in the future, ML-algorithms might become sophisticated enough that they are able to accurately predict flow-fields. However, according to the author there are more suited goals on short- and long-term basis where ML can contribute to.

- **Objective-oriented optimisation.** Instead of focusing on predicting flow-fields, ML can predict quantities of interest in a design space. For example, a ML-algorithm that predicts the mixedness of fuel and air for different fuel nozzle configurations. When the "best design" is identified, using the algorithm, it can be validated with a LES simulation. Whether this can be competitive against the current algorithms (e.g. SHERPA, GA) remains to be seen.
- **Flow control.** This field largely revolves around identifying flow regimes or instabilities. Identification and classification are key strong points of ML. Real-time flow control is still a vision and not a reality, this is however a route worth pursuing. Flow control is discussed in Section 3.2.
- **Identification of hidden physics.** Its added value is possibly best described with a classic analogy. Sir Isaac Newton (i.e. ML-algorithm) postulated the law of gravity due to a falling apple (i.e. large data-sets). ML-algorithm enable the identification of complex physics that are not obvious for researchers.
- **Turbulence modelling.** By leveraging the data from DNS and experiments to build data-driven closure models. Research on this front is ongoing but already performance gains are achieved over competing state-of-the-art methods [77].

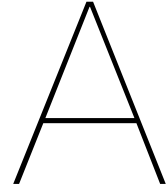


### **Concluding remark**

Machine Learning is a promising field that can supplement the current computational tools and engineers. But a transformation (as the hype sometimes suggests) of the current working principles is difficult to imagine in the foreseeable future.

Augmenting rather than replacing is the way forward.





## Appendices - Chapter 2

### A.1. Reynolds-Averaged Navier-Stokes

Engineers are typically interested in knowing a few governing flow quantities, for example the average force on a object, flow velocity or the amount of substance that has mixed [45]. RANS is computationally significantly less demanding than a LES, and is able to deliver these average properties with reasonable accuracy [45]. Its efficiency (accuracy versus computational cost) is the primary reason why RANS is the most used method to solve the NS equations. The RANS approach to fluctuating components such as turbulence is that all the unsteadiness of the flow is averaged out. Figure A.1 properly depicts this, all the fluctuating parts are averaged out over an appropriate time interval ( $T$ )

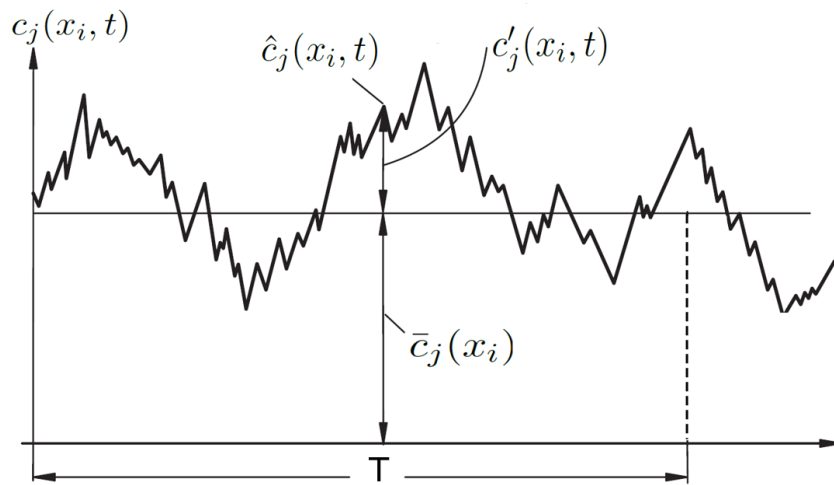


Figure A.1: Temporal development of a turbulent flow field. Obtained from the work of Durst et al. [41]

If the flow is unsteady, time averaging is not adequate and must be replaced by ensemble averaging. Ensemble averaging is similar in its approach and the number of members of the ensemble must be large enough to average out the effects of fluctuations [158]. For more information on the different available averaging methods the reader is directed towards the work of Ferziger [45] or Wilcox [158]. Depicted in FigA.1 are different components of a fluid quantity, every fluid quantity can be written as the sum of a time-averaged part ( $\bar{c}_j(x_i)$ ) and a fluctuating part ( $c'_j(x_i, t)$ ). To obtain an averaged value an appropriate time interval, ( $T$ ), must be chosen for which it can be integrated:

$$\bar{c}_j(x_i) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \hat{c}_j(x_i, t) dt \quad (\text{A.1})$$

With this the RANS equations can be derived with the Eq. 2.1 and Eq. 2.2.

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial (\bar{\rho} c_i)}{\partial x_i} = 0 \quad (\text{A.2})$$

$$\bar{\rho} \left( \frac{\partial \bar{c}_j}{\partial t} + c_i \frac{\partial \bar{c}_j}{\partial x_i} \right) = - \frac{\partial \bar{p}}{\partial x_j} - \frac{\partial}{\partial x_i^2} \left( T_{ij} - \bar{\rho} c'_i c'_j \right) + \bar{\rho} \bar{g}_j \quad (\text{A.3})$$

Figure A.2, obtained from the work of Schwarze [129], graphically depicts a RANS solution for an idealised turbulent jet. Unlike Fig. A.2 (a) which depicts a chaotic turbulent flow, Fig. A.2 (b) depicts the computed RANS averaged, smooth and orderly "turbulent" flow.

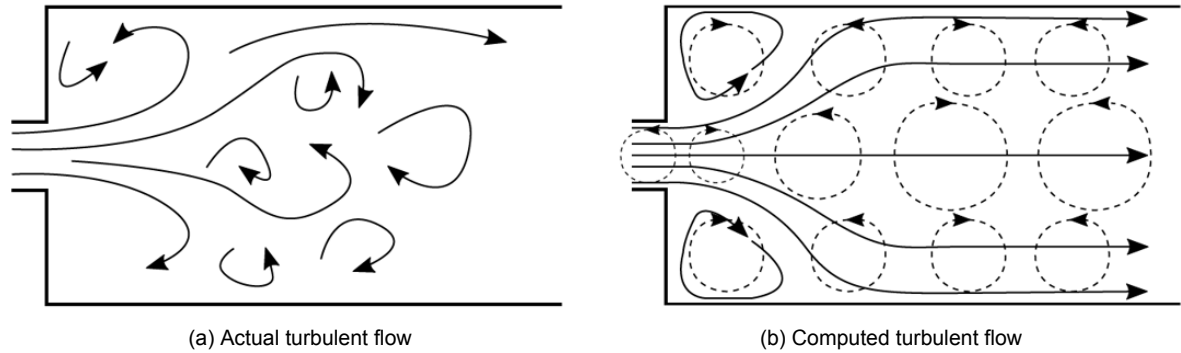


Figure A.2: Comparison between the actual turbulent flow and the approximation computed by a RANS CFD solver. Obtained from the work of Durst [41].

The last pressing topic within the RANS framework is how to model the turbulent stress tensor ( $\overline{\rho c'_i c'_j}$ ), known as the Reynolds stress tensor (Eq.A.2). This additional term arises from Reynolds-averaging [158] the non-linear convective term from the NS-equation (Eq.2.4). The tensor is symmetric and has six independent and unknown components. To be able to solve the RANS-equations, the system of equations needs to be closed and additional equations need to be found to solve for all the unknowns. The closing of ( $\overline{\rho c'_i c'_j}$ ), seen in Eq.A.2 is typically referred to as the Reynolds closure problem.

To be able to close and thus solve the RANS-equations, turbulence models are used. The function of a turbulence model is to come up with approximations for the unknown components using the known flow properties at that point [158]. For the sake of brevity, turbulence models are not extensively discussed in this document and the interested reader is directed towards [45, 103, 158]. The three most used turbulence models in RANS are  $k - \epsilon$ ,  $k - \omega$  and the hybrid Shear Stress Transport (SST)  $k - \omega$  model. They are roughly summarised as follows.

- The  $k - \epsilon$  is applicable to many turbulent flow problems [103] but has difficulties describing shear stresses and boundary layers near a wall.
- The  $k - \omega$  is significantly better in accurately solving boundary layers even under adverse pressure gradients. The biggest disadvantage of the  $k - \omega$  is that it is extremely sensitive to inlet boundary conditions for internal flows [139].
- The hybrid SST model blends the advantages of two aforementioned turbulence models but against increased computational cost [139].

For all RANS computations, performed during this thesis, the  $k - \epsilon$  model was used. The predominant reason for this is because RANS is used as an initialisation tool for the subsequent high-fidelity CFD solver, towards this purpose  $k - \epsilon$  is deemed adequate enough.

# B

## Appendices - Chapter 3

This chapter provides additional information regarding the Design of Experiments and Machine Learning. Furthermore, 2 additional Modal Decomposition Methods are described.

### B.1. Design of Experiments

In Chapter 1.2 the concepts of multi-dimensional design spaces, exploration of a design space and design points have been briefly discussed. Furthermore, it was stated in Chapter 1.2 that statistical algorithms are used to minimise the number of design points but simultaneously provide the best possible representation of the multi-dimensional design space. This is especially important for optimisation studies where every design point represents a computationally expensive CFD simulation. This statistical algorithm refers to a Design of Experiments (DoE) method and its underlying working principle. A DoE allows the user to efficiently explore a design space and understand the impact of design parameter variations on the output. A DoE approach is significantly quicker in understanding a process than for example a One-Variable-at-a-Time approach, which is moreover heavily dependant on guesswork, luck, experience and intuition for its success [4]. Since the first proposal of a DoE method by R. Fisher [47] in 1935, the DoE approach has become a well-established tool which aids in optimisation studies, robust design- and sensitivity analysis. Regarding its implementation, a DoE method is commonly the first step in a optimisation study and used to guide the choice of the design points. Subsequently, other optimisation methods (e.g. response-surface models, multi-objective pareto front) follow the DoE and making use of the gathered data at the chosen design points. The need for a DoE method can be best emphasised with the help of an example, similar to the Example 1 seen in Chapter 1 but with added details.

