

TECHNICAL UNIVERSITY OF BERLIN

Institute of Fluid Dynamics and Technical Acoustics
Laboratory for Flow Instabilities and Dynamics
Prof. Dr.-Ing. K. Oberleithner

Master Thesis

IDENTIFICATION OF COHERENT STRUCTURES IN UNSTEADY FLOWS OF JET ENGINE COMPONENTS THROUGH DYNAMIC MODE DECOMPOSITION

Maximilian Kern
Berlin, den 14.12.2020

Matriculation number: 372729
Study program: Mechanical engineering M.Sc.
Supervisors: Prof. Dr.-Ing. Kilian Oberleithner (Technical University of Berlin)
Dr.-Ing. Gero Schieffer (Rolls-Royce Deutschland)

ERKLÄRUNG DER SELBSTSTÄNDIGKEIT
DECLARATION OF AUTHORSHIP

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

This thesis is composed of my original work, and contains no material previously published or written by another person except where due reference has been made in the text. I have clearly stated the contribution by others to jointly-authored works that I have included in my thesis.

Berlin, den 14.12.2020



Maximilian Kern

ZUSAMMENFASSUNG

Strömungen in komplexen Geometrien von Triebwerkskomponenten sind aufgrund turbulenter Phänomene hoch dynamisch. Die dynamic mode decomposition wird eingesetzt, um kohärente Strukturen zu detektieren und die Dynamik komplexer Strömungsfelder aufzulösen. Die vier Varianten, standard-, exact-, compressed- und compressed sensing dynamic mode decomposition werden implementiert um ihre Leistung in der Anwendung hinsichtlich Zeitbedarf, Speicherverbrauch und Genauigkeit zu vergleichen. Insgesamt werden drei verschiedene Fälle untersucht, darunter ein zweidimensionales Feld aus künstlichen Wellen, ein generischer Fall eines dreidimensionalen Zylinders in einer Querströmung sowie ein Stator einer zweistufigen Hochdruckturbinen. Das künstliche Feld zeigt die Eignung der Verfahren, indem Ergebnisse der Moden Detektion mit den vorbestimmten Parametern verglichen werden. Exact- und compressed dynamic mode decomposition werden ferner auf einen Zylinder im Querstrom angewandt, was eine gute Übereinstimmung mit dem ursprünglichen Experiment zeigt, indem dominanten Moden im Zylinder nachlauf erfasst werden. Schließlich zeigt die Untersuchung des Stators der Turbinen die Fähigkeit der Modalzerlegung, die dominante Struktur der Stator-Rotor-Wechselwirkung im hochdynamischen Strömungsfeld aufzudecken.

ABSTRACT

Flow through complex geometries of jet engine components exhibit flow dynamics due to turbulent phenomena. Dynamic mode decomposition is deployed to detect coherent structures and resolve the dynamics within complex flow fields. Four variants, namely standard-, exact-, compressed- and compressed sensing dynamic mode decomposition are implemented to compare their performance in the application regarding timing, memory consumption and accuracy. A total of three different cases are particularly investigated including an non-flow related two dimensional field of artificial waves, a generic case of a three dimensional cylinder in a cross flow and a stator section from a two stage high pressure turbine. The artificial field evidences the capabilities of the methods by comparing results of mode detection to predetermined parameters. Exact- and compressed dynamic mode decomposition are further applied onto the cylinder in a cross flow revealing good agreement with the original experiment the data is based on by detecting the dominant modes in the cylinder wake. Finally the investigation of the stator section of the turbine case proves the ability of dynamic mode decomposition to reveal dominant structures of the stator-rotor interaction in the highly dynamical flow field.

CONTENTS

List of Figures	v
List of Tables	vii
Nomenclature	viii
Glossary	xii
1 Introduction	1
2 Theory	4
2.1 Important concepts	4
2.1.1 Coherent structures	4
2.1.2 Modes	6
2.2 Established methods for the identification of coherent structures	7
2.2.1 Proper orthogonal decomposition	7
2.2.2 Time discrete Fourier transformation	10
2.3 DMD Theory	12
2.3.1 General idea	12
2.3.2 Solution of linear ODE's of first order	12
2.4 DMD algorithm	18
2.4.1 standard DMD	19
2.4.2 Mode order	23
2.5 Alternatives of standard algorithm	24
2.5.1 exact DMD	24
2.5.2 compressed DMD and compressed sensing DMD	24
3 Data, Method, Measurement	29
3.1 Artificial flow field	29
3.2 3D Cylinder in a cross flow	32
3.3 Turbine case	33
4 Results	35

CONTENTS

4.1	Mode detection in the artificial flow field	35
4.2	Analysis of 3D cylinder in cross flow	42
4.3	Analysis of turbine rotor	52
5	Conclusion	56
	Bibliography	58
A	Appendix	60
A.1	Notes on the Pseudo-Inverse	60
A.2	The Arnoldi algorithm as basic idea to approximate eigenvalues	61
A.3	Artificial flow field	64
A.4	3D cylinder in cross flow with less snapshots	67

LIST OF FIGURES

2.1	Turbulent spectrum of model flow with sufficient high Reynolds number. . .	5
2.2	Mixing layer of two flows with equal density. Velocity ratio = 0.38, Reynolds number = 10^6 . From Bernal et al. [1979].	6
2.3	Sketch of flow field around a 2D Cylinder in a cross flow. Left to right, top to bottom: Flow field snapshot at $t = t_1$, collection of flow field snapshots, discretized flow field at $t = t_1$ with flow variable a , rearranged flow field in snapshot vector and rearranged snapshot collection in snapshot matrix.	9
2.4	Dimensions of singular value decomposition for $M > N - 1$. Top: Regular algorithm where grey part of U is not having any effect since it is multiplied by the zero entries of Σ . Bottom: economy SVD with reduced dimensions of U and Σ	21
3.1	Overview of data set #1. Left to right, top to bottom: Wave $w_{0,1}$, wave $w_{0,2}$, wave $w_{0,3}$, wave $w_{0,4}$, wave $w_{0,5}$ and the superposition of all including noise at $t = 0s$	31
3.2	Artificial flow-field from data-set # 3. Left: Flow field at $t_0 = 0s$, right: Flow field at $t_{101} = 2.5s$	32
3.3	Sketch of the domain of the 3D cylinder in a cross flow. Grey shaded planes are inlet on the left and outlet on the right. All other planes are solid.	33
4.1	Real part of modes of data set # 1 detected with standard DMD algorithm. From left to right, top to bottom, mode with frequencies of: 0.5Hz, 2Hz, 5Hz, 16Hz, 18Hz and the average mode at 0Hz	36
4.2	Physical unimportant mode of artificial flow field, detected by standard DMD. Left: real part, right: imaginary part. Frequency: 6.717Hz	36
4.3	Ritzvalues acquired by exact DMD on data set #3. Black dots indicate detected modes' Ritzvalues.	38
4.4	Deviation of detected 5Hz mode frequency with growing number of used snapshots with exact DMD.	39
4.5	Deviation of detected mode frequencies with growing time step.	40

LIST OF FIGURES

4.6	Detected modes of compressed DMD with 1% single pixel compression. Clockwise, real part of 0.5Hz, 5Hz, 18Hz and 16Hz mode.	41
4.7	Real part of modes of data set # 1 detected with compressed sensing DMD algorithm. From left to right, top to bottom, mode with frequencies of: 0.5Hz, 2Hz, 5Hz, 16Hz, 18Hz and the average mode at 0Hz	43
4.8	Slice of real part of dominant mode of SAS data with 200 snapshots. Analysed quantity: pressure. Frequency: 1270Hz.	44
4.9	Slice of real part of dominant modes of SAS data with 200 snapshots. Analysed quantity: pressure. Frequency: 2530Hz.	45
4.10	Ritzvalues of exact DMD from 200 snapshots of SAS data set. Analysed quantity: pressure. Black triangle indicates average mode.	46
4.11	Slice of real part of dominant mode at 1340Hz of URANS data with 200 snapshots. Analysed quantity: pressure.	48
4.12	Slice of real part of dominant mode at 2640Hz of URANS data with 200 snapshots. Analysed quantity: pressure.	49
4.13	Ritzvalues of exact DMD from 200 snapshots of URANS data set. Analysed quantity: x-velocity. Black triangle indicates average mode, red triangles indicate non average zero frequency modes.	50
4.14	Second dominant modes at 2530Hz detected by compressed DMD. 2D cut of 1% compression (green) overlain with 3% compression (grey) at same data threshold.	51
4.15	Ritzvalues of exact DMD from 200 snapshots of turbine rotor data set. Analysed quantity: pressure. Black triangle indicates average mode, red triangle indicates non-average zero frequency mode.	53
4.16	Scheme of azimuthal section through mode with frequency closest to F_{SRI} . Colours indicate mode magnitude.	54
A.1	Overview of data set #2. Top to bottom, left to right: Wave $w_{0,1}$, wave $w_{0,2}$, wave $w_{0,3}$, wave $w_{0,4}$, wave $w_{0,5}$ and the superposition of all including noise at $t = 0s$	66
A.2	Ritzvalues of exact DMD from 50 snapshots of SAS data set. Analysed quantity: pressure. Black triangle indicates average mode, red triangle indicates non average, zero frequency mode.	67
A.3	Ritzvalues of exact DMD from 100 snapshots of SAS data set. Analysed quantity: pressure. Black triangle indicates average mode.	67

LIST OF TABLES

3.1	Wave parameters of first data set of the artificial flow field. Angular frequency and velocity are derived from wave-number and frequency.	30
3.2	Wave parameters of third data set of the artificial flow field. Frequency and velocity are derived from wave-number and angular frequency.	32
4.1	Frequencies and corresponding deviation of detected frequencies of standard DMD, exact DMD and compressed DMD. Compressed DMD compression manner in brackets, percentage is use of actual points in space from entire data.	42
4.2	Energy criteria, frequency and absolute Ritzvalue of first seven modes detected by exact DMD. Modes are ordered according to highest energy criteria value.	47
4.3	Percentaged deviation of smallest three non-zero mode frequency of stable and unstable branch from corresponding F_{SRI} harmonic frequency. Abstract of the analyse of pressure, density and x-velocity.	54
A.1	Overview of the three given examples of creating a reduced order matrix that approximates leading eigenvalues. Method 1: Based on Krylov subspace with linear coefficients. Method 2: The Arnoldi algorithm. Method 3: Used by DMD.	64
A.2	Wave parameters of second data set of the artificial flow field. Frequency and velocity are derived from wave-number and angular frequency.	65

NOMENCLATURE

Linear algebra notation

x	Scalar value x
\underline{x}	Vector x
$\underline{\underline{X}}$	Matrix x
$\underline{\underline{X}}^{-1}$	Inverse of matrix x
$\underline{\underline{X}}^T$	Transpose of matrix x
$\underline{\underline{X}}^H$	Hermetian transpose of matrix x
$\underline{\underline{X}}^\dagger$	Pseudo inverse of matrix x
$\ x\ $	Norm of matrix x
$x^{[II]}$	Lagrange notation of second time derivative of x
$x^{[N]}$	Lagrange notation of N th time derivative of x
\langle , \rangle	Dot product

Number Sets

\mathbb{R}	Real Numbers
\mathbb{C}	Complex Numbers
\mathbb{N}	Natural Numbers

Physical and mathematical assigned characters

E_{TKE}	Turbulence kinetic energy
κ	Wavelength
\underline{x}	Vector of state (data vector, snapshot)
\underline{x}'	Fluctuating part of vector of state

\bar{x}	Mean time average of system snapshots
a	POD time coefficient
$\underline{\underline{X}}$	Snapshot matrix
M	Number of points in space
N	Number of points in time
$\underline{\underline{G}}$	Covariance matrix
φ	Eigenvector
λ	Eigenvalue
σ	Singular value
\hat{x}	Vector of state in spectral domain
ω	Angular frequency
ω_0	Fundamental frequency
T	Measurement period
Δt	Time delta in between snapshots
$\underline{\underline{\Gamma}}$	Fourier matrix
F_s	Sampling frequency
F_{max}	Maximum sampling frequency
$\underline{\underline{B}}$	System matrix of continuous ODE system
b	One dimensional discrete system scalar
c	Constant value
\underline{c}, \tilde{c}	Vector with constant values
$\underline{\xi}$	Continuous system eigenvector
μ	Continuous system eigenvalue
$\underline{\underline{I}}$	Identity matrix
$\underline{\underline{D}}$	Diagonalized system matrix
$\underline{\underline{\Xi}}$	Continuous system: matrix with all eigenvectors in columns
$\underline{\underline{A}}$	Discrete system matrix
f	Function of space and time
\mathcal{K}	Krylov subspace

NOMENCLATURE

$\underline{\underline{X}}_j$	Krylov matrix
$\underline{\underline{F}}$	Matrix of companion type
$\underline{\underline{R}}$	Residuum matrix
$\underline{\underline{H}}$	Upper Hessenberg matrix
$\underline{\underline{\Sigma}}$	Matrix with all singular values on its diagonal
$\underline{\underline{\tilde{A}}}$	Reduced order system matrix
$\underline{\underline{Q}}$	Orthonormal basis
$\underline{\underline{U}}$	Left hand side orthogonal matrix of SVD
$\underline{\underline{V}}$	Right hand side orthogonal matrix of SVD
$\underline{\underline{X}}_1$	First snapshotmatrix
$\underline{\underline{X}}_2$	Second snapshotmatrix
$\underline{\underline{\phi}}$	Full state eigenvectors
$\underline{\underline{E}}$	Energy criteria
$\underline{\underline{\Lambda}}$	Matrix with all eigenvalues on its diagonal
$\underline{\underline{T}}$	Unitary transformation matrix
$\underline{\underline{\Phi}}$	Matrix with modes in columns
$\underline{\underline{Y}}$	Unitary transformed (compressed) snapshot matrix
$\underline{\underline{\Psi}}$	Matrix transform basis
$\underline{\underline{\Upsilon}}$	Almost unitary transform basis
$\underline{\underline{Y}}_1$	Compressed first snapshot matrix
$\underline{\underline{Y}}_2$	Compressed second snapshot matrix
$\underline{\underline{C}}$	Compression matrix
$\underline{\underline{H}}$	Size of compression
$\underline{\underline{y}}$	Compressed vector of state
$\underline{\underline{s}}$	Sparse vector of state
α	Amplitude
β	Amplitude growth factor
F	Frequency
v	Wave velocity

F_{SRI} Rotor-stator interaction frequency

GLOSSARY

- CCF** Cylinder in cross-flow.
- CFD** Computational fluid dynamics.
- CoSaMP** Compressive sensing matching pursuit (Algorithm).
- DFT** Discrete Fourier transformation.
- DMD** Dynamic mode decomposition.
- LES** Large eddy simulation.
- NSE** Navier-Stokes equations.
- ODE** Ordinary differential equation.
- PIV** Particle image velocimetry.
- POD** Proper Orthogonal Decomposition.
- RAM** Random access memory.
- RANS** Reynolds-averaged Navier-Stokes.
- SAS** Scale adaptive simulation.
- SVD** Singular value decomposition.
- URANS** Unsteady Reynolds-averaged Navier-Stokes.