#### Example 2

The design of fuel-air nozzles for a new combustion chamber have to be optimised, an optimised design results in lower emissions due to better mixing between the fuel and air. Suppose, every fuel nozzle has three arbitrary design parameters ( $r$ ). Each design parameter can account for 10 different values ( $n$ ). With the help of statistics Eq.B.1 provides us with the amount of permutations. Permutations are nearly identical to combinations, but differs by having an additional constraint which states that the order of the variables must be preserved.

$$\text{Permutations} = \frac{n!}{(n-r)!} = \frac{10!}{(10-3)!} = 720 \quad (\text{B.1})$$

Three design parameters, each which accounts for 10 different values, results in 720 permutations. This would imply if no DoE is used, the full 720 design points have to be computed via CFD simulations. Needless to say, to simulate all these points is intractable, highly inefficient and therefore DoE methods are implemented in optimisation studies. Depending on which DoE method is used the number of the simulations can be significantly reduced [26]. As an initial estimate for the number of simulations, a rule of thumb [58, 87] can be

used when implementing a DoE. The rule of thumb suggests it is sufficient to run 10 simulations per design parameter. This is relative sparse sampling but for the example above this implies that it is theoretically possible to obtain the same order of exploration with 30 CFD simulations as the full 720. This is however a theoretical and idealised example but it illustrates the strength of DoE methods.

The DoE approach is well-established and at the present-day many different types of DoE exist which are described throughout literature [4, 26]. The approaches differ in their algorithms and how well they represent a design space for a limited number of design points. The full factorial methods are among the most commonly used experimental design methods, [4]. This method consists of all possible combinations of levels for all factors where the number of levels  $n$  describe the amount of values per parameter, which are investigated. The number of inputs is defined as  $r$ , from this the total number of experiments/simulations is determined as  $n^r$ . The advantage of full factorial designs is that they make efficient use of data and do not confound the effects of parameters, so that it is possible to evaluate the main and the interaction effects clearly. The disadvantage is that the number of the required experiment rises exponentially, becoming quickly too large for use.

## B.2. A brief history of Machine Learning

To write on the history ML is somewhat premature since it has really been around for a few decades and its widespread implementation even shorter. To trace back the birth of ML is contested, one can go back to the fundamental mathematical works of Bayes[9], Legendre[81] and Markov[91]. For example, Legendre proposed the first and the most simple type of ML which is a linear regression through the least-squares approach [81]. Others suggest to trace it back to the birth of computers, where the first-ever electro-mechanical binary programmable computer was created by Konrad Zuse in 1938 [172]. However, the first "real" notion of ML was postulated by the neurophysiologist Warren McCulloch and the mathematician Walter Pitts who wrote a paper [95] on neurons and recreated their function with an electrical circuit. With this paper and electrical circuit the first Neural Network (NN) was born and seen by many as the birth of ML [30]. A few years later (1950), the theory was put into practice, because of advances in computational hardware and software and it became possible to simulate a hypothetical NN [140]. In 1960 the first neural network, which goes by the name of MADALINE [156], was applied to a real world problem. MADALINE basically acts as a filter which eliminates echoes on phone lines and is currently still in use [140]. Around the same time, the Perceptron algorithm [122] was invented by F.Rosenblatt which can be seen as the first ML classification algorithm which approximates an output based on a linear function with a set of weights. Fast forward a few years, in 1969 M.Minsky (founder of the MIT AI lab) and S.Papert ushered in the AI Winter by publishing a book [96] which expressed their scepticism in the form of thorough analysis of the limitations of Perceptrons. Followed by the Lighthill report [101] report in 1973 which gave a very sceptic diagnosis of many core aspects of ML.

The Lighthill and Perceptrons report deflated the expectations and hype which resulted in the almost complete stop of work involving ML during 1970 and 1996. The ML enthusiasm was reinvigorated in 1997 when a computer (called DeepBlue) running ML algorithms beat the world champion G.Kasparov in a game of chess. This can be seen as the first point-in-time where ML outperforms a very skilled human in a complex activity. In 2006 again a boost was given to the ML field with the introduction of Deep Learning (typically referred to as a Deep Neural Network (DNN)), by Hinton et al. [59]. Hinton transformed the ML field in a sense that today its being implemented by the biggest companies and currently considered the most important and promising ML algorithm [30, 42]. With the introduction of a DNN the (very) brief historical overview of ML is finished. This section purpose is to add a historical perspective on the field of ML.

### Current impact of Machine Learning

The potential annual impact of ML algorithms is estimated by McKinsey [30] between \$9.5 and \$15.4 trillion in the global economy. McKinsey [30] states that the potential value of merely Deep Learning algorithms in the industry, aerospace and defence, with the smallest potential value could enable the annual creation equivalent to the GDP of Lebanon. International Data Corporation (IDC) [63] predicts that that in four years (2018-2022) the spending on cognitive and artificial intelligence (AI) systems, both heavily rely on ML-algorithms, will reach \$77.6 billion which is an increase of 323%.

### B.3. Additional Modal Decomposition Methods

This section provides the brief description of 2 MDM. These MDM were at one point considered to be used in combination with an interpolation technique.

#### Spectral POD

In this section the frequency domain form of the POD is described, otherwise called the Spectral POD (SPOD). This approach is similar to the classical POD proposed by Lumley, but is different in the sense that it involves decomposition of the cross-spectral density tensor. The fundamental difference between the space-only form and the spectral form is that both methods identify coherent structures but the SPOD modes evolve coherently in space and time, [148]. This implies that the SPOD method is better suited for identifying spatio-temporal coherent structures. This is a reason why also the SPOD method has been applied to a variety of fluid problems including boundary layers [149], mixing layers [17, 39] and the flow around an airfoil [1].

Regarding mathematics, obtaining modes depended both on space and time is now accomplished by defining a stochastic ensemble to consist of a collection of realisations of the time-dependent flow. For example, different runs of the same experiments are considered to be realisations of a stochastic proces. Similar to the two previous approaches, a quantity  $\lambda$  needs to be maximised which is defined as

$$\lambda = \frac{E\{|\langle u_i(\mathbf{x}, t), \phi(\mathbf{x}, t) \rangle_{x,t}|^2\}}{\langle \phi(\mathbf{x}, t), \phi(\mathbf{x}, t) \rangle_{x,t}} \quad (\text{B.2})$$

and leads to the spectral eigenvalue problem

$$\int \int \int_{space} \mathbf{S}(x, x', f) \mathbf{W}(\mathbf{x}') \psi(\mathbf{x}', f') d\mathbf{x}' = \lambda(f') \psi(\mathbf{x}, f) \quad (\text{B.3})$$

This method starts to diverge from the previous methods with the preceding equation. Where normally there was a correlation tensor  $R$  now there is a cross-spectral density tensor  $S$ , which defined as the Fourier transform pair of the  $R$  tensor. The tensor  $S$  is nuclear<sup>1</sup> and the SPOD modes are its principal components. The Fourier modes of each flow is now optimally reconstructed as

$$\hat{q}(\mathbf{x}, f) = \sum_j^{\infty} a_j(f) \psi_j(\mathbf{x}, f). \quad (\text{B.4})$$

Only the most vital equations are given for this methods due to the complex derivation. For the interested reader, see the full derivation of these equations in Appendix A [148]. The SPOD modes each oscillate at a single frequency and optimally represents the second order space-time flow statistics. In contrast to space-only POD modes, which lose any concept of temporal ordering thus the time-dependent evolution of the flow has zero impact on the definition on the POD modes [109, 148]. Another way of expressing this, is that a POD mode describing a part of a flow is not necessarily correlated with the part of the flow described by the same POD mode at a later time. The weakness of the space-only POD approach is the strength of the SPOD approach. Thus there are clear benefits in using a SPOD method, however there are also some drawbacks. The SPOD method is more complex because it requires Fourier transforms with all of its inherent complexities. Secondly, with the implementation, techniques for computing SPOD modes are not properly documented and available.

<sup>1</sup>  $\|\mathbf{S}\| = \sum_{k=1} \sigma_k(\mathbf{S})$ , where  $\sigma_k$  is defined as the largest singular value of a tensor[167]

## Balanced Proper Orthogonal decomposition

The balanced POD (BPOD) is not included in the previous section since it is combination between the *balanced truncation* and the *snapshot POD* method. As the BPOD is a method for reduced order modelling for linear dynamical systems and only recently seen an expansion to nonlinear systems [7, 78] this method is described briefly. This method is however included for the reason that it shows great promise [64, 123, 130] compared to POD method and is currently extensively researched. Part of the motivation behind the BPOD method is that it tries to improve on the limitations of the aforementioned POD method. POD modes are very effective (indeed optimal) in approximating a given data-set, they are not necessarily the best modes for describing the governing dynamics which generate that data-set. Sometimes low-order modes maybe critical for the dynamics, for example fluid problems in which acoustic resonances occur. Due to their low variance, these modes will be truncated and therefore important information is lost. The BPOD counters this problem by having two types of modes, balancing and adjoint modes, which have to be balanced. Two paragraphs below this be further discussed.