INTRODUCTION

To design state of the art jet engines a deep insight into the dynamics of fluid flow is indispensable. A detailed understanding of turbulence and its implications is one of the keys to push fluid flow machinery towards more sustainability and less specific fuel consumption. Highly precise prediction of flow is hence necessary for the engineering processes in order to meet the demanding requirements. Strong flow dynamics are undesirable since they can cause noise, efficiency loss or high cycle fatigue¹ of structural components. With their identification and potential gain of information about energy, frequency and spatial extent their root cause can be revised or modifications can be made to shift dominant frequencies to avoid crucial resonances.

In general, fluid flow may vary in time and space whereat time-dependant variance is referred to as unsteady or dynamic. Most practical fluid flow applications feature time dependant flow behaviour. In jet engines these dynamics emerge from movable geometries or as self excited instabilities as a result of turbulence. The latter occurs within the most basic flow-geometry configurations. In applications such as turbofan jet engines with complex geometries, highly dynamical internal flows are always present. These flows occur in turbulent boundary layers, wakes of compressor- and turbine blades and vanes or in other shear flows such as the mixing layer between cold bypass- and hot core engine flow. Other sources of dynamic flows are obstacles in the flow such as steps and gaps, scoops, rivets and bolts in the boundary walls, disturbing the freestream. Dynamic flows emerging from turbulence have many facets and are therefore categorized. Turbulence is a multi-scale phenomena comprising a large variety of time and length scales. These

¹Material fatigue due to high frequent change in load stress.

scales are roughly divided into two parts, the large and the small scale structures. The latter behaves seemingly random and reaches down to a size where its turbulent kinetic energy is dissipated into heat. However large scale structures can be as large as the size of the geometry. Other than its counterpart, they can occur in an characteristic periodic manner [Holmes et al., 2012, p.5]. If so, they correlate in space and time and are therefore referred to as coherent structures. A typical well known examples is large vortex shedding in the wake of blunt bodies. As such, coherent structures can play a dominant role in turbulent wakes. To identify and gain information of the dynamics of coherent structures, time and space discretized flow data must be available.

The industrial standard to gain flow data is to simulate flow fields using the Reynolds-averaged Navier-Stokes (RANS) equations. This method has prevailed over others due to its high precision over time-consumption ratio. Using the RANS equations, the short time-consumption is the main benefit, since its precision is compared to more advanced methods rather small. This is due to the limitations of the RANS turbulence models and the fact that the averaging is not capable of educing time resolved processes. Formerly, flow dynamics were not efficient predictable and requirements of engine components were not as demanding as today. Therefore many extreme conditions were accepted by using steady, averaged data and adding relatively high safety factors². This way to design an engine guaranteed safe operations at the cost of over-sizing components and ultimately increasing weight. Today, in contrast, safety factors are decreased if possible. Therefore maximum stress loads on components must be evaluated precisely and peak states of dynamic flows have to be calculable. In order to achieve this, three dimensional methods for data acquisition are commonplace. Due to economical reasons most of the used methods are still steady, with the assumption of a time-independent, constant conditions of the flow. With the increase of computational power, unsteady methods continue to evolve economical compatible and are consequently used more frequent in industry to gain more knowledge about dynamics.

Due to their real-time resolution, unsteady methods increase the amount of data output drastically, leading to difficulties in their post-processing analysis. With modern approaches of turbulent simulation, time discrete data is acquired and coherent structures are contained in the data. Because of the data size a conventional investigation is exceedingly time-consuming, therefore it is desirable to use a post-processing tool that extracts important information about the flow in time and space, identify main dynamics of the flow or to resolve coherent structures spatially.

²Safety factor is the ratio between bearable stress of a component over its expected peak stress.

To overcome this dilemma, having precise but in-transparent results, methods have been developed to decompose time resolved flow fields into so called *modes*. These modes extract information about physical processes and dynamics of the flow-field and provide decomposed information compared to the vast amount of data from the simulation output. One of these data decomposition methods is the so called dynamic mode decomposition (DMD). In the work at hand the DMD is implemented and verified on several generic cases to prove its application. Advantages and flaws of modifications of the original algorithm are discussed and evaluated. A selected variation is then used to analyse the flow field of a 3D cylinder in a cross flow and a rotor from a gas turbine. The DMD provides a simplified model of the flow and combines features from other established methods. The main idea behind the decomposition was introduced by Rowley et al. in 2009 and Schmid in 2010. Ever since many variations of the DMD have been created and though it was invented for the use in fluid mechanics it is also applicable in many other disciplines with high dynamics like the financial market and the spread of pandemics.

The present work discusses the main algorithm behind DMD and shows advantages and flaws in its applications. Results of DMD analyses will be looked at closely. Therefore a artificial test case is investigated to compare decomposed results with predetermined parameters. Secondly, a computational reproduced wind tunnel flow experiment that was part of previous work in the department is looked at to verify mode detection with the experimental results. In a final step jet engine data is investigated to proof in the field capabilities of DMD.

In the following chapter, the foundation will be given to fully understand the main idea behind the DMD. Other important approaches to analyse time-resolved data are outlined before the theory of the DMD and some of its modifications will be discussed in detail. The subsequent chapter 3 provides additional information on the data that was analysed to clarify questions that that might arise in the scope of the fourth chapter. In chapter 4 the results of the DMD analysis are presented. A comparison in between the different DMD methods will be pointed out. Furthermore the general algorithms spectral limits and computational specifications are discussed. A resume with further outlook is given in the last chapter.

2.1 Important concepts

2.1.1 Coherent structures

The term coherent structure is a widely used term in fluid dynamics to describe the nature of turbulent flows. There is no clear-cut definition of the term itself, therefore the next paragraphs will provide an overview to clarify what coherent structures are.

To understand which structures are exactly defined as coherent structures, a closer look onto the turbulent spectrum is necessary. To obtain the turbulent spectrum, the flow is spatial Fourier transformed and the structures are ordered by their characteristic wavelength κ . Each wavelength contains different amount of Energy E_{TKE} as shown in figure 2.1 where the energy is specified logarithmic on the ordinate and the wave number logarithmic on the abscissa. The concept states that this spectrum is divisible into three ranges. The largest structures with small wavelengths are classified as the energy containing range. Their length scale is of order similar to the dimensions of the flow-geometry. The very smallest structures are in the dissipation range. Their size is of order of the Kolmogorov-size which is several times larger than the mean free path of the gas' molecules [Holmes et al., 2012, p.19]. In between is the inertial subrange located. The term cascade derives from the assumption that the large scale structures contain energy that they extract from the freestream and pass it on to smaller structures. This turbulent energy is so transferred by the inertial subrange into the dissipation range where the energy is dissipated into heat.

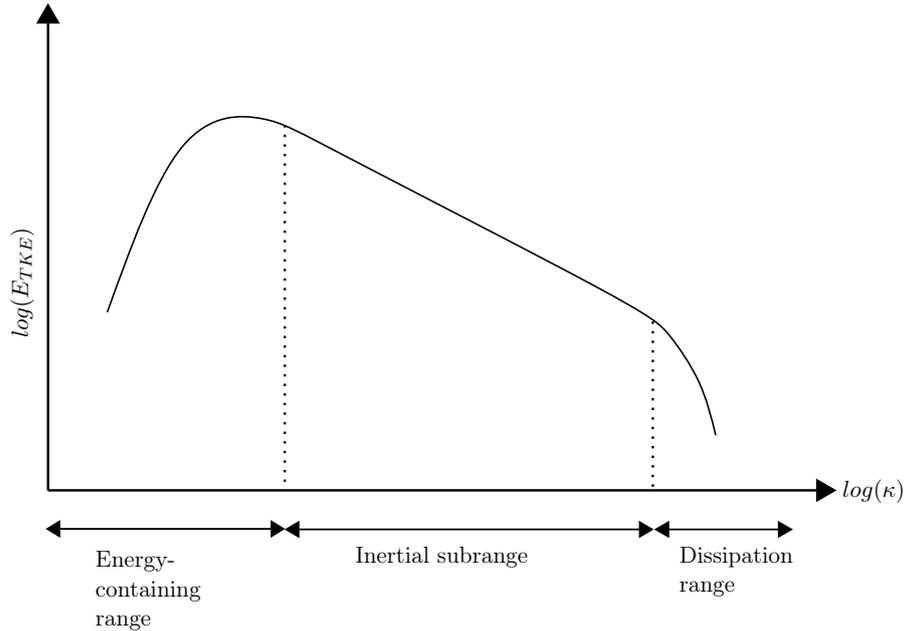


Figure 2.1: Turbulent spectrum of model flow with sufficient high Reynolds number.

The declaration of a structure as coherent is not trivial. Though certain attributes can be defined to categorize a structure as coherent. These are the following:

- large spatial size. In the energy cascade located in the energy containing range and typically in size of order of the geometry or the *integral length*.
- reoccurring, apparently periodic behaviour.
- remaining stable for a longer time than other turbulent structures.
- connected to a physical quantity, typically vorticity.

Rakishly said, coherent structures are large vertices that appear in an periodic manner with small turbulent fluctuations overlain. In figure 2.2 a side view of a mixing layer is shown where coherent structures are clearly visible as vortex pattern that emerges from the trailing edge of a flat plate upstream. Other examples are the vortex rings from a vortex canon or starting jet. Here the longevity of such a structure becomes descriptive since the structure stays stable from its discharge until its aim. Another

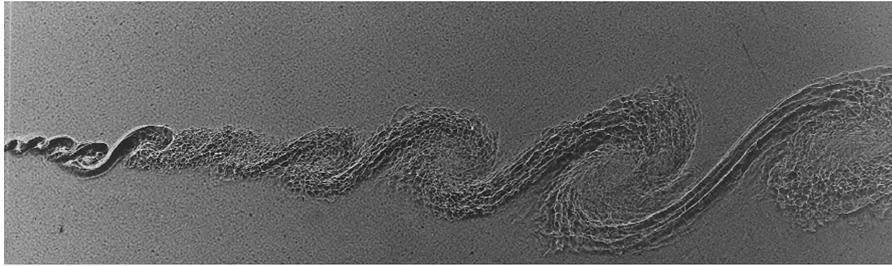


Figure 2.2: *Mixing layer of two flows with equal density. Velocity ratio = 0.38, Reynolds number = 10^6 . From Bernal et al. [1979].*

typical appearance are wingtip vortices from airplane wings or helicopter. The latter one is an example of how coherent structures can cause noise by rotor blades cutting through wingtip vortices from the preceding blade. This phenomena is known as blade slap. If turbulent structures are classified by their wave number, a large proportion is incoherent. Because of their chaotic behavior they are mainly describable by statistical means. However, because of their reoccurring nature coherent structures can be described with other mathematical approaches.

2.1.2 Modes

The term mode is widely used in this paper. It is part of the name of the main focus, the dynamic mode decomposition and therefore a short explanation shall be given.

In general, modes are solutions of oscillating systems in which a mode describes a spatially stationary oscillation. The system may consist of waves overlain, each characterised by its own wave properties such as a frequency, wave number, damping and amplitude. If the system is purely linear, the modes act as a superposition of the entire field. In fact, adding them up together gives a representation of the general field itself, a principle of linear systems. Modes show spatial patterns where a certain wave is dominant and thus information about the physical background of why the field behaves in its way can be derived.

A one dimensional example is a guitar string where waves travel along the string. It creates one single tone that is audible with a certain tonal character. In fact, this tone is a superposition of many sinusoidal waves [Feynman et al., 1962, chpt.49-2] and the magnitude of each existing frequency is determined by the length of the string, its strain and the resonance of the instruments body. If this field is decomposed, the modes will be the different sinusoidal standing waves on the string. If flow fields are decomposed into

modes, oscillations are spatially revealed. The number of modes and their properties will be unique to each system. The composition is sort of signature of the system itself. If for example boundary conditions are changed, this composition of modes and its properties will change. A requirement to extract modes is that there is some sort of oscillation in the system. In a turbulent flow field for example, small turbulent motion behaves chaotic. A modal decomposition of this strong non-linear motion is less meaningful since there is no harmonic oscillation.

2.2 Established methods for the identification of coherent structures

With the increase of data output from unsteady data measurements in the last decades many methods for analysing coherent structures have been developed. A good overview is provided by Taira et al.. This pool of methods is roughly dividable into two groups. One group is the time independent, non-dynamical oriented group. These methods such as Q-criterion or Lambda-2 criterion are based on analysing flow solutions from one point in time only by analysing fields of vorticity. With these tools it is possible to depict vertices in its spatial extends. Dynamical or spectral information is not gained. Another tool analysing spatial extends only is the spatial Fourier analysis. On the other hand side there are methods that take more than one point in time into account. They detect coherent structures by their dynamics - independent from the used flow variable. Most famous are the temporal Fourier decomposition and the proper orthogonal decomposition. Because the dynamic mode decomposition is related to them, a short outline shall be given in the sections below.

2.2.1 Proper orthogonal decomposition

One of the most common analysis techniques on unsteady flow data is the proper orthogonal decomposition (POD). It is also known as principal component analysis and is used in many subjects besides fluid dynamics like statistics or finance. There are many variations of this technique either as further development of the original algorithm or as basis of other techniques. Its mathematical main ingredient, the singular value decomposition (SVD), is also a main part of the DMD and as such POD modes are a sort of byproduct of the DMD. From the vast varieties of POD methods, the here outlined is referred to as *classical* or *spatial* POD.

The POD's aim is to decompose the fluctuating quantity \underline{x}'_t of a flow field. Since the

technique is focused on fluctuations only, the global offset or average of the data is not of interest. The mean time average \bar{x} is therefore subtracted from the original signal x_t

$$\underline{x}'_t = x_t - \bar{x}. \quad (2.1)$$

This fluctuating part is then decomposed into basis functions that sum up to best approximate the actual signal. These functions are, as the name of the technique implies, orthogonal. To quantify their size they are not only orthogonal but orthonormal, complemented by a scalar value a , quantifying their magnitude. Moreover, the corresponding scalar value a is time dependant and the orthonormal modes $\underline{\varphi}$ are space dependant

$$\underline{x}'_t = \sum_j a_{j,t} \underline{\varphi}_j. \quad (2.2)$$

The decomposition's objective is to find the vectors $\underline{\varphi}$ to receive a good approximation of the data vector \underline{x}' with the least amount of modes. This problem is solved by stacking all time discrete solutions or *snapshots* \underline{x} into a *snapshotmatrix* $\underline{\underline{X}}$ where all time discrete data is listed in column vectors. Therefore the series of time resolved data points, referred to as *snapshots* have to be restructured. These snapshots are the fields of the physical quantity respectively state variable x at a discrete time t_n . They can be one, two or three dimensional in space. In the here referred to as *snapshot vector*, each discrete point of the field at a time is stacked into a single vector, disregarding the dimensions of the original field (see Figure 2.3). With that in mind we can form the so called *snapshot-matrix*

$$\underline{\underline{X}} = \begin{pmatrix} | & | & \dots & | \\ \underline{x}'_1 & \underline{x}'_2 & \dots & \underline{x}'_N \\ | & | & \dots & | \end{pmatrix} \in \mathbb{R}^{M \times N} \quad (2.3)$$

where all the single snapshot vectors are aligned and each one of them fills one column of $\underline{\underline{X}}$. The number of rows M denotes the total number of spatial points and N stands for the total number of temporal points which is the number of snapshots.

From here the square covariance matrix $\underline{\underline{G}}$ ¹ is set up by

$$\underline{\underline{G}} = \underline{\underline{X}} \underline{\underline{X}}^T \in \mathbb{R}^{M \times M}. \quad (2.4)$$

¹The actual covariance matrix is divided by a factor: $\underline{\underline{X}} \underline{\underline{X}}^T / (1 - M)$ for the sake of readability this shall be omitted and we assume that this constant factor is part of the eigenvalue λ .

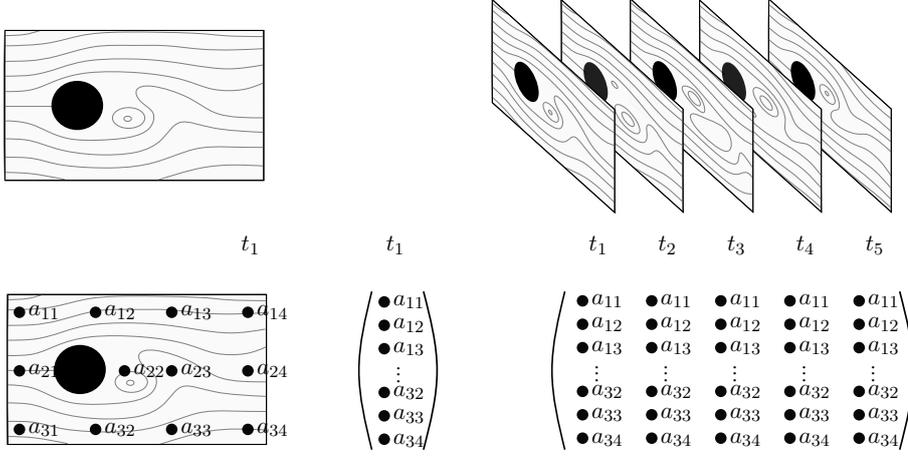


Figure 2.3: Sketch of flow field around a 2D Cylinder in a cross flow. Left to right, top to bottom: Flow field snapshot at $t = t_1$, collection of flow field snapshots, discretised flow field at $t = t_1$ with flow variable a , rearranged flow field in snapshot vector and rearranged snapshot collection in snapshot matrix.

In (2.4) the entries are built from multiplication of each point in time and space with each point in time and space. To receive the orthonormal vectors from above we need to solve the eigenvalue problem of

$$\underline{\underline{G}} \underline{\varphi}_j = \lambda_j \underline{\varphi}_j \quad (2.5)$$

with the systems eigenvalues λ_j . Solving the eigenpairs of a covariance matrix as it is in equation 2.4 and 2.5 is also the objective from the SVD where singular values σ and the above defined eigenvalues are related via

$$\lambda_j = \sigma_j^2.$$

A more detailed description of the SVD is given below in section 2.4.1. When applying classical POD, the resulting modes that are gained are the eigenpairs of the covariance matrix of the system and referred to as POD-modes.

The temporal coefficient from 2.2 are received by evaluating the dot product of the snapshot fluctuation \underline{x}'_t and the modes $\underline{\varphi}_j$

$$a_{j,t} = \langle \underline{x}'_t, \underline{\varphi}_j \rangle.$$

2.2.2 Time discrete Fourier transformation

The Fourier analysis is a well known mathematical tool to decompose signals from time to spectral domain. Some of its concepts are important for the DMD algorithm and therefore outlined in this section. In the methods of identifying coherent structures, Fourier transformations plays a major role since the reoccurring nature of coherent structures is good resolvable with a transformation that is based on trigonometric functions. There are four major classifications of Fourier transformation resulting from the nature of the signal. It can be a periodic or non-periodic signal and it can be discrete or continuous. Applied on fluid flow data the signal is non-periodic and discrete. This is tackled by so called discrete Fourier transformation (DFT).

The discrete Fourier coefficients \hat{x} , the hat marking signal in the spectral domain, are derived from the convolution of the signal x with *sin* and *cosin* functions. The trigonometric functions are displayed by Euler's formula $e^{i\omega} = \cos(\omega) + i\sin(\omega)$ and the integral of the continuous convolution in time N , becomes a sum over all discrete signal points

$$\hat{x}_k = \frac{1}{N} \sum_{n=0}^{N-1} x_n e^{-i\omega_k n}. \quad (2.6)$$

Here ω_k is k -times ($k = 0, 2, \dots, N - 1$) the fundamental frequency ω_0 . The fundamental frequency is the smallest frequency that is captured by the signal. In time discrete signals, this is the inverse of the measured period T and carried out as angular frequency with factor 2π , the fundamental frequency results in

$$\omega_0 = \frac{2\pi}{T} \quad (2.7)$$

and set into equation (2.6) the approximated Fourier coefficients are

$$\hat{x}_k = \frac{1}{N} \sum_{n=0}^{N-1} x_n e^{-ik \frac{2\pi}{N\Delta t} n}. \quad (2.8)$$

Suitable for linear algebra the sum from (2.8) can, by substituting

$$\gamma = e^{-i \frac{2\pi}{N\Delta t}},$$

be written as

$$\hat{\underline{x}} = \underline{\underline{\Gamma}} \underline{x}$$

with the square Fourier Matrix

$$\underline{\underline{\Gamma}} = \frac{1}{N} \begin{pmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & \gamma & \gamma^2 & \gamma^2 & \dots & \gamma^{(N-1)} \\ 1 & \gamma^2 & \gamma^4 & \gamma^6 & \dots & \gamma^{2(N-1)} \\ 1 & \gamma^3 & \gamma^6 & \gamma^9 & \dots & \gamma^{2(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \gamma^{(N-1)} & \gamma^{2(N-1)} & \gamma^{3(N-1)} & \dots & \gamma^{(N-1)(N-1)} \end{pmatrix}. \quad (2.9)$$

Spectral limits to DFT are the Fundamental frequency on the low end and Shannon-Nyquist criteria on the high end. The latter one states that if a signal is sampled with the sampling frequency

$$F_s = \frac{1}{\Delta t}$$

the highest frequency that can be captured is half the sample frequency

$$F_{max} = \frac{F_s}{2} = \frac{1}{2 \Delta t}. \quad (2.10)$$

Within the application of identifying coherent structures, the here shown time-discrete DFT is able to capture spectral information from **one** point in space and a series of time discrete points. Therefore spectral information is angular frequency ω . In fluid flow this data can be obtained by for example classic hot wire probe measurements or monitor probes in CFD calculations. As a different approach this DFT can also be carried out as spatial DFT. Then the data vector consists of equidistant spatial points at **one** point in time. The spectral information is thus the wavelength.

2.3 DMD Theory

2.3.1 General idea

The DMD is a tool to analyse dynamic systems that can generally be described through

$$\frac{d\underline{x}}{dt} = \mathbf{f}(\underline{x}, \underline{x}^{[II]}, \underline{x}^{[III]}, \dots, \underline{x}^{[N]}, t, \tau) \quad (2.11)$$

where the function \mathbf{f} describes the dynamics of a state vector \underline{x} and its higher order time derivatives $\underline{x}^{[II]}, \dots, \underline{x}^{[N]}$, at a time t , while τ denotes any other function inputs. In physics, the system dynamics \mathbf{f} can be such as the Schrödinger's equation in quantum mechanics, the Maxwell equations in electromagnetism or the Navier-Stokes equations (NSE) in fluid dynamics to name a few well known examples. All these systems are analyzable with DMD to extract spatial coherent structures - presupposed the systems are discretized. In the case of analysing structures in fluid flows the dynamical system relation is described by the NSE. Nevertheless DMD makes no use of the underlying equations. With its so called *equation free approach* only data has to be fed into the algorithm and no system equation has to be known. Though the NSE is are partial differential equations, low-dimensional flow behaviour can be described by ordinary differential equations (ODE) [Tu et al., 2014, p.1]. The DMD is particularly useful due to its ability to investigate nonlinear dynamics. The large scale flow phenomena mentioned above are often described through nonlinear ODEs. DMD is able to describe these nonlinear dynamics with an approximation through a set of linear ODEs. This is strongly related to the so called Koopman analysis. The Koopman operator is linear but infinite and is able to describe nonlinear finite operators of nonlinear dynamics such as the large scale flow phenomena described above. The DMD is hereby approximating this Koopman analysis by a linear but finite approximation. More details to the relation between DMD and Koopman analysis can be found in Kutz et al. [2016a] or Rowley et al. [2009]. Due to the close connection of DMD to linear ODEs, important concepts are pointed out in the following section.

2.3.2 Solution of linear ODE's of first order

In the following chapter linear ordinary differential equations (ODE) of first order will be discussed since they built the foundation of the DMD algorithm. This knowledge is fundamental to the theory of DMD and therefore explained before the actual algorithm will be discussed in section 2.4.

In physics, various problems are described by more than one equation. If the concerning equations are ODEs one speaks of a ODE system. These systems can be linear or non linear, they can contain derivations of different order and their coefficients can have various properties. Their linearity comes from the additivity and homogeneity of the equation's operators [Feynman et al., 1962, chpt.25-1]. Here we will focus on systems with constant coefficients, meaning that the coefficients in front of the derivations are no function of time or space - they remain constant. Within the pool of linear systems, the ODE's of first order containing only first derivatives, protrude. They follow straightforward rules for solving and their largest benefit is that higher order ODE's can always be converted into these systems [Boyce and DiParma, 2012, p.360]. Typical examples in physics where this scheme is used are to solve spring-mass systems and electrical resistor-capacitor-coil systems.

In the following section, the solution of a general linear systems of first order with constant coefficients, referred to as *continuous system*, will be discussed. In the subsequent part, the *discrete system* will be explained in detail. This focuses on the same systems but in a discretized time and space which is the way we come across when trying to understand the essence of the DMD.

Continuous system

A continuous linear system with constant coefficients is given by

$$\frac{d\underline{x}}{dt} = \underline{B} \underline{x} \quad (2.12)$$

where $\underline{x} \in \mathbb{R}^M$ describes the state vector, t the time and $\underline{B} \in \mathbb{R}^{M \times M}$ the systems operator with constant coefficients. Here, M denotes the number of equations. In addition, the equation is homogeneous since the disturbance function that could be added to the right hand side is zero.

The most simple example of such is the one dimensional system with $M = 1$, resulting in one scalar coefficient $b \in \mathbb{R}$. The solution to this example is given by:

$$\begin{aligned} \frac{dx}{dt} &= bx \\ x &= c e^{bt} \end{aligned} \quad (2.13)$$

with some integration constant $c \in \mathbb{R}$. The exponential term displays that for values of b that are positive, the system will increase with time and will hence become unstable. If

b is negative the system is asymptotically stable since values of x become smaller while propagating further in time [Boyce and DiParma, 2012, p.396].

The solution for a one dimensional system in (2.13) can be transferred onto a multidimensional system with $M > 1$ [Karpfinger, 2014, p.599]. A general solution for all M equations can be found by

$$\underline{x} = \tilde{\underline{c}} e^{\underline{B}t}. \quad (2.14)$$

with the integration constants contained in $\tilde{\underline{c}}$.

The solution is derived from the eigenvalue decomposition of the system matrix. Note that in this case we assume that the matrix \underline{B} is diagonalizable, meaning that their algebraic multiplicity is equal to the number of equations and with that, all M eigenvalues and eigenvectors are computable [Karpfinger, 2014, p.357]. With that in mind, \underline{B} is decomposable in it's diagonalized form

$$\underline{D} = \underline{\Xi}^{-1} \underline{B} \underline{\Xi} \quad (2.15)$$

where $\underline{\Xi}$ is a matrix with all eigenvectors of the system matrix saved in columns. Rearranging (2.15) and substituting it into the general system solution derived from (2.14) we receive

$$\underline{x}(t) = \tilde{\underline{c}} e^{\underline{\Xi} \underline{D} \underline{\Xi}^{-1} t}$$

where $\tilde{\underline{c}} \in \mathbb{R}^M$ is a vector that contains all constants. With the calculation rule for exponential equations

$$e^{\underline{\Xi} \underline{D} \underline{\Xi}^{-1}} = \underline{\Xi} e^{\underline{D}} \underline{\Xi}^{-1}$$

we receive

$$\begin{aligned} \underline{x}(t) &= \underline{\Xi} e^{\underline{D}} \underbrace{\underline{\Xi}^{-1} \tilde{\underline{c}}}_{\underline{c}} \\ \underline{x}(t) &= \underline{\Xi} e^{\underline{D}} \underline{c}. \end{aligned}$$

A solution for the constants in \underline{c} can be found by solving the equation in (2.18) for $t = 0$.

From this general solution in (2.14) we assume that

$$\underline{x}_j = \underline{\xi}_j e^{\mu_j t} \quad (2.16)$$

is one single solution with $j \in \{1, \dots, M\} \subset \mathbb{N} \setminus \{0\}$. To determine what μ_j and $\underline{\xi}_j$ are the single solution (2.16) is substituted into the system equation (2.12)

$$\frac{d}{dt}(\underline{\xi}_j e^{\mu_j t}) = \underline{B} \underline{\xi}_j e^{\mu_j t}.$$

Calculating the differential quotient on the left hand side and canceling the exponential functions on both sides the equation reduces to

$$\underline{B} \underline{\xi}_j = \mu_j \underline{\xi}_j. \quad (2.17)$$

reorganized we receive

$$(\underline{B} - \mu_j \underline{I}) \underline{\xi}_j = 0$$

with the identity matrix \underline{I} . The equation displays the solution for μ is found by finding the zeros of the characteristic polynomial of the linear operator \underline{B} [Karpfinger, 2014, p.356]. Therefore μ_j are (*continuous*) *eigenvalues* and $\underline{\xi}_j$ are (*continuous*) *eigenvectors* of the system operator \underline{B} [Boyce and DiParma, 2012, p.397]. The problem is linear and it's Wronskian determinate nonzero [Boyce and DiParma, 2012, p.403], meaning that the system's solutions are linear independent. The scalars μ_j and the vectors $\underline{\xi}_j$ form an eigenspace of the system operator. All solutions are linear independent and can hence be added to a general solution [Boyce and DiParma, 2012, p.391]

$$\underline{x}(t) = c_1 e^{\mu_1 t} \underline{\xi}_1 + c_2 e^{\mu_2 t} \underline{\xi}_2 + \dots + c_M e^{\mu_M t} \underline{\xi}_M \quad (2.18)$$

with the constants c_1, c_2, \dots, c_M , the eigenvalues $\mu_1, \mu_2, \dots, \mu_M$, and the eigenvectors $\underline{\xi}_1, \underline{\xi}_2, \dots, \underline{\xi}_M$. This general continuous solution is transferable onto discrete ODE systems. This is explained in the next section.