The POD aspect is introduced in the BPOD method because the *balanced truncation* by itself is computationally intractable. As an example, assume there are  $n = 10^6$  DoF/Grid Points. Due to the Gramians<sup>2</sup>, it requires close to 3.6 terabytes of storage memory [124]. Therefore, the POD method is required to provide an approximation to the balanced truncation method and making it tractable for high-dimensional systems. The *balanced truncation* method was first proposed by Moore [98] and has become a standard method in the Control Theory field. The BPOD unsurprisingly has many similarities with a POD method, both produce a set of modes which describe coherent structures in a flow. Also the computations are in many regards similar.

The BPOD is a modal decomposition method which extracts two sets of modes referred to as the balancing and adjoint modes. The two set of modes are ranked by two properties, controllability and observability. Terms which originally stem from the field of Control Theory. The most controllable state represents the most easily excited input, whereas the most observable state represent the largest corresponding excited output. The dimension reduction is again achieved by reducing the original problem down to the the most important, thus the most controllable and observable, modes. The difficulty is that for non-normal<sup>3</sup> systems the balancing between the two is complex since either they have high controllability or vice versa. Balancing involves appropriately choosing a coordinate system in which the controllability and observability align. For both the POD and the BPOD a fluid flow can be reconstructed as

$$\mathbf{q}(\mathbf{x}, t) = \sum_{j=1}^n a_j(t) \phi_j(\mathbf{x}) \quad (\text{B.5})$$

where  $\phi_j$  are the balancing and adjoining modes. For POD the modes are orthonormal whereas for BPOD the modes are biorthogonal which means they satisfy  $\langle \phi_j, \psi_k \rangle = \sigma_{jk}$  through which  $a_j$  is determined by  $a_j(t) = \langle \mathbf{q}, \psi_j \rangle$ . The mathematics behind the BPOD is kept intentionally very brief, the interested reader is directed towards the work of Rowley [123]. The last thing which remains to be explored is the strength and weakness of the BPOD methods. The summation of both are provided in the lists below.

### Strengths:

- By having two types of modes, it adds a control parameter which results in better truncation of the governing dynamics.
- The balanced POD excels in capturing the dynamics of non-normal systems. The BPOD is able to capture small energy fluctuations which are highly observable and can be retained when desired.
- Due to its strong ties with the field Control Theory this method is suitable for flow and feedback control [124, 145].

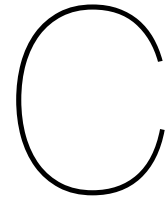
### Weaknesses:

- The method has been mostly applied to linear dynamics such as linearized NS problems. Various extensions are started to appear for nonlinear dynamics [7, 78].
- BPOD is impossible to perform with experimental measurements due the lack of adjoint simulations. Again, variations exist on the BPOD method which makes this possible.

<sup>2</sup>Gramians are are  $n \times n$  symmetric matrices

<sup>3</sup>A non-normal system is the case for many shear-flows





# Appendices - Chapter 4

## C.1. CFD Physics settings

The following StarCCM+ settings are used for the computation of the Vortex Street and the Mixing Layer.

Table C.1: Physics models for both the von Karman Vortex Street and Mixing Layer

Physics models		
<ul style="list-style-type: none"><li>• All y+ Wall Treatment</li><li>• Exact Wall Distance</li><li>• Proximity interpolation</li><li>• Gradients</li></ul>	<ul style="list-style-type: none"><li>• Segregated fluid enthalpy</li><li>• Segregated flow</li><li>• Three dimensional</li><li>• Ideal gas: air</li></ul>	<ul style="list-style-type: none"><li>• Non-Reacting</li><li>• Incompressible</li><li>• Turbulent</li><li>• LES: WALE</li></ul>

## C.2. Pope and CFL criteria

This section provides the mesh, Pope and CFL number figures for the Von Karman Street and Mixing Layer. Represented below are the Pope and CFL number for design point 1 (see Table 4.3).

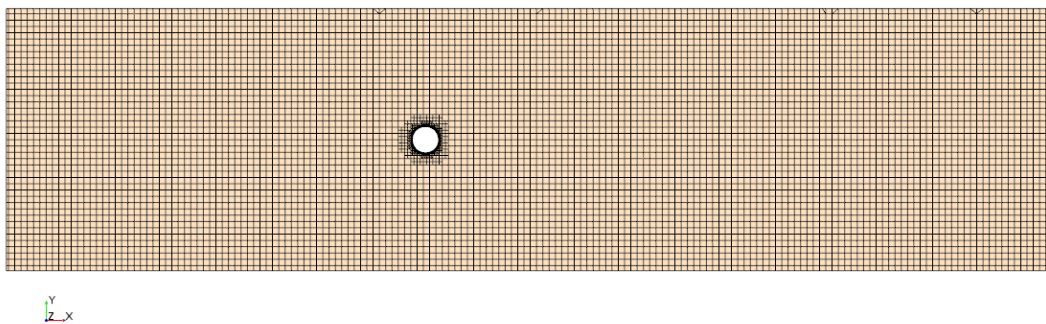


Figure C.1: Cross-section mesh for the Von Karman Vortex Street

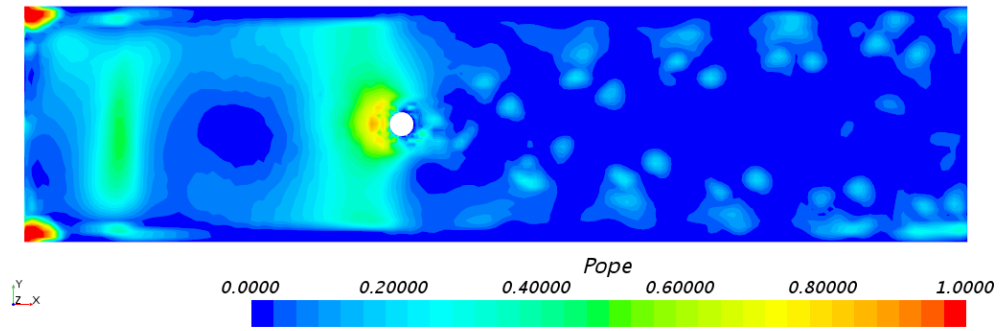


Figure C.2: Pope for the base-case of the V.Karman Street.

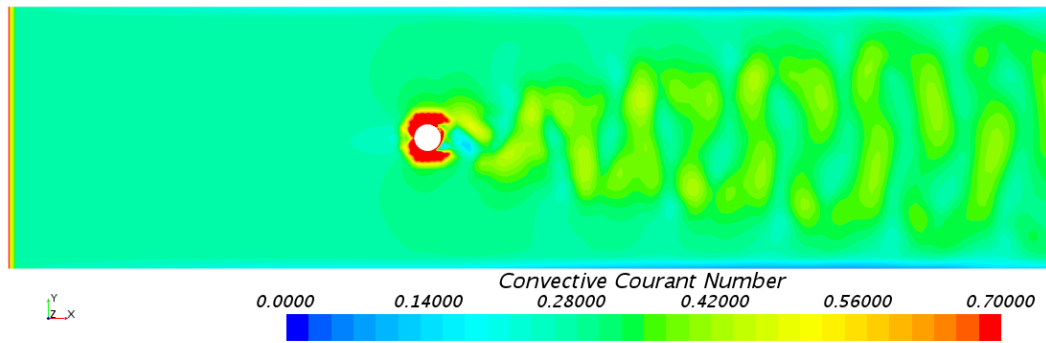


Figure C.3: CFL for the base-case of the V.Karman Street.

Represented below are the Mesh, Pope and CFL number for the mixing layer for design point 4 (see Table 4.6).