Discrete system

Like the continuous system discussed above, the system can also be displayed in discrete time and space. Hence the system equation for a discretized, linear, homogeneous ODE

system of first order with constant coefficients is

$$\underline{x}_{k+1} = \underline{A} \underline{x}_k \quad (2.19)$$

with $\underline{x}_k = \underline{x}(k\Delta t)$, $\Delta t = \text{constant}$, the discrete time index k and the operator $\underline{A} \in \mathbb{R}^{M \times M}$. Within the discrete formulation, values are defined at discrete times $t_k = k\Delta t$. Substituting this discrete formulation into (2.18) we receive

$$\underline{x}_k = \sum_{j=1}^M \underline{\varphi}_j e^{\mu_j k \Delta t} c_j \quad (2.20)$$

and for a propagated step in time $t = (k+1)\Delta t$ we receive

$$\begin{aligned} \underline{x}_{k+1} &= \sum_{j=1}^M \underline{\varphi}_j e^{\mu_j (k+1) \Delta t} c_j \\ &= \sum_{j=1}^M \underline{\varphi}_j e^{\mu_j k \Delta t} c_j e^{\mu_j \Delta t} \end{aligned} \quad (2.21)$$

$$= \sum_{j=1}^M \underline{x}_k e^{\mu_j \Delta t}. \quad (2.22)$$

The solution for the present time and the solution for one time step propagated are the same, except for one exponential term that shall be defined as

$$\lambda_j = e^{\mu_j \Delta t}. \quad (2.23)$$

Substituting (2.20) and (2.21) into the system equation while considering the recent definition we receive

$$\underline{A} \sum_{j=1}^M \underline{\varphi}_j \left(e^{\mu_j k \Delta t} c_j \right) = \sum_{j=1}^M \underline{\varphi}_j \left(e^{\mu_j k \Delta t} c_j \right) \lambda_j.$$

The scalar values in brackets cancel out of the vector equation and therefore we get

$$\underline{A} \sum_{j=1}^M \underline{\varphi}_j = \sum_{j=1}^M \underline{\varphi}_j \lambda_j. \quad (2.24)$$

Equation (2.24) has a solution when

$$\underline{\underline{A}} \underline{\varphi}_j = \underline{\varphi}_j \lambda_j, \quad j \in 1, \dots, M \quad (2.25)$$

applies. Therefore $\underline{\varphi}_j$ and λ_j are (*discrete*) *eigenvectors* respectively (*discrete*) *eigenvalue* of the system matrix $\underline{\underline{A}}$.

Let us assume $\underline{x}(t = 0) = \underline{x}_0$ is a known, initial condition. If the propagated time step is substituted by (2.22) including the definition of λ in (2.23) we receive

$$\begin{aligned} \underline{\underline{A}} \underline{x}_{k-1} = \underline{x}_k &= \sum_{j=1}^M \lambda_j \underline{x}_{k-1} \\ &= \sum_{j=1}^M \lambda_j^2 \underline{x}_{k-2} \\ &= \dots \\ &= \sum_{j=1}^M \lambda_j^k \underline{x}_0 \end{aligned}$$

including the initial condition \underline{x}_0 . The k -th time step is determined by

$$\underline{x}_k = \sum_{j=1}^M \lambda_j^k c_j \underline{\varphi}_j. \quad (2.26)$$

While the eigenvectors and eigenvalues are computable from the given system matrix the missing variables to receive a full general solution are the constants c_j . To receive the full solution, the constants can be acquired by evaluating (2.26) for $t = 0$. Since the time at the initial condition is defined to be zero, the exponential function in (2.20) becomes equal to one. Consequently the solution of the initial state is described as

$$\underline{x}_0 = \sum_{j=1}^M c_j \underline{\varphi}_j. \quad (2.27)$$

and can be solved by solving the minimization problem $\min \|\underline{x}_0 - \underline{\Phi} \underline{c}\|$ where $\underline{\Phi}$ is a matrix with all discrete eigenvectors $\underline{\varphi}_j$ in its columns.

Comparing the continuous solution in (2.18) and the discrete solution in (2.20) it reveals that the eigenvectors are the same in both derivations. For continuous eigenvalues μ the

relation

$$\mu_k = \frac{\ln(\lambda_k)}{\Delta t}$$

is derived from (2.23).

2.4 DMD algorithm

When analysing the NSE, the physical quantities that can be analyzed are the flow variables such as pressure, velocity, temperature and other quantities that are derived from them such as total pressure or the magnitude of the Mach number. To capture the dynamics of coherent structures, spatial high resolved data it is commonly acquired by the use of computational fluid dynamics (CFD) computations or experimental data from particle image velocimetry (PIV). Since the latter one is capable of capturing velocity fields only, the usual quantities to analyse are therefore velocity fields or derived vorticity fields [Kutz et al., 2016a, p.28]. Both ways of acquiring fluid data usually provide a number of spatial data points M that is of several magnitudes higher than temporal data points N . Therefore it is reasonable to assume that $M \gg N$. In order to handle the data at hand it is, in the same manner as displayed in figure 2.3, set up into a *snapshot-matrix*

$$\underline{\underline{X}} = \begin{pmatrix} | & | & \dots & | \\ \underline{x}_1 & \underline{x}_2 & \dots & \underline{x}_N \\ | & | & \dots & | \end{pmatrix} \quad (2.28)$$

where all the single snapshot vectors are aligned and each one of them fills one column of $\underline{\underline{X}}$. The number of rows M denotes the total number of spatial points and N stands for the total number of temporal points which is the number of snapshots.

With the data at hand in snapshot vectors we can set up a linear system that evolves in time

$$\underline{x}_{k+1} = \underline{\underline{A}} \underline{x}_k. \quad (2.29)$$

This system maps the initial state at the time $t = t_k$ onto its future state at $t = t_{k+1}$ with the constant system operator $\underline{\underline{A}}$. This is a reduced form of the general dynamic system in (2.11) with the assumptions of a discrete, linear system with constant coefficients. A solution of such a system was derived in section 2.3.2, showing that the eigenvalues and eigenvectors of the system are a solution. To understand the underlying system behavior

it is necessary to extract these eigenpairs of $\underline{\underline{A}}$.

A closer look onto the dimensions of the system equation in (2.29) reveals that the system operator is of size $\underline{\underline{A}} \in \mathbb{R}^{M \times M}$ equal to the number of spatial points squared. As mentioned above, the number of spatial points is usually of high order compared to the number of temporal points. CFD computations for example are using computation grids that contain large number of spatial points. A high resolution, transient CFD calculation with a high resolved turbulence model can for example easily consist of ten or hundred millions of spatial points while the temporal range of interest for the number of points is often only a few hundred points [Kutz et al., 2016a, p.28]. This results in huge data matrices that need high computational effort to be decomposed. Apart from the computational effort to solve the system, also storage space on the computing machine can be a limited factor to high spatial resolved data. In order to minimize computational effort and storage space it is necessary to reduce the size of the system while obtaining the leading eigenvalues and eigenvectors from the full system. This reduction is closely linked to the Arnoldi algorithm [Rowley et al., 2009, p.1] which is discussed in detail in appendix A.2.

2.4.1 standard DMD

The algorithm explained in the following section is the basic idea behind the DMD method. It was introduced in the fluid dynamics community by Schmid in 2010. The algorithm of the basic idea will be referred to as *standard algorithm*.

The main procedures of the algorithm is categorized in five main steps. At first an existent set of spatial and time resolved data in form of a snapshot-matrix as in (2.28) is preprocessed to define the linear system. This is followed by the second step, the recreation of the system operator and the third step which is the reduction of the system operator. The fourth step is to compute the eigenpairs of the obtained system and at last the spatial modes approximations of the full system are reconstructed in a fifth step.

To set up a linear system equation with more than two snapshots \underline{x}_{n+1} and \underline{x}_n (compare (2.19)), the given snapshot matrix $\underline{X} \in \mathbb{R}^{M \times N}$ is transformed into two slightly different matrices. One containing all but the last snapshot and one containing all but the first

snapshot, denoted by the subscripts 1 and 2

$$\underline{\underline{X}}_1 = \begin{bmatrix} \underline{x}_1 & \underline{x}_2 & \cdots & \underline{x}_{N-1} \end{bmatrix} \in \mathbb{R}^{M \times N-1} \quad (2.30)$$

$$\underline{\underline{X}}_2 = \begin{bmatrix} \underline{x}_2 & \underline{x}_3 & \cdots & \underline{x}_N \end{bmatrix} \in \mathbb{R}^{M \times N-1} \quad (2.31)$$

where N denotes the total number of snapshots. The matrix $\underline{\underline{X}}_1$ will be referred to as *first snapshot matrix* and $\underline{\underline{X}}_2$ as *second snapshot matrix*. With these matrices the basic linear system equation (2.29) can be rewritten as

$$\underline{\underline{X}}_2 = \underline{\underline{A}} \underline{\underline{X}}_1. \quad (2.32)$$

Comparing this notation of the linear system with the definition of the Krylov matrix in (A.1) it is visible that the first snapshot matrix spans the Krylov subspace of $\underline{\underline{A}}$. The Krylov subspace is related to the Arnoldi Algorithm to approximate leading eigenvalues of a matrix. A basis of it can be used to project the matrix into a smaller matrix that contains the leading eigenvalues of the original matrix. See appendix A.2 for more details. Equation (2.32) is set to find a basis of the first snapshot matrix in order to reduce the system operator $\underline{\underline{A}}$. Since $\underline{\underline{X}}_1$ is of size $M \times (N - 1)$, the reduced system operator will be of size $(N - 1) \times (N - 1)$. To project $\underline{\underline{A}}$ onto such a Krylov basis it is necessary to compute $\underline{\underline{A}}$ in first place.

To solve the system equation for the unknown operator it is necessary to compute the inverse of the first snapshot matrix to receive $\underline{\underline{A}} = \underline{\underline{X}}_2 \underline{\underline{X}}_1^{-1}$. But since $(N - 1) \ll M$ the snapshot matrices have more rows than columns. They are so called *tall and skinny*. For such non-square matrices there exists no inverse [Karpfinger, 2014, p.79]. Instead, the system can be solved by computing the pseudo inverse of the first snapshot matrix. The pseudo inverse can be computed by $\underline{\underline{X}}_1^\dagger = (\underline{\underline{X}}_1^T \underline{\underline{X}}_1)^{-1} \underline{\underline{X}}_1^T$ (see appendix A.1) or with the use of the Singular Value Decomposition (SVD). Therefore the matrix is split into a unitary square matrix $\underline{\underline{U}}$, a matrix $\underline{\underline{\Sigma}}$ with all singular values σ_{ij} on it's diagonal and the hermetian transposed ² $\underline{\underline{V}}^H$

$$\underline{\underline{X}}_1 = \underline{\underline{U}} \underline{\underline{\Sigma}} \underline{\underline{V}}^H. \quad (2.33)$$

In this case the SVD used is the so called *economy SVD* which, in case of $M > N - 1$, does not save unused columns of $\underline{\underline{U}}$. Also the rows that are zero in $\underline{\underline{\Sigma}}$ are ignored as well.

²The hermetian transposed is the transposed and complex conjugated matrix.

Compare figure 2.4 top to bottom.

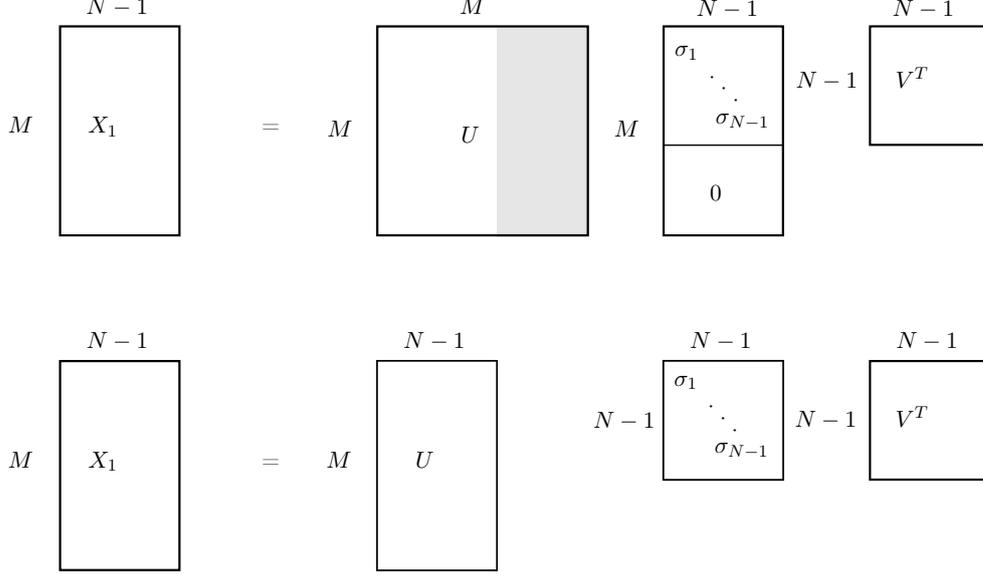


Figure 2.4: Dimensions of singular value decomposition for $M > N-1$. Top: Regular algorithm where grey part of U is not having any effect since it is multiplied by the zero entries of Σ . Bottom: economy SVD with reduced dimensions of U and Σ

To solve the system equation in (2.32) for the operator $\underline{\underline{A}}$ the pseudo-inverse of the first snapshot matrix is received via SVD. Therefore we write

$$\begin{aligned} \underline{\underline{X}}_2 &= \underline{\underline{A}} \underline{\underline{X}}_1 \\ \underline{\underline{X}}_2 &= \underline{\underline{A}} \underline{\underline{U}}^1 \underline{\underline{\Sigma}} \underline{\underline{V}}^H. \end{aligned} \quad (2.34)$$

From this point we can multiply $\underline{\underline{V}}$, $\underline{\underline{\Sigma}}^{-1}$ and $\underline{\underline{U}}^H$, the inverse matrices of the SVD, from the right side

$$\begin{aligned} \underline{\underline{X}}_2 \underbrace{\underline{\underline{V}} \underline{\underline{\Sigma}}^{-1} \underline{\underline{U}}^H}_{\underline{\underline{X}}_1^\dagger} &= \underline{\underline{A}} \underbrace{\underline{\underline{U}} \underline{\underline{\Sigma}} \underline{\underline{V}}^H \underline{\underline{V}} \underline{\underline{\Sigma}}^{-1} \underline{\underline{U}}^H}_{\underline{\underline{I}}} \\ \underline{\underline{X}}_2 \underline{\underline{V}} \underline{\underline{\Sigma}}^{-1} \underline{\underline{U}}^H &= \underline{\underline{A}} \end{aligned} \quad (2.35)$$

so the right hand side of the equation reduces to the system operator times the identity matrix. Note that $\underline{\underline{U}}$ and $\underline{\underline{V}}$ are by the definition of the SVD [Karpfinger, 2014, p.392] orthogonal matrices and therefore its inverses are equal to it's transposed $\underline{\underline{U}}^{-1} = \underline{\underline{U}}^T$ and

their transposed times themselves is equal to the identity matrix $\underline{\underline{V}}^T \underline{\underline{V}} = \underline{\underline{I}}$ [Karpfinger, 2014, p.137]³. Hence there is only one, computational expensive matrix inversion of $\underline{\underline{\Sigma}}$ to perform. But since $\underline{\underline{\Sigma}}$ is an diagonal matrix the inversion is cheap compared to dense matrices.

For the above derived matrix $\underline{\underline{A}}$, order reduction is achieved by using the eigenvalue approximation based on the Arnoldi algorithm, where the base of the Krylov subspace is derived via SVD. Reviewing this decomposition as described in (A.8), it is pointed out that the first r columns of $\underline{\underline{U}}$, forming the orthonormal base, are the same as the computed $\underline{\underline{U}}$ by the economy SVD in (2.33). Hence the base to project $\underline{\underline{A}}$ onto its Krylov subspace has already been computed by (2.33). Resulting in no extra computation effort for receiving the necessary base of the Krylov subspace. Substituting the system equation in (2.35) into the system projection in (A.9) we receive the system of reduced order

$$\begin{aligned} \underline{\underline{\tilde{A}}} &= \underline{\underline{U}}^T \underline{\underline{A}} \underline{\underline{U}} \\ &= \underline{\underline{U}}^T \underline{\underline{X}}_2 \underline{\underline{V}} \underline{\underline{\Sigma}}^{-1} \underline{\underline{U}}^T \underline{\underline{U}} \\ &= \underline{\underline{U}}^T \underline{\underline{X}}_2 \underline{\underline{V}} \underline{\underline{\Sigma}}^{-1}. \end{aligned} \tag{2.36}$$

Note that the computation of the full system matrix of size $M \times M$ is not necessary, as $\underline{\underline{\tilde{A}}}$ in (2.36) is received by the second snapshot matrix and the components from the SVD of the first snapshot matrix. This means that since the SVD is used for the recreation of the operator $\underline{\underline{A}}$ and for the basis of the Krylov subspace $\underline{\underline{U}}$, these two steps are actually not separated within the algorithm. They are proceeded at once. This is also beneficial for the storage space of the used computing machine since no large $M \times M$ matrix has to be saved. Furthermore, the use of the economy SVD leads to the fact that no matrix of the actual algorithm exceeds the dimensions $M \times N$. A look at the resulting reduced system $\underline{\underline{\tilde{A}}}$ shows that it is of size $(N - 1) \times (N - 1)$ instead of $M \times M$. In practical applications where many more spatial than temporal points $M \gg N$ are common, this projection can easily reduce the number of matrix entries by more than 99%⁴.

The eigenvalues and eigenvectors of the $\underline{\underline{\tilde{A}}}$ are received by using general eigenvalue algo-

³The use of the economy SVD is giving a non-square and therefore non-orthogonal matrix U . Because we still multiply it with Σ we can use the orthogonal properties and then ignoring the columns that become zero when multiplying with Σ .

⁴Reduction of matrix entries is given by $100 \cdot (1 - \frac{(N-1)^2}{M^2})\%$. A considered "spatial coarse resolved" 2D CFD computation was performed having 45000 spatial points and 120 temporal, shrinking the number of matrix entries from $\underline{\underline{A}}$ to $\underline{\underline{\tilde{A}}}$ by 99.99929%.

rithms that solve

$$\underline{\tilde{A}} \underline{\varphi}_i = \lambda_i \underline{\varphi}_i$$

to receive all eigenvectors $\underline{\varphi}_i$ and eigenvalues λ_i . The eigenvalues approximate a set of $(N - 1)$ of the total M eigenvalues of the full system. Since these eigenvalues are an approximation of the full state eigenvalues they will be referred to as *Ritzvalues* which derives from the Rayleigh–Ritz method, another algorithm to find approximated eigenvalues of a matrix. The full system eigenvectors, referred to as modes, do not coincide with the ones from the reduced system. They need to be calculated differently.

In the standard algorithm the full state modes $\underline{\phi}_i \in \mathbb{C}^{M \times 1}$ are received by a transformation of the reduced eigenvectors by the matrix \underline{U}

$$\underline{\phi}_l = \underline{U} \underline{\varphi}_l \quad (2.37)$$

where $l = 1, \dots, N - 1$ [Schmid, 2010, p.9]. Other possibilities of this step will be discussed in the following chapter.

2.4.2 Mode order

An important issue when handling DMD analysis is the fact that the way of how modes are ordered according to their importance is not predetermined. A common way is to define the order according to their scale value c from (2.27). This value weighs the importance of each mode on its impact to recreate the first snapshot. In this order the Ritzvalue of the corresponding mode is not taken into account. Within in the pool of modes there are some modes that might have a large scale value but decay fast according to their small Ritzvalue. Often, these modes are physically less important and so the approach to order modes after their scale value is not satisfying. Therefore, another method was introduced by [Tissot et al., 2014, p.412] by defining an energy criterion that includes mode magnitude, frequency and damping factor

$$E_i = \|\phi_i\|^2 \frac{e^{2\tilde{\mu}_i T} - 1}{2\tilde{\mu}_i T}. \quad (2.38)$$

This equation (2.38) includes a factor similar to the continuous eigenvalue derived from the absolute value of the discrete eigenvalue $\tilde{\mu}_i = \frac{\log(|\lambda_i|)}{\Delta t}$ and the measurement period $T = (N - 1) \Delta t$.

2.5 Alternatives of standard algorithm

Since the introduction of DMD at the end of the two-thousands many varieties and modifications of the original algorithm have been published. Algorithms such as *exact DMD* Tu et al. [2014] show improvements in precision of the mode creation whereas others like *randomized DMD* Erichson et al. [2019] or *compressed DMD* focus on the improvement of speed. The *compressed sensing DMD* Brunton et al. [2015] is an algorithm that identifies modes on heavily sub sampled data. Not only speed and completion are in the focus, there is also a modification known as *multi resolution DMD* Kutz et al. [2016b] that enhances the identification of transient phenomena or *DMD with control* Proctor et al. [2016] that includes a control to optimize the dynamics of the underlying system. To enhance the identification of coherent structures in jet engine components some of these algorithms have been investigated and are explained below.

2.5.1 exact DMD

The *exact DMD* algorithm is mainly the same algorithm as the above derived *standard DMD* except for the creation of the modes. The latter one recreates the full state modes from an projection of the eigenvectors of the reduced system onto the POD modes as shown in equation (2.37). Tu et al. derived another way to compute modes by

$$\underline{\phi}_i = \frac{1}{\lambda_i} \underline{X}_2 \underline{V} \underline{\Sigma}^{-1} \underline{\varphi}_i \quad (2.39)$$

referred to as *exact* modes. Compared to the original definition these modes lie in the span of \underline{X}_2 instead of the span of \underline{X}_1 [Tu et al., 2014, p.7]. If the last snapshot \underline{x}_n in (2.31) is a linear combination of all previous snapshot then the exact modes and the standard or projected modes are the same [Tu et al., 2014, p.8].

2.5.2 compressed DMD and compressed sensing DMD

The standard DMD algorithm is computational efficient due to the use of the truncated SVD and the prevention of handling matrices of size $M \times M$. With higher fidelity of the data, whereas in CFD computations especially the spatial discretization can reach tens or hundreds of million points, the SVD can become a bottleneck in the computation. The compressed DMD algorithm uses a way of compressing data before handling it into the exact DMD algorithm. Subsequently, the DMD is computed on the compressed data before modes are, in the last step, decompressed to the original full state size.

The referred to as compressed sensing DMD algorithm uses the same background of compression but here the initial situation is such that the data is not at hand in its full state but rather a sub-sampled measurement. In applications this is the case when for example experimental measurements are expensive and a full state is not available. Then compressed sensing is reconstructing full state modes form these sub-sampled data. The background in both cases, the compression, is similar to procedures known from audio compression format *mp3* or picture compression format *jpeg* [Kutz et al., 2016a, p.134].

The basic idea behind both approaches is that the DMD algorithm is invariant to right and mostly invariant to left unitary transformations ⁵. In case of a right unitary transformations where for example rows of each snapshot matrix \underline{X}_1 and \underline{X}_2 are swapped in the same manner, the output of the DMD is the same. Let us assume we have a unitary transform basis $\underline{T} \in \mathbb{R}^{M \times M}$ then we receive the following equations, where \underline{Y} is the transformed data and the index x identifies matrices derived from the original data \underline{X} respectively the index y identifies matrices derived from the transformed data \underline{Y}

$$\underline{Y}_1 = \underline{T} \underline{X}_1 \quad (2.40)$$

$$\underline{Y}_1 = \underline{T} \underline{U}_{\underline{x}} \underline{\Sigma}_{\underline{x}} \underline{V}_{\underline{x}}^H.$$

If this transformed data is the input for the DMD algorithm, [Kutz et al., 2016a, p.143] show that the resulting eigenvalues and eigenvectors are related via

$$\underline{\Lambda}_{\underline{y}} = \underline{\Lambda}_{\underline{x}} \quad (2.41)$$

$$\underline{\Phi}_{\underline{y}} = \underline{T} \underline{\Phi}_{\underline{x}}, \quad (2.42)$$

where the matrices $\underline{\Lambda}$ represent diagonal matrices with the eigenvalues on its diagonal. Note that the modes $\underline{\Phi}_{\underline{y}}$ in (2.42) derived from the transformed data \underline{Y} are the transformed modes from the original modes $\underline{\Phi}_{\underline{x}}$. Therefore the above stated DMD is *mostly* invariant to left unitary transformations [Kutz et al., 2016a, p.142].

Equations (2.40) - (2.42) show that we can transform our data and extract mostly the same information as with non-transformed data. Since the transform basis is of size $M \times M$ there has no compression been so far though. For this another principle has to be derived.

In an appropriate basis, most data is sparse, meaning that the majority of vector entries

⁵Unitary transformations are transformations that preserve the inner product. For a unitary matrix \underline{T} applies: $\underline{T}^H \underline{T} = \underline{I}$ with \underline{I} representing the identity matrix.

is zero. In some transform basis $\underline{\Psi}$ this leads to

$$\underline{x} = \underline{\Psi} \underline{s} \quad (2.43)$$

where \underline{s} is the sparse data vector. An example for a transform basis is the inverse of the Fourier matrix in (2.9), where the entries of the sparse vector would be the Fourier coefficients.

Assuming we have data \underline{x} at hand which is sparse in some transform basis, we can compress it with a compression matrix $\underline{C} \in \mathbb{R}^{M \times H}$ where the dimension $H < M$ is chosen by the user to determine the ration of the compression. It defines the size of compressed data vectors $\underline{y} \in \mathbb{R}^H$

$$\begin{aligned} \underline{y}_i &= \underline{C} \underline{x}_i \\ \underline{y}_i &= \underbrace{\underline{C} \underline{\Psi}}_{\underline{\Upsilon}} \underline{s}_i. \end{aligned} \quad (2.44)$$

If the product of compression matrix and transform basis fulfill the so called *restricted isometry property* then the resulting matrix $\underline{\Upsilon}$ is an almost unitary transform to the sparse vector \underline{s} [Kutz et al., 2016a, p.135]. This is the case when the compression matrix \underline{C} is incoherent to the transform basis $\underline{\Psi}$. In theory, this applies when the rows of the compression matrix \underline{C} are uncorrelated to the columns of the transform basis $\underline{\Psi}$. Suggestions for the compression matrix are three different ways that are in most cases sufficiently enough incoherent with the transform basis:

- Random projection matrix with random entries.
- Gaussian random projection matrix where entries in its rows are Gaussian distributed.
- Single pixel measurement where rows are zero except for one entry that is one.

Compressed DMD works with the assumption that the data \underline{X}_1 and \underline{X}_2 are sparse in some basis. Then the compression matrix \underline{C} can be inserted in (2.44) instead of the transform basis \underline{T} . Note that \underline{C} itself is no unitary transform basis but with the product of \underline{C} and \underline{X}_1 it is a unitary transform basis to the sparse representation of \underline{X}_1 . The transform basis $\underline{\Psi}$ is therefore not needed nor calculated in this approach. With the invariance of DMD to left and right unitary transformations, the compression matrix

can be set into the system as

$$\begin{aligned}\underline{Y}_1 &= \underline{C} \underline{X}_1 \\ \underline{Y}_2 &= \underline{C} \underline{X}_2.\end{aligned}$$

Executing the DMD with the compressed data \underline{Y} we receive the eigenvalues $\underline{\Lambda}_y$ and the modes $\underline{\Phi}_y$. From (2.41) we see that the eigenvalues are the same as the one from the non-compressed system. The eigenvectors of the full system $\underline{\phi}_{i,x}$ can be recreated, similar to (2.39), from

$$\begin{aligned}\underline{\phi}_{i,y} &= \frac{1}{\lambda_i} \underline{Y}_2 \underline{V}_y \underline{\Sigma}_y^{-1} \underline{\varphi}_{i,y} \\ \underline{C} \underline{\phi}_{i,x} &= \frac{1}{\lambda_i} \underline{C} \underline{X}_2 \underline{V}_y \underline{\Sigma}_y^{-1} \underline{\varphi}_{i,y} \\ \underline{\phi}_{i,x} &= \frac{1}{\lambda_i} \underline{X}_2 \underline{V}_y \underline{\Sigma}_y^{-1} \underline{\varphi}_{i,y}\end{aligned}\tag{2.45}$$

Compressed sensing DMD is based on the same idea that the original data is sparse in some basis. The initial difference to start with is that not the original full state data \underline{X} is at hand but some sub-sampled data \underline{Y} . This sub-sampled data can be inserted into the DMD algorithm to acquire eigenvalues and sub-sampled eigenvectors, but the recreation of full state modes as in (2.45) is not possible since \underline{X}_2 is not at hand. This is solved by the use of an algorithm that seeks for the sparsest solution

$$\underline{s}_{min} = \underset{s}{\operatorname{argmin}} \|\underline{s}\|_0 \quad \text{solving} \quad \underline{y} = \underline{C} \underline{\Psi} \underline{s}.\tag{2.46}$$

In this case $\|\underline{s}\|_0$ is a l_0 quasi norm [Needell and Tropp, 2009, p.302] with least amount of nonzero entries to fulfill the equation for the sub-sampled data in (2.44). This optimization in (2.46) is non-convex and therefore only solvable by brute force. Hence algorithms such as the *compressive sensing matching pursuit* (CoSaMP) were developed using a l_1 minimization. This algorithm results with high probability in the solution of (2.46) approaching the sparse vector \underline{s}_{min} if the restricted isometry principle is applicable to $\underline{C} \underline{\Psi}$ and the number of sub-sampled measurements H sufficiently large enough.

The sub-sampled modes and the full state modes are related via

$$\underline{\Phi}_y = \underline{C} \underline{\Phi}_x.\tag{2.47}$$

Since our data is assumed to be sparse in some basis and the full state eigenvectors are

a solution, there exists also a sparse solution of eigenvectors $\underline{\underline{\Phi}}_s$

$$\underline{\underline{\Phi}}_x = \underline{\underline{\Psi}} \underline{\underline{\Phi}}_s. \quad (2.48)$$

With the compressed representation in (2.47) and the sparse representation of data in (2.48) the sparse modes are a function of

$$\underline{\underline{\Phi}}_s = (\underline{\underline{C}} \underline{\underline{\Psi}})^{-1} \underline{\underline{\Phi}}_y.$$

The solution for $\underline{\underline{\Phi}}_s$ is achieved by inserting $(\underline{\underline{C}} \underline{\underline{\Psi}})^{-1}$ and $\underline{\underline{\Phi}}_y$ into the CoSaMP-algorithm to find a sparse solution. In a last step the full state modes are created by (2.48).

DATA, METHOD, MEASUREMENT

In the work at hand, the application of DMD is proceeded in three steps. At first, the scripts for standard DMD, exact DMD, compressed DMD and compressed sensing DMD are tested on the here referred to as *artificial flow field*. In this step an artificial field with imposed oscillations is set up, following an analysis to detect mode characteristics and to compare them to predetermined field parameters. Subsequent, the methods are in the following step used to analyse a three dimensional flow field around a cylinder in a cross flow. This data was acquired by a previous project on SAS validation in the department by Schreiber. Based on an experiment by Ackerman et al., this data is used to verify DMD on actual flow data. In a final step, data from a jet engine component is analysed. In this chapter the origin of the analysed data shall be discussed.

3.1 Artificial flow field

As a verification test to evaluate mode characteristics, a simple, two-dimensional field is created with several waves overlain. The aim is to identify the predetermined wave properties correctly. Especially the detection of predetermined frequencies and amplification factors is investigated. Furthermore, DMDs capability of resolving each given wave as mode under varying circumstances will be discussed. Therefore a baseline data set #1 is created. In order to check different characteristics of modes this baseline data-set is further slightly modified and labeled as set #2, #3 and #4.