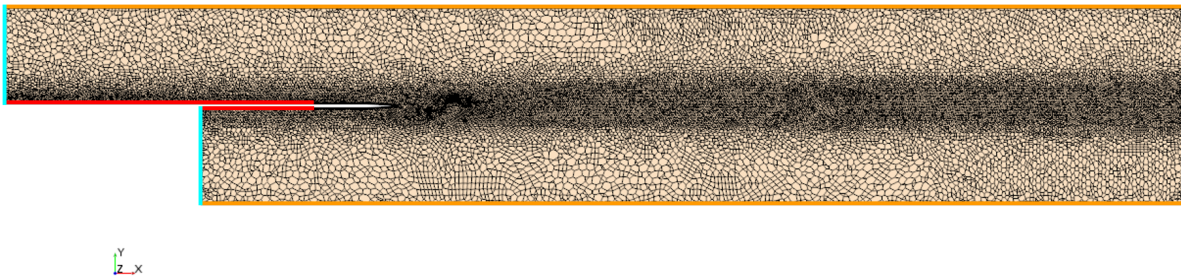


Figure C.4: Cross-section mesh for the Mixing Layer

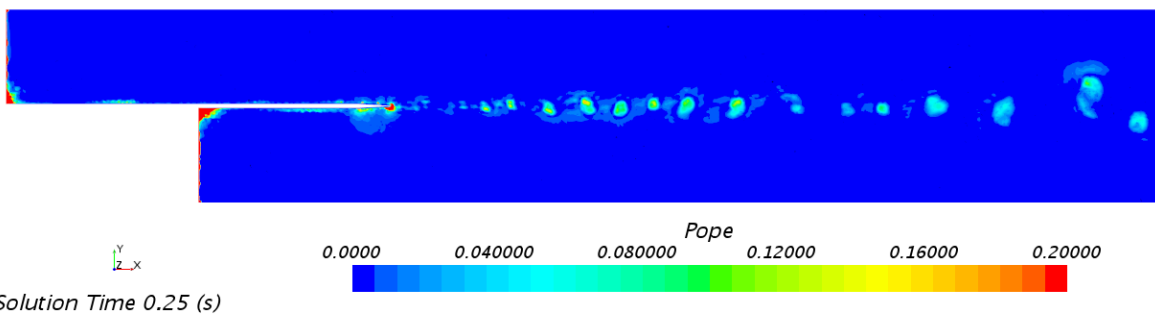


Figure C.5: Pope for the base-case of the Mixing Layer

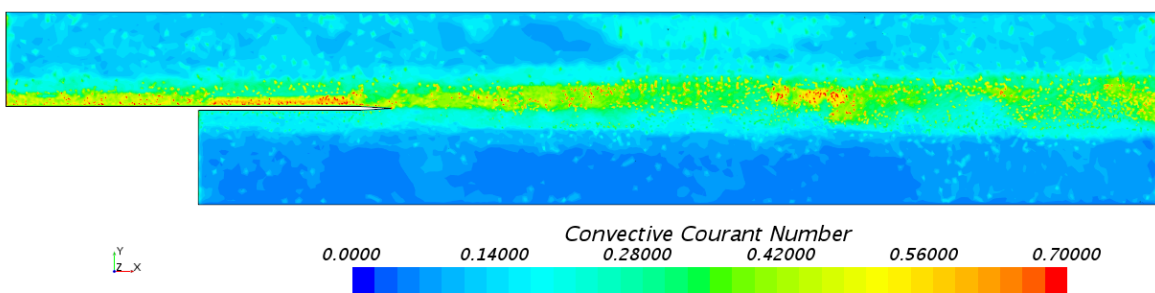


Figure C.6: CFL for the base-case of the Mixing layer

### C.3. Mesh study Mixing Layer

A minor mesh study is performed for the Mixing Layer application case. Three different meshes are compared against experimental data.

- *Coarse mesh*: included a total cell count of 53666.
- *Medium-fine mesh*: included a total cell count of 83634.
- *Fine mesh*: included a total cell count of 122456.

No large differences between the meshes can be seen from Fig.C.7. Therefore, the most coarse mesh is used during this project. These figures are computed in a relative early stage in this research where not a strong emphasis was put on obtaining publication quality graphs. Unfortunately, these figures can not be reworked due to the deletion of the data.

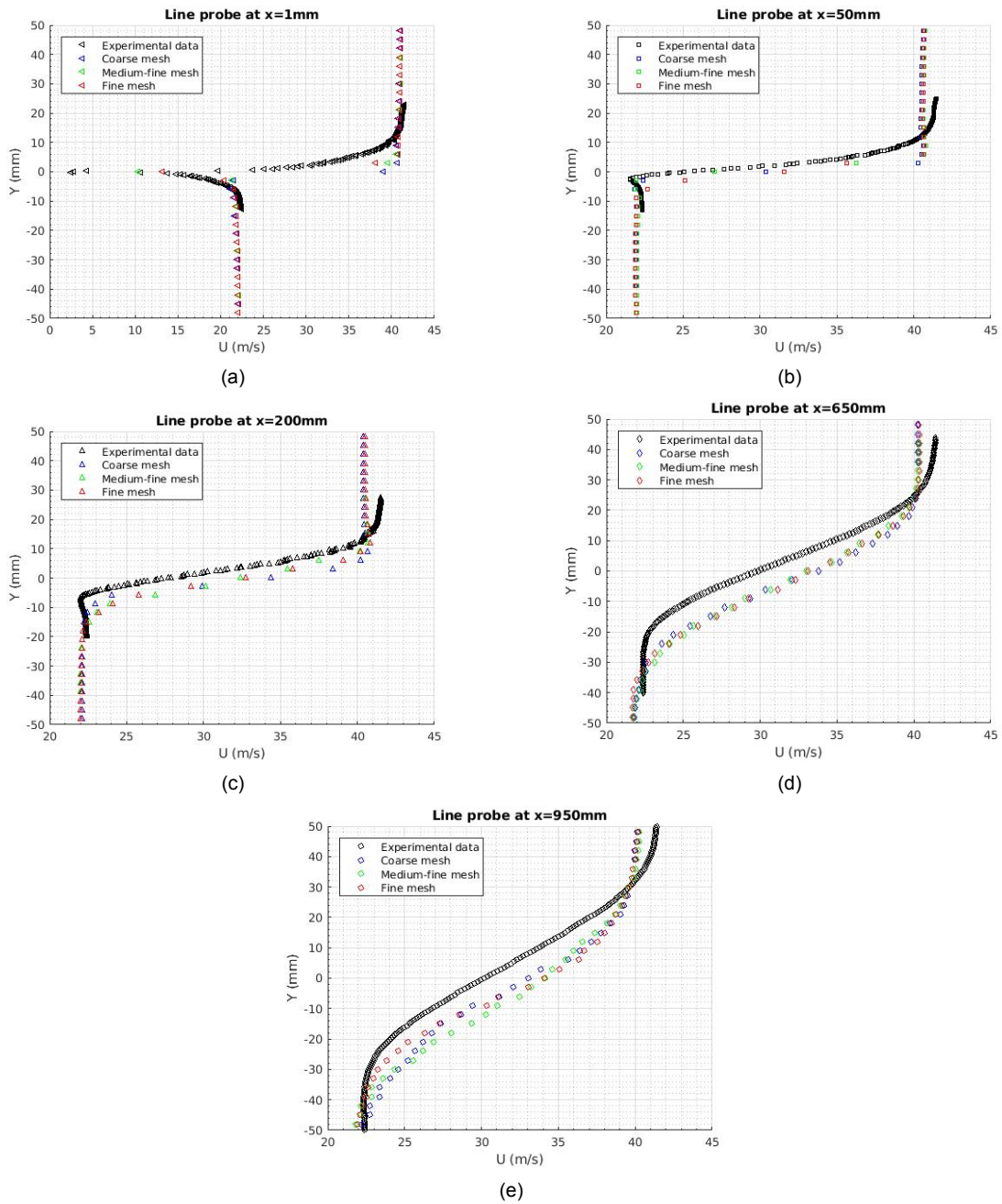
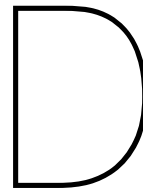


Figure C.7: Velocity profiles for 3 different meshes



# Appendices - Chapter 5

## D.1. Probe points

This section provides 2 additional figures. The first is for the definition of the probe points. The second is for the definition of 3 different domains within the geometry.

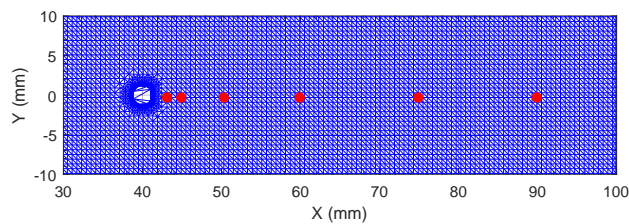


Figure D.1: 5 probe points indicated by the red dots, in the Von Karman domain

Table D.1: Probe points exact position in the Von Karman domain

P.P	X (mm)	Y (mm)
1	47,9	0,3
2	50,3	0,3
3	55,1	0,3
4	70,1	0,3
5	90,5	0,3

The red rectangle highlights the inlet domain. The green rectangle highlights the cylinder domain and the purple rectangle highlights the outlet domain.

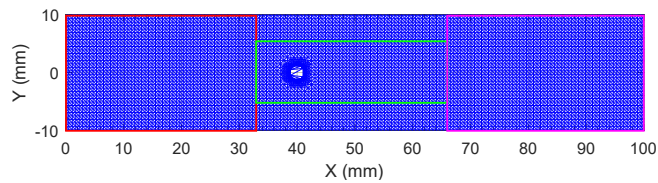


Figure D.2: Different defined domains within the Von Karman geometry

This page provides information on the probe points inside the Mixing Layer domain.

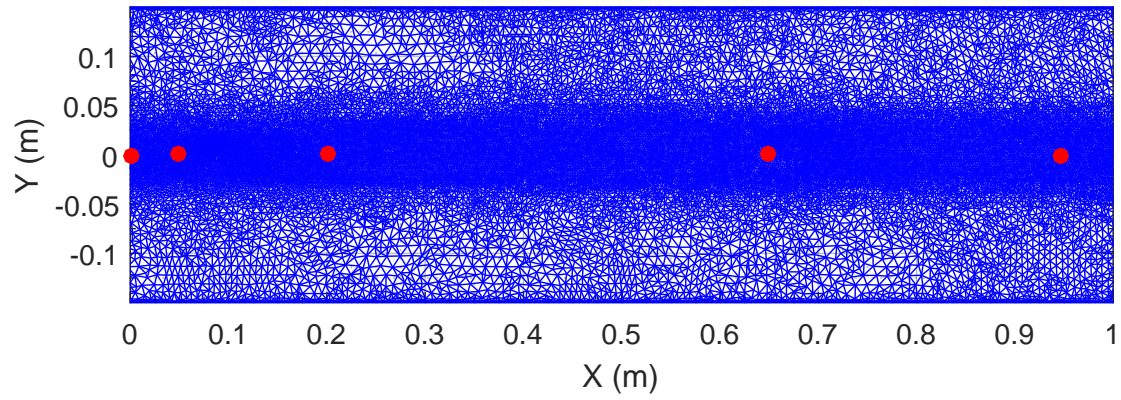


Figure D.3: 5 probe points indicated by the red dots, in the Mixing Layer domain

Table D.2: Probe points exact position in the Mixing Layer domain

P.P	X (mm)	Y (mm)
1	0,61	-0,36
2	49,60	+0,47
3	200,90	+0,29
4	649,30	+0,02
5	946,30	-0,23



## D.2. Error quantification

The following section is written on the topic of error quantification. As stated in Chapter 1.2, the product of exploring a design space is that at every design point its transient flow-field is obtained, be that via high-fidelity CFD simulations or a ML approach. Since the flow-field is the product, it allows the user to perform the first qualitative error quantification. Visual inspection by itself is not enough to judge the overall quality of an ML approach, however it does lend itself to quickly identifying problem areas or approximation failures. To emphasise this point, Fig. D.4 is given which depicts a high-fidelity CFD simulation and an emulation result obtained with a ML approach.

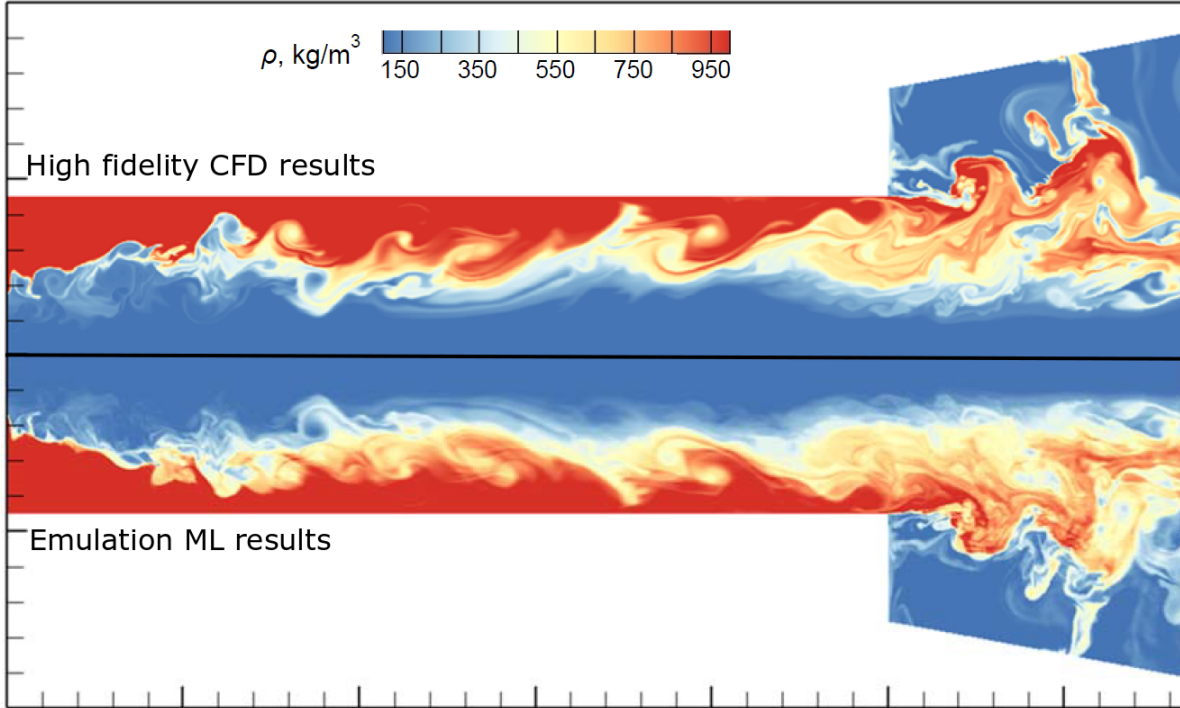


Figure D.4: Comparison between results obtained via high-fidelity CFD and an approximation, also referred to as an emulation, obtained via CFD-ML approach. The flow-field is from left to right.

In Fig.D.4 problem areas can quickly be identified, in which the result obtained from ML fails to approximate the exact CFD simulation. Unsurprisingly, the ML approach struggles in the heavily turbulent areas such as the downstream wake. Visual inspection allows the user to quickly judge the approximation but to supplement visual inspection other methods must be used which quantifies the quality of the approximation. The complexity of the error quantification topic lies not in the lack of methods but in the lack of one method which is all-encompassing and which can describe the quality of a ML approximation in one scalar value. For this reason, throughout this document two different types of error quantification are used to compliment each other. The first method is a Mean-Residual-Error (MRE) which is defined as [88]

$$MRE(t; S) = \frac{\int_S |\vec{c}(\vec{x}, t; S) - \tilde{\vec{c}}(\vec{x}, t; S)| d\vec{x}}{\int_S |\vec{c}(\vec{x}, t; S)| d\vec{x}}, \quad (D.1)$$

where  $S$  denotes the chosen spatial domain and the  $\vec{c}(t; S)$  represents the exact CFD flow-field and  $\tilde{\vec{c}}(t; S)$  the approximation. Equation D.1 is written in its continuous form, its discrete form replaces the integrals with a summation ( $\Sigma$ ) over the spatial coordinates. If a scalar MRE value is desired, the temporal MRE values are averaged over time.

The second method is known as a Root-Mean-Square (RMS) error and is formulated as

$$RMS(\vec{x}; S) = \sqrt{\frac{1}{T} \int_1^T (\vec{c}(\vec{x}, t; S) - \vec{\tilde{c}}(\vec{x}, t; S))^2 dt}. \quad (\text{D.2})$$

The RMS error is relatively similar to a MRE, however the RMS error highlights the areas where the approximation fails and suppresses the areas of decent approximation which is achieved by squaring the deviation. Equation D.1 and D.2 are both expressed in their temporal form, however regularly are these equations computed in the frequency domain. This is to obtain an error quantification at a certain frequency of interest. With the two types of error quantification discussed, lets introduce the overall used guideline on how to asses the quality of a CFD-ML approach. The guideline uses qualitative and quantitative methods to asses the approximation which are described by the following steps.

1. Visual inspection of temporal flow-field. This quickly gives a global understanding of the approximation quality.
2. Fluctuations figures, from the time and frequency domain, provide an additional quantitative method. Local representative information probes are distributed throughout the regions of interests. These allow the user to determine how adequately the CFD-ML approach is able to predict the dynamics (fluctuations) of the exact CFD result.
3. Error quantification compresses all the spatial and temporal information from the deviation between the CFD-ML approximation and the exact CFD result in a scalar value. The used methods for the error quantification are MRE (Eq.D.1) and RMS (Eq.D.2) value.



# Bibliography

- [1] L.I. Abreu, A.V Cavalieri, and W. Wolf. Coherent Hydrodynamic Waves and Trailing-edge Noise. In *23rd AIAA/CEAS Aeroacoustics Conference*. American Institute of Aeronautics and Astronautics, 2017.
- [2] S. Ahmed, H.E. Kadi, and A. Alsharif. Three-dimensional Turbulent Swirling Flow Reconstruction Using Artificial Neural Networks. *Journal of Mechanical Engineering and Automation*, 4(1):1–9, 2014.
- [3] D. Amsallem and C. Farhat. Interpolation Method for Adapting Reduced-Order Models and Application to Aeroelasticity. *AIAA Journal*, 46(7):1803–1813, 2008.
- [4] J. Antony. *Design of Experiments for Engineers and Scientists Second Edition*. Elsevier, 2014.
- [5] S. Arntzen. Data and Scripts from MSc. Thesis: Prediction of Flow-Fields. [https://github.com/TUdelft-SArntzen/MScThesis-Prediction\\_of\\_FlowFields](https://github.com/TUdelft-SArntzen/MScThesis-Prediction_of_FlowFields), 2019.
- [6] N. Aubry, P. Holmes, J.L. Lumley, and E. Stone. The dynamics of coherent structures in the wall region of a turbulent boundary layer. *Journal of Fluid Mechanics*, 192:115–173, 1988.
- [7] S. Bagheri, L. Brandt, C.W. Rowley, and D.S. Henningson. Model Reduction of the Nonlinear Complex Ginzburg – Landau Equation. *Journal of Applied Dynamical Systems*, 9(4):1284–1302, 2010.
- [8] M. Balajewics. *A New Approach to Model Order Reduction of the Navier-Stokes Equations A New Approach to Model Order Reduction of the Navier-Stokes Equations*. PhD thesis, Duke University, 2012.
- [9] T. Bayes and J. Canton. An essay towards solving a problem in the doctrine of chances, 1763.
- [10] A. Behdad. *Enhancement of fuel-air mixing in a lean premixed gas turbine combustion system*. Master thesis, RWTH Aachen University, 2015.
- [11] P. Benner, S. Gugercin, and K. Willcox. A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems. *SIAM Review*, 57(4):483–531, 2015.
- [12] E. Berger, M. Sastuba, D. Vogt, B. Jung, and H. Ben. Estimation of perturbations in robotic behaviour using Dynamic Mode Decomposition. *Advanced Robotics*, 29(5):331–343, 2015.
- [13] G. Berkooz, P. Holmes, and J.L. Lumley. The Proper-Orthogonal Decomposition in the Analysis of Turbulent Flows. *Annual Review of Fluid Mechanics*, 25(1):539–575, 1993.
- [14] S. Bhatnagar, Y. Afshar, S. Pan, K. Duraisamy, and S. Kaushik. Prediction of aerodynamic flow fields using convolutional neural networks. *Computational Mechanics*, 64(2):525–545, 2019.
- [15] R. Bouffanais, M.O. Deville, and E. Leriche. Large-eddy simulation of the flow in a lid-driven cubical cavity. *Physics of Fluids*, 19, 2007.
- [16] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2010.
- [17] C. Braud, D. Heitz, G. Arroyo, L. Perret, and J.P. Bonnet. Low-dimensional analysis , using POD , for two mixing layer – wake interactions. *International Journal of Heat and Fluid Flow*, 25: 351–363, 2004.