Data set #1 is the baseline for all other sets of the artificial flow field. The domain for the baseline and all other data sets are chosen to range from $-\pi$ until π meters

in both x - and y -direction. The x -direction is divided into 100 equidistant points and the y -direction in 150 equidistant points, resulting in a total number of 15000 spatial points. The data set consists of 101 time steps with a time delta in between snapshots of $\Delta t = 0.025s$, resulting in an overall measurement period of $T = 2.5s$. In this field five different waves w_n are overlain:

1. One dimensional wave traveling in $+x$ direction

$$w_{n,1}(x, y, t_n) = \alpha_1 \beta_1^{t_n} \sin(\kappa_1 x - \omega_1 t_n)$$

2. One dimensional wave traveling in $+y$ direction

$$w_{n,2}(x, y, t_n) = \alpha_2 \beta_2^{t_n} \sin(\kappa_2 y - \omega_2 t_n)$$

3. One dimensional wave traveling in $-x$ direction

$$w_{n,3}(x, y, t_n) = \alpha_3 \beta_3^{t_n} \sin(\kappa_3 x + \omega_3 t_n)$$

4. Two dimensional, spherical wave with center at $[\frac{\pi}{2}, \frac{\pi}{2}]$

$$w_{n,4}(x, y, t_n) = \alpha_4 \beta_4^{t_n} \sin(\kappa_4 \sqrt{(x - \frac{\pi}{2})^2 + (y - \frac{\pi}{2})^2} - \omega_4 t_n)$$

5. Two dimensional structure traveling in $+y$ direction

$$w_{n,5}(x, y, t_n) = \alpha_5 \beta_5^{t_n} (\sin(\kappa_5 x) + \sin(\kappa_5 y + \omega_5 t_n))$$

where α represents the amplitude, β the amplitude growth value, κ is the wavelength and ω the angular frequency. The values for the wave properties are given in table 3.1. To receive the final field w_n these five wave sets are added up together as following

$$w_n = w_{n,1} + w_{n,2} + w_{n,3} + w_{n,4} + w_{n,5} + noise .$$

The last entry *noise* being a random noise added with a maximum magnitude of 10% of the average wave field amplitude $((\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5)/5)$.

	α	β	κ	ω	\mathbf{F}	\mathbf{v}
	[m]	[-]	[1/m]	[1/s]	[Hz]	[m/s]
$w_{n,1}$	5	1	1	3.142	0.5	3.142
$w_{n,2}$	0.1	1	10	12.566	2	1.257
$w_{n,3}$	20	1	5	31.416	5	6.283
$w_{n,4}$	4	1	19	100.531	16	5.291
$w_{n,5}$	1	1	5	113.097	18	22.619

Table 3.1: Wave parameters of first data set of the artificial flow field. Angular frequency and velocity are derived from wave-number and frequency.

In figure 3.1, each wave as well as their superposition as artificial flow field, including the

noise are displayed at $t = 0s$.

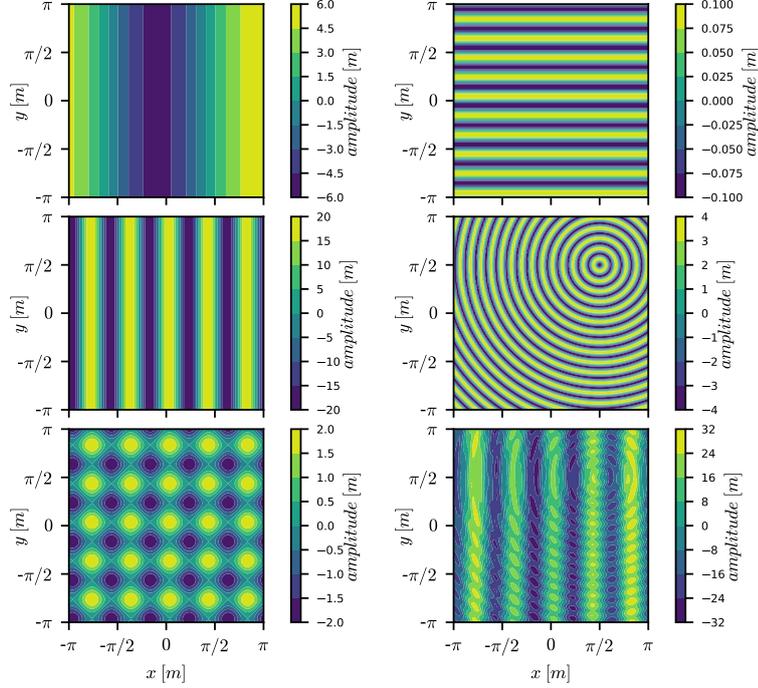


Figure 3.1: Overview of data set #1. Left to right, top to bottom: Wave $w_{0,1}$, wave $w_{0,2}$, wave $w_{0,3}$, wave $w_{0,4}$, wave $w_{0,5}$ and the superposition of all including noise at $t = 0s$.

Data set #2 is the same as the baseline except for the spatial resolution of points. While the domain range is the same, x - and y -direction are both resolved with 80 points. This reduces the number of points by 57% compared to the baseline. Memory reduction and computational speed up were the main thoughts on this set. See appendix A.3 for additional details.

Data set #3 shares again all of its properties with the baseline set except for the amplitude growth factor β . A value $\beta = 1.05$ displays a total growth in amplitude compared to the previous time-step by 5%. See table 3.2 for the exact values. The initial and the last snapshots of this data set are presented in figure 3.2.

Data set #4 is a data set consisting of 21 sets similar to the baseline case. The baselines' time delta is $\Delta t = 0.025s$. Here the first set has the baseline case time-delta and then for every new set the time delta is raised by $0.0005s$ resulting in 21 time-delta steps until the last set has a time delta $\Delta t = 0.035s$.

	α	β	κ	ω	\mathbf{F}	\mathbf{v}
	[m]	[-]	[1/m]	[1/s]	[Hz]	[m/s]
$w_{n,1}$	5	1.05	1	3.142	0.5	3.142
$w_{n,2}$	0.1	1.1	10	12.566	2	1.257
$w_{n,3}$	20	1	5	31.416	5	6.283
$w_{n,4}$	4	0.97	19	100.531	16	5.291
$w_{n,5}$	1	0.9	5	113.097	18	22.619

Table 3.2: Wave parameters of third data set of the artificial flow field. Frequency and velocity are derived from wave-number and angular frequency.

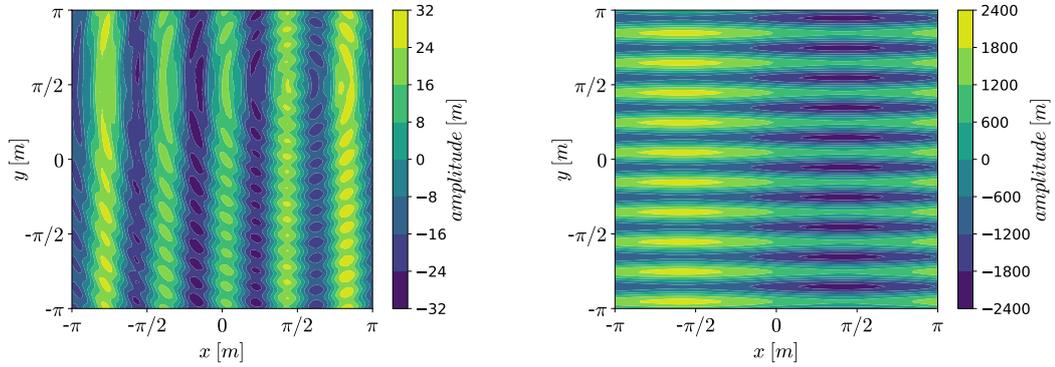


Figure 3.2: Artificial flow-field from data-set # 3. Left: Flow field at $t_0 = 0s$, right: Flow field at $t_{101} = 2.5s$

3.2 3D Cylinder in a cross flow

Data investigated from a three dimensional cylinder in a cross-flow (CCF) is based on an experiment from Ackerman et al. in which pressure distribution on a cylinder in a cross flow was investigated. In a previous thesis at the department by Schreiber the experiment was used to validate CFD data calculated with the scale adaptive simulation (SAS) turbulence model. The SAS-model can operate in RANS as well as LES modelling without a distinct grid dependency [Menter et al., 2012, p.1]. The model combines the advantage of resolving turbulent structures with LES while using RANS-modelling in wall proximity. For the CCF-case it showed good agreement of cylinder surface pressure distributions from the original experiment. Within the current investigation another unsteady data set was created. In addition to the SAS data set a URANS modeling was created on the same mesh. A monitor point on the cylinder surface, similar to a pressure probe in the original experiment revealed two dominant frequencies at 1300Hz and 2570Hz. Within the experiment of the Ackerman et al. [2009] these frequencies were

detected as fundamental vortex shedding frequencies on the cylinders surface.

The section of the measurement is a rectangular cuboid with, in stream-wise direction, $0.381m$ in width, $1.524m$ in height and $3.581m$ long, see figure (3.3). The cylinder is mounted in the rear third of the section and has a diameter of $37.26mm$. Based on this dimension and the freestream inlet velocity the Reynoldsnumber of the flow is 6.83×10^5 . Here the experiment with a free stream Mach number of 0.7 was reconstructed in the simulation. In this domain a hybrid mesh is used to perform URANS and SAS calculations with a layer of prism cells on the cylinder's surface. The remaining domain is filled with tetrahedral cells. The solver used to calculate the data is in-house flow solver Hydra. For both models, SAS and URANS 200 snapshots were saved. The time step in between snapshots is $\Delta t = 2e - 5$ seconds.

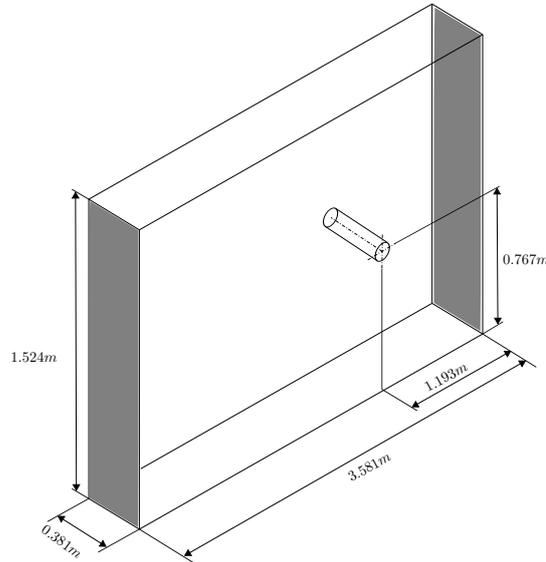


Figure 3.3: Sketch of the domain of the 3D cylinder in a cross flow. Grey shaded planes are inlet on the left and outlet on the right. All other planes are solid.

3.3 Turbine case

To apply the analysis on jet engine data a high pressure engine turbine was investigated. The data emerge from a URANS simulation with in-house code on a two stage turbine with, in stream wise direction, stator rotor stator rotor configuration. Because of the axial symmetry in azimuthal direction, a circular sector with azimuthal periodic boundary

conditions was set to be the domain. The setup includes cooling channels in the stators and in the first rotor as well as cavities in between rotating and static components. The mesh for the geometry is split into four pieces, in each stage one for the rotor and one for the stator including cooling ducts. The interfaces in between these four meshes are cuts in axial directions between the mesh blocks. Due to the move-able geometry of this set up it is not possible to analyse it completely since the move of geometries itself would act as a mode, spoiling the physical meaningful result. Another issue is the projection of the results onto the grid. If the grid moves, there is a new grid for each time step. Therefore it is not possible to project results onto one single grid. Therefore, the analysis was executed on the first rotor section only.

The first rotor section mesh contains 39 million cells. The in-house flow solver Hydra is node based, therefore DMD reads the corresponding 13 million nodes. In total 200 snapshots were saved for the calculation. The rotor spins with a angular frequency of ω_{rotor} . Its upstream stator cascade has a total number of n_{stator} stators. As a result the rotor passes the stator trailing edges with a frequency of

$$F_{SRI} = \frac{1}{2\pi} n_{stator} \omega_{rotor} \quad (3.1)$$

which is the stator-rotor interaction frequency.

RESULTS

4.1 Mode detection in the artificial flow field

In this section results from the artificial flow field are presented, whereat the focus lies on the proper detection of modes regarding spatial extent and oscillation frequency. Additionally, the detection of mode stability is analysed as well as the influence of changing the time step and the total number of used time steps to further comprehend DMDs characteristics. Besides, the introduced compressed DMD and compressed sensing DMD are investigated on the same basic data set to quantify their performance and to obtain a comparison to standard- and exact DMD.

Baseline data set

Results from the investigation of the baseline data set with standard DMD are shown in figure 4.1. Modes are displayed that best fit the predetermined five frequencies from the data set. Within their spatial extend, three modes at $0.5Hz$, $5Hz$ and $16Hz$ match the initial waves $w_{n,1}$, $w_{n,3}$ and $w_{n,4}$ in figure 3.1. Their spatial orientation including wavelengths, matches the given wave input in the data field. The $18Hz$ mode shows, compared to the initial wave $w_{n,5}$, only structures in y-direction. Due to its wave characteristics it has no velocity in x-direction. By definition of $w_{n,5}$ the time dependant oscillation is a sin-wave travelling in y-direction only. As a result, the correspondent mode detected shows spatially only oscillations in y-direction. The wave pattern of $w_{n,5}$ in x-direction, which is time independent, has no influence on the spatial character of the resulting mode. The artificial wave $w_{n,2}$ has a frequency of $2Hz$. Indeed the analysis

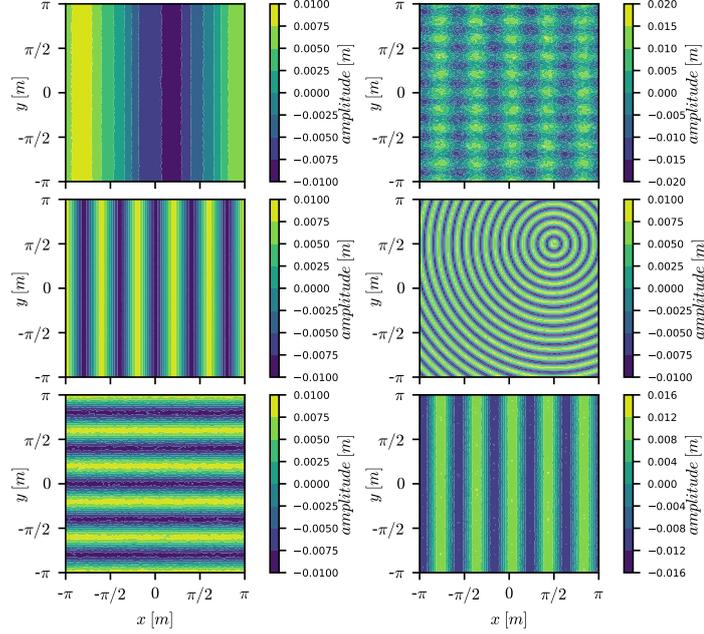


Figure 4.1: Real part of modes of data set # 1 detected with standard DMD algorithm. From left to right, top to bottom, mode with frequencies of: 0.5Hz, 2Hz, 5Hz, 16Hz, 18Hz and the average mode at 0Hz

shows that the mode is spectral detected but its spatial extend is not clearly referable to wave $w_{n,2}$. A closer look onto the mode shows that it's origin, wave $w_{n,2}$, is visible with wave pattern in y-direction. In addition, distinctive wave crests of the average mode are also visible. Data set # 1 contains $N - 1 = 100$ modes, where many physically unimportant modes show characteristics of the average mode. This is also the case for the 2Hz mode. A physical unimportant mode is shown in figure 4.2. Hereby, most non-physical important modes show artifacts of the average mode but owning various different frequencies.