- [18] T.A. Brenner, R.L. Fontenot, P.G.A. Cizmas, T.J.O. Brien, and R.W. Breault. A reduced-order model for heat transfer in multiphase flow and practical aspects of the Proper-Orthogonal Decomposition. *Computers and Chemical Engineering*, 43:68–80, 2012.
- [19] B.W. Brunton, L.A. Johnson, J.G. Ojemann, and J.N. Kutz. Extracting spatial–temporal coherent patterns in large-scale neural recordings using Dynamic Mode Decomposition. *Journal of Neuroscience Methods*, 258:1–15, 2016.
- [20] S.L. Brunton and J.N. Kutz. *Data-Driven Science and Engineering*. Cambridge, 2019.
- [21] S.L. Brunton and B.R. Noack. Closed-Loop Turbulence Control: Progress and Challenges. *Applied Mechanics Reviews*, 67(5), 2015.
- [22] S.L. Brunton, J.L. Proctor, and J.N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.
- [23] S.L. Brunton, B.R. Noack, and J.C. Loiseau. From the pod-galerkin method to sparse manifold models. In *Handbook on Model Order Reduction*. De Gruyter GmbH, Germany, 2018.
- [24] S.L. Brunton, B.R. Noack, and P. Koumoutsakos. Machine Learning for Fluid Mechanics. Preprint for Annual Reviews, 2019.
- [25] A.L. Caterini and D.E. Chang. *Deep Neural Networks in a Mathematical Framework*. Springer, 2018.
- [26] M. Cavazutti. *Optimization Methods : From Theory to Design*. Springer, 2013.
- [27] Y.H. Chang. *High-Fidelity emulation of spatio-temporally evolving flow dynamics*. PhD thesis, Georgia Institute of Technology, 2018.
- [28] Y.H. Chang, L. Zhang, X. Wang, and S. Mak. Kernel-Smoothed Proper-Orthogonal Decomposition (KSPOD)-based emulation for prediction of spatiotemporally evolving flow dynamics, 2018.
- [29] S. Chaturantabut and D.C. Sorensen. Nonlinear Model Reduction via Discrete Empirical Interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.
- [30] M. Chui, J. Manyika, M. Miremadi, N. Henke, R. Chung, P. Nel, and S. Malhotra. Notes from the AI frontier. Insights from hundreds of use cases. Technical report, McKinsey&Company, 2018.
- [31] CMI. The seven Millennium Prize Problems. <http://www.claymath.org/millennium-problems/millennium-prize-problems>, 2000.
- [32] A. Colombo and A. Crivellini. Assessment of a sponge layer non-reflecting boundary treatment for high-order CAA/CFD computations. *Computers and Fluids*, 140:1339–1351, 2016.
- [33] B. Colvert, M. Alsalman, and E. Kanso. Classifying vortex wakes using neural networks. *Bioinspiration and Biomimetics*, 13(2), 2018.
- [34] B. de Silva, D.M. Higdon, S.L. Brunton, and J.N. Kutz. Discovery of Physics from Data: Universal Laws and Discrepancy Models, 2019.
- [35] R.J. Dedden. Model Order Reduction using the Discrete Empirical Interpolation Method. Master’s thesis, Technical University of Delft, 2012.
- [36] S. Dederichs, N. Zarzalis, P. Habisreuther, C. Beck, and W. Krebs. Assessment of a Gas Turbine NO<sub>x</sub> Reduction Potential Based on a Spatiotemporal Unmixedness Parameter. *Journal of Engineering for Gas Turbines and Power*, 135:1–8, 2013.
- [37] J. Degroote, J. Vierendeels, and K. Willcox. Interpolation among reduced-order matrices to obtain parameterized models for design, optimization and probabilistic analysis. *International Journal for Numerical Methods in Fluids*, 63:207–230, 2009.

- [38] J. Delville. Description of the data base: Plane turbulent mixing layer from C.E.A.T Poitiers. [http://cfm.mace.manchester.ac.uk/ercoftac/database/cases/case34/Case\\_data/](http://cfm.mace.manchester.ac.uk/ercoftac/database/cases/case34/Case_data/), 1997.
- [39] J. Delville, L. Ukeiley, L. Cordier, J.P. Bonnet, and M. Glauser. Examination of large-scale structures in a turbulent plane mixing layer. Part 1. Proper-Orthogonal Decomposition. *Journal of Fluid Mechanics*, 391:91–122, 1999.
- [40] Z. Doulgerakis, M. Yianneskis, and A. Ducci. On the interaction of trailing and macro-instability vortices in a stirred vessel-enhanced energy levels and improved mixing potential. *Chemical Engineering Research and Design*, 7:412–420, 2009.
- [41] F. Durst. *An introduction to the Theory of Fluid Flows*. Springer, 2008.
- [42] EconSight. Artificial Intelligence. Technical Report January, EconSight, Basel, 2019.
- [43] N.B. Erichson, L. Mathelin, Z. Yao, S.L. Brunton, M.W. Mahoney, and J.N. Kutz. Shallow Learning for Fluid Flow Reconstruction with Limited Sensors and Limited Data, 2019.
- [44] G. Falkovich, G. Boffetta, M. Shats, and A.S. Lanotte. Introduction to Focus Issue : Two- Dimensional Turbulence. *The Physics of Fluids*, 29(110901):29–30, 2017.
- [45] J.H. Ferziger and M. Peric. *Computational methods for fluid dynamics*. Springer, 2002.
- [46] U. Fey, M. König, and H. Eckelmann. A new Strouhal-Reynolds-number relationship for the circular cylinder in the range  $47 < Re < 2 \times 10^5$ . *Physics of Fluids*, 10(7):1547–1549, 1998.
- [47] R.A. Fisher. *The design of experiments*. Oliver & Boyd, Edinburgh, 1935.
- [48] T. Franck, N. and Poinso. Wall-Adapting Local Eddy-Viscosity models for simulations in complex geometries, 1998.
- [49] K. Fukami, K. Fukagata, and K. Taira. Super-resolution reconstruction of turbulent flows with machine learning. *Journal of Fluid Mechanics*, 870:106–120, 2019.
- [50] A.I. Galushkin. *Neural Network Theory*. Springer, 2007.
- [51] M. Gavish and D.L. Donoho. The optimal hard threshold for singular values is  $4/\sqrt{3}$ . *IEEE Transactions on Information Theory*, 60(8):5040–5053, 2014.
- [52] William K. George. Insight into the Dynamics of Coherent Structures from a Proper-Orthogonal Decomposition, 1988.
- [53] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson. Optimal parameter selection for the Alternating Direction Method of Multipliers (ADMM): Quadratic problems. *IEEE Transactions on Automatic Control*, 60(3):644–658, 2015.
- [54] S. Glavaski, J.E. Marsden, and R.M. Murray. Model reduction, centering, and the karhunen-loeve expansion. In *Proceedings of the 37th IEEE Conference on Decision and Control*, pages 2071–2076, 1998.
- [55] J. Grosek and J.N. Kutz. Dynamic Mode Decomposition for Real-Time Background / Foreground Separation in Video, 2014.
- [56] M. Gross, M. Pollefeys, B. Solenthaler, S. Jeong, and L. Ladicky. Data-driven Fluid Simulations using Regression Forests, 2013.
- [57] X. Guo, W. Li, and F. Iorio. Convolutional neural networks for steady flow approximation. In *Proceedings of the 22<sup>nd</sup> ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 481–490. ACM, 2016.
- [58] F.M. Hemez and S. Atamturktur. The dangers of sparse sampling for the quantification of margin and uncertainty. *Reliability Engineering and System Safety*, 2011.