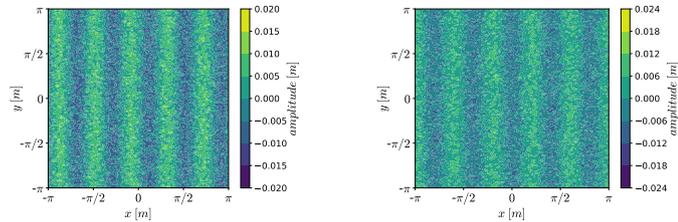


Figure 4.2: Physical unimportant mode of artificial flow field, detected by standard DMD. Left: real part, right: imaginary part. Frequency: 6.717Hz

Overall, standard DMD and exact DMD have detected the frequencies with less than 0.5% accuracy in data set #1. See table 4.1. There has been no significant differences in the two analysis of this data set by standard DMD and exact DMD. As pointed out in section 2.5.1 this applies when the last snapshot in the collection of snapshots is a linear combination of all previous snapshots. The results are satisfying in terms of identifying the modes and their frequency. Wave $w_{n,5}$ points out that only the time dependant oscillations play a role in the detection. Furthermore, the low-amplitude wave $w_{n,2}$ reveals difficulties in spatial reconstruction due to the presence of waves with amplitudes of several magnitudes larger.

Mode stability

In order to overview the stability behaviour of the modes it is practical to plot the Ritzvalues in the complex plane. In the plot a unit circle is added to indicate the region of stability. If an absolute Ritzvalue is greater than one, the mode is growing with time and the Ritzvalue lies outside of the unit circle. In contrary, decaying mode Ritzvalues lie inside the circle. Stable modes have an absolute Ritzvalue of one and lie on the circle. The position along the circle and therefore the angle of the Ritzvalue in the complex plane is proportional to the frequency of the mode. Zero frequency values lie therefore along the positive part of the real axis where the complex angle is zero. As a result, the stable average mode, oscillating at zero Hertz, lies on the point of intersection of positive real axis and unit circle. In general, the modes come in complex conjugated pairs which makes the real axis a symmetry axis in the plot. The further Ritzvalues lie on the unit circle away from the average mode, the larger their frequencies. Due to the symmetry, Ritzvalues with largest possible frequency lie on the negative real axis.

Analysis of data set #3 is used to assess the influence of a decaying and growing amplitude on the absolute Ritzvalues. The analysis shows that the stability of the modes is identified by the absolute Ritzvalues. In fact, in this case they meet with the predetermined β values in the data set, indicating the growth or decay of the mode amplitude with each time step. In figure 4.3 the corresponding Ritzvalues are shown. Noticeable is that the mode with smallest amplitude growth factor of 0.9 is not detected. As revealed in the baseline analysis, modes with small amplitude compared to other present modes with large amplitudes can have issues in the proper spatial recreation. Here the modes initial amplitude at $t = t_0$ is 1. Its final amplitude is $0.9^{100} = 2.65e - 05$. The immense decay over time might be the reason for DMD not to detect the mode properly. In contrast, the other four modes are detected and marked black in the plot. The according Ritzvalues are

found by comparing the detected modes frequencies with the predetermined frequencies. Here the frequencies for the detected modes coincide with a deviation of less than $\pm 0.02\%$. If the absolute Ritzvalue is larger than one, which applies for wave $w_{n,1}$ and $w_{n,2}$, they lay clearly outside the unit circle. Similar wave $w_{n,3}$ with an absolute Ritzvalue of one is right on the unit circle and $w_{n,4}$, with a decaying amplitude growth factor of 0.97, is located within the circle.

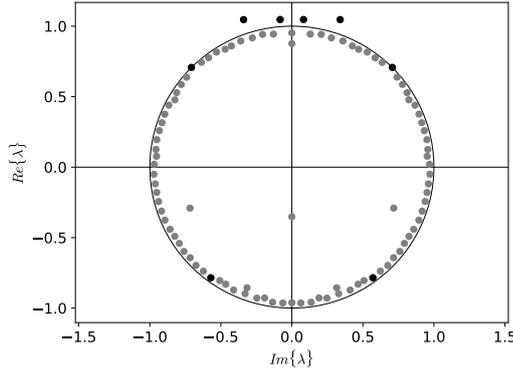


Figure 4.3: Ritzvalues acquired by exact DMD on data set #3. Black dots indicate detected modes' Ritzvalues.

Influence of time step and number of time steps

The artificial flow field data set #1 is tested to detect wave $w_{n,3}$ with a small amount of snapshots only. Since the data set itself consists of snapshots with fixed time steps, the total measurement period T is extended with each more time step taken into account. In fact, each additional time step changes the fundamental frequency defined in (2.7). With $N = 101$ snapshots and a time step of $\Delta t = 0.025s$, the data set's fundamental frequency is at $0.4Hz$. Hence this is the lowest frequency that can oscillate in one complete measurement period. Exact DMD is successively executed, each time with one more snapshot added. The frequency detection of the $5Hz$ mode is shown in figure 4.4. For each execution of DMD the frequency closest to $5Hz$ is divided by $5Hz$ and plotted on the ordinate to evaluate their over- or underestimation. On the abscissa the number of used snapshots for each DMD execution is plotted. With a small number of snapshots used, the modes correct frequency is over- and underestimated. Up until a number of 5 used snapshots this is visible. From 5 used snapshots upwards, the frequency is detected within a deviation of $\pm 1\%$. From 5 until 100 used snapshots this deviation oscillates around the value of one in a seemingly sinusoidal manner. The sinusoidal amplitude

grows until around 50 used snapshots and then decays again. At 100 used snapshots the oscillating deviations amplitude is zero. The reason for this sinusoidal behaviour has not been clarified so far. Taking a total of 5 snapshots into account results in an

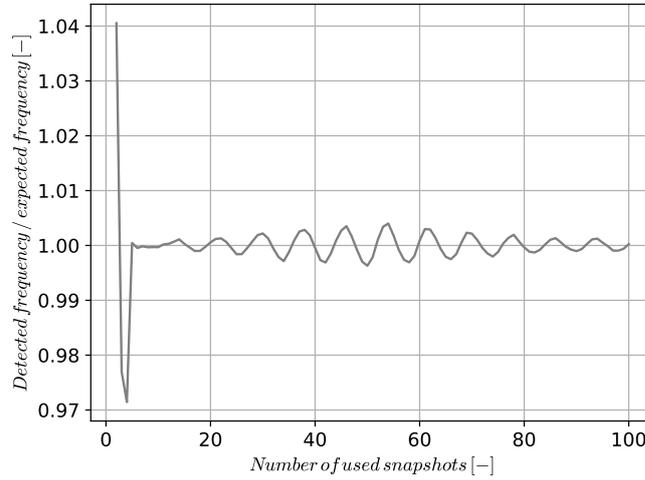


Figure 4.4: Deviation of detected 5Hz mode frequency with growing number of used snapshots with exact DMD.

overall time period of $0.1s$ and hence in an fundamental frequency of $10Hz$. The detected modes frequency is $5Hz$, showing that DMD captures frequencies below the fundamental frequency. In fact, DMD captures the frequency even though the oscillating spatial points have not performed one entire oscillation period.

Within the spectral detection of modes, an analysis of data set #4 shows how the Shannon-Nyquist sampling theorem applies. According to the sample theorem the time step necessary to resolve the $16Hz$ wave is $(2 \times 16Hz)^{-1} = 0.03125s$ and for the $18Hz$ it is $(2 \times 18Hz)^{-1} = 0.02778s$. Time steps in data set #4 vary in between $0.025s$ and $0.035s$. In figure 4.5 the deviation of detected frequency versus the used time step is shown. The deviation is the ratio of detected frequency over the expected, predetermined frequency of the modes. Within the successive increase of the time delta, the sample frequency for the $16Hz$ and the $18Hz$ fall below Nyquist-Shannon sample frequency. As soon as the time step is exceeds the critical value, the detected frequencies are underestimated. All other frequencies are detected precisely since their critical time steps are above 0.035 seconds.

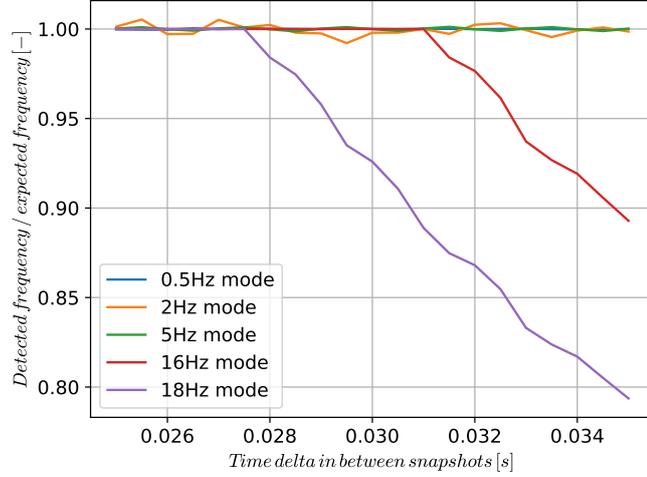


Figure 4.5: Deviation of detected mode frequencies with growing time step.

Compressed DMD

The baseline data set #1 is also investigated by compressed DMD to test its application with the promising outlook to decrease computation time. Since the analysis of upcoming engine component data can be quite large, the speed up and the handling of fewer, compressed data seems promising. As pointed out in section 2.5.2, compressed DMD is performed using the three different types of compression matrices. Before executing compressed DMD the amount of compression needs to be chosen. In these cases it is set to 5% of the original data, resulting in 750 instead of 15000 spatial points M used. Results from Compressed DMD showed that all 5 frequencies and spatial modes are detected using only 5% of the spatial data. Deviations compared to the expected frequencies lie within $\pm 0.145\%$ except for the detection of the low amplitude wave $w_{n,2}$ detected by compressed DMD using random compression matrix. Its deviation is 2.782% from the predetermined 2Hz as shown in table 4.1. Within the spatial extend of the modes of this data set there is no significant difference to the modes detected by exact DMD.

In addition to the comparison of varying compression methods, compressed DMD with single pixel compression matrix is executed with different percentage of compression. Derived from the original data, these analysis hold only 1% and 10% of the original data. Consequently the data size is reduced from 15000 down to 1500 and 150 spatial points. Spectral, the ten percent compression, similar to the previous mentioned five percent compression, detects all frequencies within 0.752% maximum deviation. Contrary the

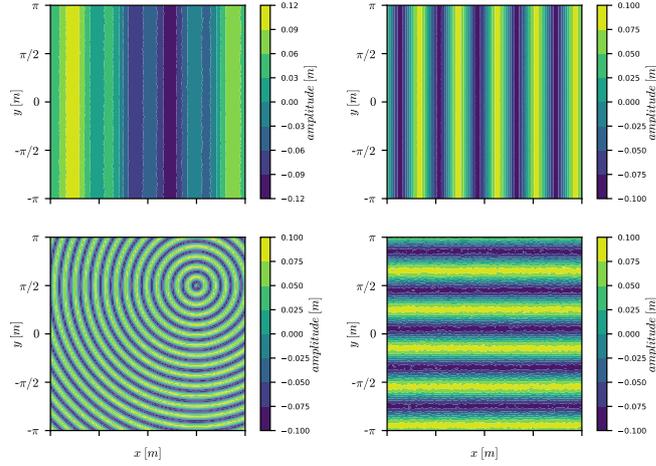


Figure 4.6: Detected modes of compressed DMD with 1% single pixel compression. Clockwise, real part of 0.5Hz, 5Hz, 18Hz and 16Hz mode.

one percentage compression is detecting all frequencies except the low-amplitude wave at $2Hz$. Its closest detected differs by 15.7% from the predetermined $2Hz$, see table 4.1. Spatially, both methods proof that detected modes are mapped without significant differences to standard- and exact DMD. The four modes detected by one percent compression are displayed in figure 4.6. Spatially the compression had notifiable effect on the modes.

Compressed sensing DMD

Within the usage of compression to evaluate benefits for handling large CFD data cases, compressed sensing DMD is executed on data set #2. Compression is set to 15% of the original spatial data and a single pixel compression is chosen. For the recreation of the modes, a cosin transform basis is used and the number of sparse entries in the sparse approximation is set to be 2% of the original spatial input. As a result, the sparse approximation vector contains only 128 entries instead of the full state 6400 entries. The outcome of modes is displayed in figure 4.7. Comparing the spatial detection of the modes to the baseline, detected by standard DMD in figure 4.1, compressed sensing modes are spatially not as defined as their baseline pendant. This applies for example to the spherical $16Hz$ mode displayed in the central plot on the right hand side in figure 4.7. The main properties of the spherical wave are retraceable, though the mode is spatially not perfectly reconstructed. However all sinusoidal waves are all clearly detectable. Again the low amplitude wave at $2Hz$ shows in its detected mode artifacts of the dominant $5Hz$

	0.5Hz	2Hz	5Hz	16Hz	18Hz
Frequency	[Hz]	[Hz]	[Hz]	[Hz]	[Hz]
standard DMD	0.499988	2.005807	5.001209	16.000045	17.999757
exact DMD	0.499988	2.005807	5.001209	16.000045	17.999757
comp. DMD (random) 5%	0.499256	2.055648	5.000566	15.999706	17.999846
comp. DMD (ran. gauss) 5%	0.500255	2.00142	5.00015	16.00088	17.999159
comp. DMD (single pixel) 10%	0.500173	2.015044	5.001266	16.000044	18.001134
comp. DMD (single pixel) 5%	0.499324	1.987492	5.001009	16.000096	18.001216
comp. DMD (single pixel) 1%	0.498605	2.313885	4.989504	15.999949	17.995816
Deviation	[%]	[%]	[%]	[%]	[%]
standard DMD	-0.002	0.290	0.024	0.000	-0.001
exact DMD	-0.002	0.290	0.024	0.000	-0.001
comp. DMD (random) 5%	-0.149	2.782	0.011	-0.002	-0.001
comp. DMD (ran. gauss) 5%	0.051	0.071	0.003	0.005	-0.005
comp. DMD (single pixel) 10%	0.035	0.752	0.025	0.000	0.006
comp. DMD (single pixel) 5%	-0.135	-0.625	0.020	0.001	0.007
comp. DMD (single pixel) 1%	-0.279	15.694	-0.210	0.000	-0.023

Table 4.1: Frequencies and corresponding deviation of detected frequencies of standard DMD, exact DMD and compressed DMD. Compressed DMD compression manner in brackets, percentage is use of actual points in space from entire data.

wave which is, similar to the baseline case, detectable in most other physical unimportant modes. Spectral properties are similar to the results from compressed DMD. The compressed sensing algorithm uses the same method as the compressed DMD algorithm to acquire the reduced order eigenpairs. While reduced order eigenvalues (Ritzvalues) are the same for full state and reduced state, the derived frequencies of the modes are the same for compressed sensing DMD and compressed DMD. The main difference is the reconstruction of the full state eigenvectors, therefore the reconstructed modes can differ from each other.