- [59] G.E. Hinton, S. Osindero, and Y.W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.
- [60] P. Holmes, J.L. Lumley, G. Berkooz, and C.W. Rowley. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge, 2012.
- [61] T. Hothorn. CRAN Task View: Machine Learning and Statistical Learning. <https://cran.r-project.org/web/views/MachineLearning.html>, 2019.
- [62] C. Huang, K. Duraisamy, and C. Merkle. Challenges in Reduced Order Modelling of Reacting Flows. *AIAA*, 2018.
- [63] IDC and M. D’Aquila. Worldwide Spending on Cognitive and Artificial Intelligence Systems Forecast to Reach \$77.6 Billion in 2022, According to New IDC Spending Guide. <https://www.idc.com/getdoc.jsp?containerId=prUS44291818>, 2018.
- [64] M. Ilak and C.W. Rowley. Modelling of transitional channel flow using balanced Proper-Orthogonal Decomposition. *Physics of Fluids*, 20(034103):1–17, 2008.
- [65] T. Indlekofer. Evaluation of StarCCM+ to Predict Thermoacoustic Instabilities using Large Eddy Simulations. Master’s thesis, Karlsruhe Institute of Technology, 2017.
- [66] S. Indolia, A.K. Goswami, S.P. Mishra, and P. Asopa. Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach. *Procedia Computer Science*, 132:679–688, 2018.
- [67] M.I. Jordan and T.M. Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349, 2015.
- [68] M.R. Jovanović, P.J. Schmid, and J.W. Nichols. Low-rank and sparse Dynamic Mode decomposition. *Center for Turbulence Research Annual Research Briefs*, pages 139–152, 2012.
- [69] M.R. Jovanović, P.J. Schmid, and J.W. Nichols. Sparsity-promoting Dynamic Mode Decomposition. *Physics of Fluids*, 26(2):1–22, 2014.
- [70] E. Kaiser, B.R. Noack, L. Cordier, A. Spohn, M. Segond, M. Abel, G. Daviller, J. Östh, S. Krajinović, and R.K. Niven. Cluster-based reduced-order modelling of a mixing layer. *Journal of Fluid Mechanics*, 754:365–414, 2014.
- [71] R.H. Kaser. *Fractal grids as static mixers for modern lean premixed combustion systems*. Master thesis, University of Twente, 2016.
- [72] J.P. Keating, J.E. Michalek, and J.T. Riley. A note on the optimality of the Karhunen-Loeve expansion. *Pattern Recognition Letters*, 1(4):203–204, 1983.
- [73] R.H. Kraichnan. Inertial Ranges in Two-Dimensional Turbulence. *The Physics of Fluids*, 10(7), 1967.
- [74] R.H. Kraichnan and D. Montgomery. Two-dimensional turbulence. *Reports on Progress in Physics*, 43(5):547–619, 1980.
- [75] P.K. Kundu, I.M. Cohen, and D.R. Dowling. *Fluid Mechanics*. Elsevier, 2015.
- [76] J. Nathan Kutz, Steven L. Brunton, Dirk M. Luchtenburg, Clarence W. Rowley, and Jonathan H. Tu. On Dynamic Mode Decomposition: Theory and applications. *Journal of Computational Dynamics*, 1(2):391–421, 2014.
- [77] J.N. Kutz. Deep learning in fluid dynamics. *Focus on Fluids*, 814:1–4, 2017.
- [78] S. Lall, J.E. Marsden, and S. Glavaski. A subspace approach to balanced truncation for model reduction of nonlinear control systems. *International Journal on Robust and Nonlinear Control*, pages 1–21, 2000.
- [79] T. Lassila, A. Manzoni, A. Quarteroni, and G. Rozza. Model order reduction in fluid dynamics: Challenges and perspectives, 2014.

- [80] C. Lee, J. Kim, D. Babcock, and R. Goodman. Application of neural networks to turbulence control for drag reduction. *Physics of Fluids*, 9(1740), 1997.
- [81] A.M. Legendre. Nouvelles méthodes pour la détermination des orbites des comètes, 1805.
- [82] H. Lewy, K. Friedrichs, and R. Courant. Über die partiellen differenzgleichungen der mathematischen physik. *Mathematische Annalen*, 100:32–74, 1928.
- [83] John H. Lienhard. *Synopsis of Lift, Drag and Vortex Frequency Data for Rigid Circular Cylinders*. Technical Extension Service, Washington, 1966.
- [84] A. Liné. Eigenvalue spectrum versus energy density. *Chemical Engineering Research and Design*, 108:13–22, 2015.
- [85] H.F.S. Lui and W.R. Wolf. Construction of reduced-order models for fluid flows using deep feed-forward neural networks. *Journal of Fluid Mechanics*, 872:963–994, 2019.
- [86] J.L. Lumley. The Structure of Inhomogeneous Turbulence. In *Atmospheric Turbulence and Wave Propagation*, pages 166–178, Moscow, 1967.
- [87] S. Mak, C.L. Sung, X. Wang, S.T. Yeh, V.R. Joseph, V. Yang, and C.F.J. Wu. An efficient surrogate model for emulation and physics extraction of large eddy simulations, 2017.
- [88] S. Mak, C.L. Sung, X. Wang, S.T. Yeh, Y.H. Chang, R. Joseph, V. Yang, and C.F.J. Wu. An Efficient Surrogate Model for Emulation and Physics Extraction of Large Eddy Simulations. *Journal of the American Statistical Association*, 0(0):1–14, 2018.
- [89] A. Mani. On the reflectivity of sponge zones in compressible flow simulations. *Center for Turbulence Research Annual Research Briefs*, pages 117–133, 2010.
- [90] Jordan Mann and J. Nathan Kutz. Dynamic mode decomposition for financial trading strategies. *Quantitative Finance*, 16(11):1643–1655, 2016.
- [91] A.A. Markov. Extensions of the limit theorems of the calculus of probability onto a sum of magnitudes connected into a chain. St. Petersburg, 1908.
- [92] B. Marr. 27 Incredible Examples of AI and Machine Learning in Practice. <https://www.forbes.com/sites/bernardmarr/2018/04/30/27-incredible-examples-of-ai-and-machine-learning-in-practice/>, 2018.
- [93] G. Matheron. Principles of Geostatistics. *Economic Geology*, 58:1246 – 1266, 1963.
- [94] P. McCarthy. Processing Power Compared. <https://pages.experts-exchange.com/processing-power-compared>, 2015.
- [95] W. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5, 1943.
- [96] M. Minsky and S. Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, 1969.
- [97] P. Moin. High-Fidelity Numerical Simulations of Multiphysics Turbulent Flows in Complex Geometries, 2012.
- [98] B.C. Moore. Principal Component Analysis in Linear Systems : Controllability , Observability , and Model Reduction. *IEEE Transactions on Automatic Control*, 26(1), 1981.
- [99] A. Narasingam and J.S.I. Kwon. Data-driven identification of interpretable reduced-order models using sparse regression. *Computers and Chemical Engineering*, 119:101–111, 2018.
- [100] NASA. von Karman Vortices. <https://earthobservatory.nasa.gov/images/2613/von-karman-vortices>, 1999.



- [101] R.M. Needham, H.C. Longuet-Higgins, and D. Michie. Lighthill report: artificial intelligence. [http://www.chilton-computing.org.uk/inf/literature/reports/lighthill\\_report/contents.htm](http://www.chilton-computing.org.uk/inf/literature/reports/lighthill_report/contents.htm), 1973.
- [102] V.B. Nguyen, M. Buffoni, K. Willcox, and B.C. Khoo. Model reduction for reacting flow applications. *International Journal of Computational Fluid Dynamics*, 28(3-4):91–105, 2014.
- [103] F.T.M. Nieuwstadt, B.J. Boersma, and J. Westerweel. *Turbulence: Introduction to theory and applications of turbulent flows*, 2016.
- [104] B.R. Noack. From snapshots to modal expansions – bridging low residuals and pure frequencies. *Journal of Fluid Mechanics*, 802:1–4, 2016.
- [105] B.R. Noack, M. Morzynski, and G. Tadmor. *Reduced-Order Modelling for Flow Control*. Springer, 2011.
- [106] N.T. Ouellette. Turbulence in two dimensions. *Physics Today*, 65(5), 2012.
- [107] V. Parezanović, J.C. Larentie, C. Fourment, J. Delville, J.P. Bonnet, A. Spohn, T. Duriez, L. Cordier, B.R. Noack, M. Abel, M. Segond, T. Shaqarin, and S.L. Brunton. Mixing Layer Manipulation Experiment. *Flow, Turbulence and Combustion*, 94(1):155–173, 2015.
- [108] T.J. Poinsot and S.K. Lele. Boundary Conditions for Direct Simulations Compressible Viscous Flows. *Journal of Computational Physics*, 101:104–129, 1992.
- [109] A. Pollard, L. Castillo, L. Danaila, and M. Glauser. *Whither Turbulence and Big Data in the 21st Century ?* Springer, 2017.
- [110] S.B. Pope. *Turbulent Flows*. Cambridge University Press, 2000.
- [111] S.B. Pope. Ten questions concerning the large-eddy simulation. *New Journal of Physics*, 6(35), 2004.
- [112] J.L. Proctor and P.A. Eckhoff. Discovering dynamic patterns from infectious disease data using Dynamic Mode Decomposition. *International Health*, 7:139–145, 2015.
- [113] J. Pruvost, J. Legrand, and P. Legentilhomme. Three-Dimensional Swirl Flow Velocity-Field Reconstruction Using a Neural Network With Radial Basis Functions. *Journal of Fluids Engineering*, 123(4):920, 2002.
- [114] M. Raissi. Deep Hidden physics models: Deep learning of nonlinear partial differential equations, 2018.
- [115] M. Raissi and G.E. Karniadakis. Hidden physics models: Machine learning of nonlinear partial differential equations, 2017.
- [116] M. Raissi, Z. Wang, M.S. Triantafyllou, and G.E. Karniadakis. Deep learning of vortex-induced vibrations. *Journal of Fluid Mechanics*, 861:119–137, 2019.
- [117] Daniel Reinhardt. Optimization of Combustor Air Inflow for Low NOx Emissions. Master's thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2017.
- [118] D. Rempfer. On low-dimensional Galerkin models for fluid flow. *Theoretical and Computational Fluid Dynamics*, 14(2):75–88, 2000.
- [119] J. Richards and X. Jia. *Remote Sensing Digital Image Analysis: an Introduction*. Springer, 2006.
- [120] N. Riley and P. Drazin. *The Navier-Stokes Equations: A Classification of Flows and Exact Solutions*. Cambridge, 2007.
- [121] M.K. Rivera. *The Inverse Energy Cascade of Two-Dimensional Turbulence*. PhD thesis, University of Pittsburgh, 2001.
- [122] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain., 1958.

- [123] C.W. Rowley. Model Reduction for Fluids, using Balanced Proper-Orthogonal Decomposition. *International Journal on Bifurcation and chaos*, pages 1–24, 2005.
- [124] C.W. Rowley and S.T.M. Dawson. Model Reduction for Flow Analysis and Control. *Annual Reviews of Fluid Mechanics*, 2017.
- [125] P. Sagaut, S. Deck, and M. Terracol. *Multiscale and Multiresolution Approaches in Turbulence*. Imperial College Press, 2006.
- [126] W.H.A Schilders, H.A. van der Vorst, and J. Rommes. *Model Order Reduction*. Springer, 2008.
- [127] P. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010.
- [128] P.J. Schmid, L. Li, M.P. Juniper, and O. Pust. Applications of the Dynamic Mode Decomposition. *Theoretical and Computational Fluid Dynamics*, 25(1-4):249–259, 2011.
- [129] R. Schwarze. *CFD Modellierung: Grundlagen und Anwendungen bei Strömungsprozessen*. Springer, 2012.
- [130] O. Semeraro, S. Bagheri, L. Brandt, and D.S. Henningson. Feedback control of three-dimensional optimal disturbances using reduced-order models. *Journal of Fluid Mechanics*, pages 1 – 40, 2011.
- [131] A.A. Shah, W.W. Xing, and V. Triantafyllidis. Reduced-order modelling of parameter-dependent, linear and nonlinear dynamic partial differential equation models. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2200), 2017.
- [132] M. Sieber, C.O. Paschereit, and K. Oberleithner. Spectral Proper-Orthogonal Decomposition. *Journal of Fluid Mechanics*, 792:798–828, 2016.
- [133] F.R.S.T. Simpson. *Trigonometry, Plane and Spherical: with the construction and applications of logarithms*. Gale ECCO, 1714.
- [134] L. Sirovich. Turbulence and the dynamics of coherent structures . I - Coherent structures . II - Symmetries and transformations . III - Dynamics and scaling. *Quarterly of Applied Mathematics*, 45(3):561 – 571, 1987.
- [135] Lawrence Sirovich and B. Knight. Kolmogorov Inertial Range for Inhomogeneous Turbulent Flows, 1990.
- [136] J. Smagorinsky. General circulation experiments with the primitive equations, 1. the basic experiment. *Monthly Weather Review*, 91:99–164, 1963.
- [137] T.R. Smith, J. Moehlis, and P. Holmes. Low-dimensional modelling of turbulence using the proper-orthogonal decomposition: A tutorial. In *Nonlinear Dynamics*, pages 275–307. Springer, 2005.
- [138] Siemens PLM Software. HEEDS MDO Manual, 2018.
- [139] Siemens PLM Software. STAR-CCM+ Manual, 2018.
- [140] Department of Computer Science Stanford University. Neural Networks History: the 1940's to the 1970's. <https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/index.html>, 2000.
- [141] B. Steinfurth and F. Haucke. Coherent Structures in the Actively Controlled Wake of a High-Lift Configuration. *AIAA Journal*, 56(10):3848–3856, 2018.
- [142] V Strouhal. Über eine besondere Art der Tonerregung. *Annalen der Physik*, 241(10):216–251, 1878.
- [143] S.P. Symon. *Reconstruction and Estimation of Flows Using Resolvent Analysis and Data-Assimilation*. PhD thesis, California Institute of Technology, 2018.

- [144] K. Taira. Revealing essential dynamics from high-dimensional fluid flow data and operators, 2018.
- [145] K. Taira, S.L. Brunton, S.T.M. Dawson, C.W. Rowley, T. Colonius, B.J. McKeon, O.T. Schmidt, S. Gordeyev, V. Theofilis, and L.S. Ukeiley. Modal Analysis of Fluid Flows: An Overview. *AIAA Journal*, 55(12), 2018.
- [146] K. Taira, M.S. Hemati, S.L. Brunton, Y. Sun, K. Duraisamy, S. Bagheri, S.T.M. Dawson, and C.A. Yeh. Modal Analysis of Fluid Flows: Applications and Outlook, 2019.
- [147] J. Tompson, K. Schlachter, P. Sprechmann, and K. Perlin. Accelerating eulerian fluid simulation with convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning*, ICML'17, pages 3424–3433. JMLR, 2017.
- [148] A. Towne, O.T. Schmidt, and T. Colonius. Spectral Proper-Orthogonal Decomposition and its relationship to Dynamic-Mode Decomposition and Resolvent Analysis. *Journal of Fluid Mechanics*, 847:821–867, 2018.
- [149] M. Tutkun and W.K. George. Lumley decomposition of turbulent boundary layer at high Reynolds numbers. *Physics of Fluids*, 29(020707), 2017.
- [150] R.J. Vacaro. *SVD and signal processing, II: algorithms, analysis and applications*. Elsevier, 1991.
- [151] Milton van Dyke. *An Album of Fluid Motion*, 1982.
- [152] M.E. Wall, A. Rechtsteiner, and L.M. Rocha. Singular value decomposition and principal component analysis. In *A practical approach to microarray data analysis*, pages 91–110. Kluwer Academic Publishers, 2002.
- [153] J.X. Wang and H. Xiao. Data-Driven CFD Modeling of Turbulent Flows Through Complex Structures. Pre-print for Elsevier, 2016.
- [154] X. Wang, S.T. Yeh, Y.H. Chang, and V. Yang. A high-fidelity design methodology using LES-based simulation and POD-based emulation: A case study of swirl injectors. *Chinese Journal of Aeronautics*, 31(9):1855–1869, 2018.
- [155] C. White, D. Ushizima, and C. Farhat. Neural Networks Predict Fluid Dynamics Solutions from Tiny Datasets. Pre-print, 2019.
- [156] B. Widrow. An adaptive "ADALINE" Neuron using Chemical "Memistors". Technical report, Stanford University, 1960.
- [157] Wikipedia. List of datasets for machine-learning research. [https://en.wikipedia.org/w/index.php?title=List\\_of\\_datasets\\_for\\_machine-learning\\_research&oldid=921041165](https://en.wikipedia.org/w/index.php?title=List_of_datasets_for_machine-learning_research&oldid=921041165), 2019.
- [158] D.C. Wilcox. *Turbulence Modeling for CFD*. DCW Industries, Inc., 2006.
- [159] M.O. Williams, I.G. Kevrekidis, and C.W. Rowley. A Data-Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.
- [160] B. Wohlberg. ADMM Penalty Parameter Selection by Residual Balancing, 2017.
- [161] D. Xiao, C.E. Heaney, L. Mottet, F. Fang, W. Lin, I.M. Navon, Y. Guo, O.K. Matar, A. G. Robins, and C.C. Pain. A reduced order model for turbulent flows in the urban environment using machine learning. *Building and Environment*, 148(July 2018):323–337, 2019.
- [162] F. Xiao, W. Deng, L. Peng, C. Cao, K. Hu, and X. Gao. MSDNN: Multi-Scale Deep Neural Network for Salient Object Detection, 2018.
- [163] Y. Xie, E. Franz, M. Chu, and N. Thuerey. tempoGAN: a Temporally Coherent, Volumetric GAN for Super-Resolution Fluid Flow. *ACM Transactions on Graphics*, 37(4), 2018.



- [164] J. Xu, C. Huang, and K. Duraisamy. Reduced-order modeling framework for combustor instabilities using truncated domain training, 2019.
- [165] S.T. Yeh, X. Wang, C.L. Sung, S. Mak, Y.H. Chang, and L. Zhang. Common Proper-Orthogonal Decomposition-Based Spatio-temporal Emulator for Design Exploration. *AIAA*, 56(6), 2018.
- [166] J. Yu and J.S. Hesthaven. Flowfield Reconstruction Method Using Artificial Neural Network. *AIAA Journal*, 57(2):482–498, 2019.
- [167] M. Yuan and C.H. Zhang. On tensor completion via nuclear norm minimization. *Foundations of Computational Mathematics*, 16(4):1031–1068, 2016.
- [168] Z.J Zhang and K. Duraisamy. Machine Learning Methods for Data-Driven Turbulence Modeling. In *22nd AIAA Computational Fluid Dynamics Conference*, pages 1–18, 2015.
- [169] D. Zhao, D. Liu, and M. Zhu. A surrogate model for thermal characteristics of stratospheric airship. *Advances in Space Research*, 61:2989–3001, 2018.
- [170] R. Zimmermann. A Locally Parametrized Reduced-Order Model for the Linear Frequency Domain Approach to Time-Accurate Computational Fluid Dynamics. *SIAM Journal on Scientific Computing*, 36(3):508–537, 2014.
- [171] R. Zimmermann. Manifold interpolation and model reduction. In *Handbook on Model Order Reduction*. De Gruyter GmbH, Germany, 2019.
- [172] K. Zuse. *The Computer—My Life*. Springer-Verlag, Berlin, Heidelberg, 1993.