Compressed sensing is performed on a reduced sized data set with 6400 spatial points instead of 15000 from the baseline data set. While the exact DMD algorithm with the larger data set runs through in a matter of minutes, the compressed sensing algorithm runs through in a matter of days. In fact, the CoSaMP algorithm uses the majority of time for the sparse approximation of the modes. This is why the use of compressed sensing DMD is only recommended when full state data vectors are not at hand.

4.2 Analysis of 3D cylinder in cross flow

The investigation of the three dimensional cylinder in a cross flow through DMD is used to apply the above described varieties of the algorithm onto real field data. Results from

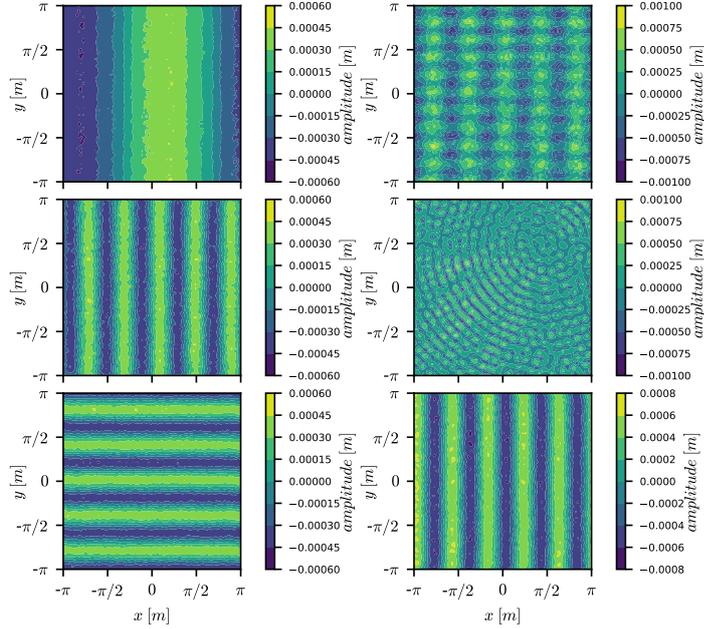


Figure 4.7: Real part of modes of data set # 1 detected with compressed sensing DMD algorithm. From left to right, top to bottom, mode with frequencies of: 0.5Hz, 2Hz, 5Hz, 16Hz, 18Hz and the average mode at 0Hz

the experiment are used to analyse the oscillations of the flow field due to the vortex shedding in the cylinder wake and to compare spectral properties from the experiment with the results of DMD. Within the application of the algorithms, advantages and flaws of the DMD methods are investigated when applying DMD on such large data sets from unsteady flow CFD computations.

SAS data set

The analysis of the SAS data set is performed with the exact DMD algorithm. The 200 snapshots from the data set reveal two dominant modes at $1270Hz$ and $2530Hz$. These modes are detectable in the analysis of pressure, density, x-velocity and y-velocity. Within all four flow quantities the dominant frequencies are in close proximity of those of the vortex shedding frequency detected by Schreiber in the CFD data and Ackerman et al. in the experimental data.

In figure 4.8 the dominant mode is displayed. The mode is oscillating at a frequency of $1270Hz$ and has strong appearance in the cylinder wake. Its peak oscillation is located on the downstream facing cylinder surface. In upstream direction, these strong oscillating

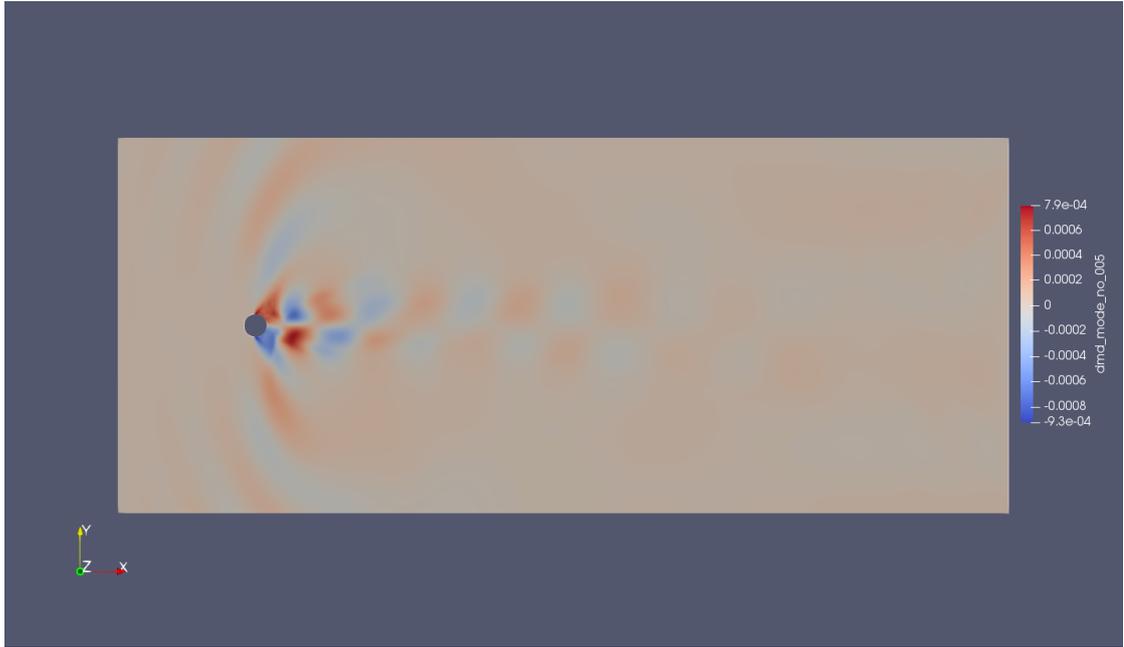


Figure 4.8: Slice of real part of dominant mode of SAS data with 200 snapshots. Analysed quantity: pressure. Frequency: 1270Hz.

locations end with a distinctive, sharp line. This is the shock wave of the cylinder flow occurring due to the transsonic flow. Furthermore, the peak oscillations downstream the cylinder continue within the cylinder wake. With further distance downstream the oscillations becoming less distinctive. This is due to vortex shedding that resolves with sufficient great distance downstream, resulting in decaying pressure fluctuations. Another conspicuity is the influence of the oscillating pressure field in the freestream, visible along the y-axis above and underneath the cylinder. Consequently the pressure fluctuations occurring in the wake due to vortex shedding, influence the pressure field not only in the wake but also in the freestream.

The second dominant mode at 2530Hz is the first higher harmonic, peaking at 1.992 times the first dominant frequency. Compared to its basic mode, this higher harmonic mode is symmetric to the x-z plane in which the cylinder axis lies. Locations of oscillations are similar to the one from the basis 1270Hz mode but with greater wavelength of the higher harmonic mode. Again the pressure field in close proximity above and underneath the cylinder shows oscillations. With greater distance downstream of the cylinder the oscillations diverge from the x-z plane and occur close to the border of the turbulent wake and the non-turbulent flow.

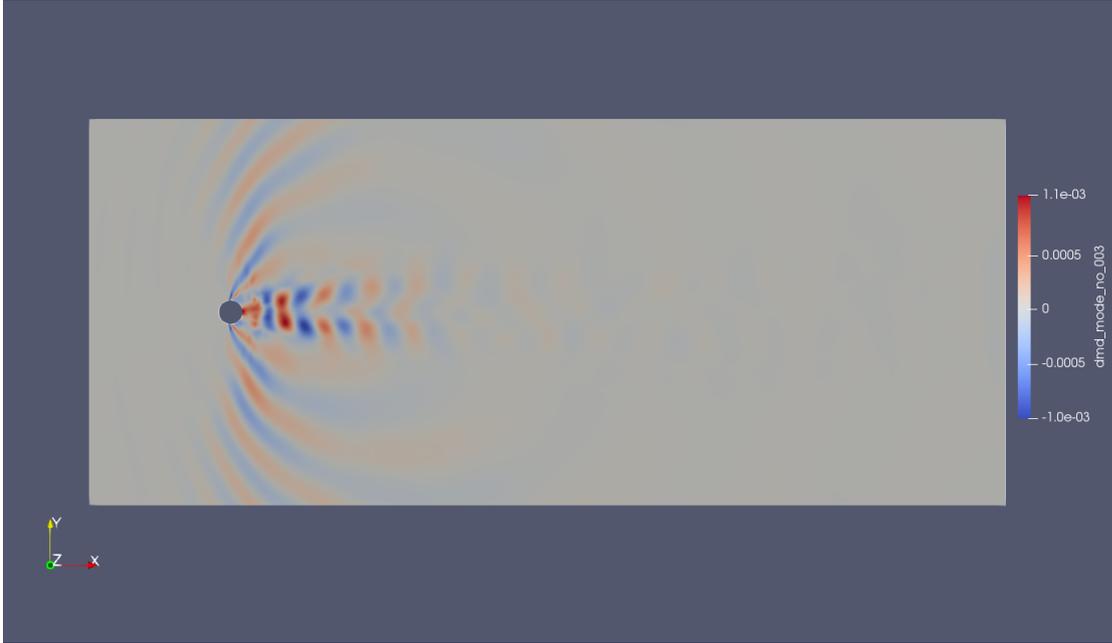


Figure 4.9: Slice of real part of dominant modes of SAS data with 200 snapshots. Analysed quantity: pressure. Frequency: 2530Hz.

A harmonic frequency is a frequency that is a positive integer multiple of a significant baseline frequency. Since the complex angle in the Ritzvalue plot is proportional to the frequency, harmonic frequencies have also a positive integer multiple of the baseline frequency. A series of harmonic frequencies is thus visible in the Ritzvalue plot when their Ritzvalues have constant complex angles in between each other, starting from the average mode Ritzvalue. The composition of the $N - 1 = 199$ modes show that DMD is able to capture up to the 5th harmonic frequency. Other modes that are not harmonic frequencies of the first dominant mode show mainly high frequency content. Spatially this high frequency content is often located in high turbulent areas such as the cylinder boundary layer or in the turbulent separation bubble right downstream of the cylinder. This is possibly due to the algorithm trying to resolve highly non-linear dynamical flows.

In the Ritzvalue plot in figure 4.10 the eigenvalues are distributed along the unit circle. The average mode is located on the unit circle at $0Hz$, crossing the real axis at positive real value of one. Following the circle in either direction, the first two large yellow dots, assigned with a large energy criteria value, are the two dominant modes described above. Note that each eigenvalue, except the average mode, appears as complex conjugated pair

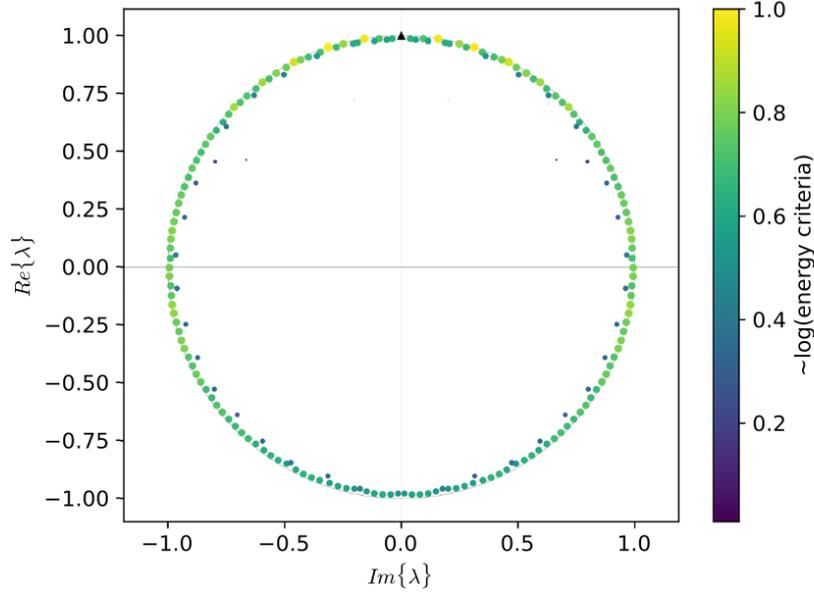


Figure 4.10: Ritzvalues of exact DMD from 200 snapshots of SAS data set. Analysed quantity: pressure. Black triangle indicates average mode.

and therefore each eigenvalue is mirrored by the real axis. Starting at the average mode, the further a Ritzvalue is located away along the circle, the larger the modes frequency. In fact, the angle of the Ritzvalue in the complex plane is, as stated earlier, an indication of how large its frequency is. The complex angle of these values is a positive integer multiple of the baseline Ritzvalue angle and therefore a series of at least two harmonic frequencies.

The data set shows that high frequencies own smaller values in the energy criteria. The data sample frequency is $50kHz$. The analysis show that DMD resolves frequencies until $25kHz$, half of the sample frequency which confirms Shannon-Nyquist theorem.

The analysis of different flow variables does not show any significant differences in mode detection except that no dominant mode is detected by analysing the velocity component in z-direction. Regarding the highly two dimensional set up of the cylinder in figure 3.3, oscillations in the direction of the cylinder axis are not expected. Also no complete stable mode is detected by velocity in z-direction. Maximum absolute values of Ritzvalues, indicating the stability of a mode, are 0.994 in the z-direction whereas maximum values of pressure, density and the remaining velocity components are 1, see table 4.2. A slight

mode order		1	2	3	4	5	6	7
energy criteria [-]	pressure	1.004	0.799	0.799	0.743	0.743	0.736	0.736
	density	1.003	0.814	0.814	0.754	0.754	0.620	0.620
	x-velocity	1.002	0.747	0.747	0.670	0.670	0.446	0.446
	y-velocity	0.978	0.908	0.908	0.545	0.545	0.527	0.527
	z-velocity	0.398	0.398	0.395	0.395	0.352	0.352	0.337
frequency [Hz]	pressure	0	-2533	2533	1265	-1265	3814	-3814
	density	0	-2530	2530	1265	-1265	-3812	3812
	x-velocity	0	1273	-1273	-2539	2539	-5115	5115
	y-velocity	0	1266	-1266	785	-785	2528	-2528
	z-velocity	771	-771	3654	-3654	3352	-3352	6152
Ritzvalue [-]	pressure	1.000	0.999	0.999	0.998	0.998	0.998	0.998
	density	1.000	0.999	0.999	0.999	0.999	0.997	0.997
	x-velocity	1.000	0.998	0.998	0.998	0.998	0.995	0.995
	y-velocity	1.000	1.000	1.000	0.996	0.996	0.996	0.996
	z-velocity	0.994	0.994	0.994	0.994	0.993	0.993	0.993

Table 4.2: Energy criteria, frequency and absolute Ritzvalue of first seven modes detected by exact DMD. Modes are ordered according to highest energy criteria value.

difference in the different flow quantities is the order of the modes. Analysing density and pressure DMD ranked the dominant mode at $2530Hz$ at first place and the $1270Hz$ mode second, showing that selecting the modes can be non trivial.

A reduction of the SAS data set is performed to see whether this large scale dominant frequency modes are sufficiently detectable with less snapshots. A reduced data set of 50 snapshots is able to detect the $1230Hz$ mode at $1296Hz$, whereas the second dominant frequency is not clearly detected. A rise of the number of used snapshots to 100 detected both dominant modes and ranked them, according to their energy criteria, to the top. This is visible in their Ritzvalue plot, see appendix A.4, where the dominant modes Ritzvalues are clearly specifiable while using 50 and 100 snapshots of the data set only.

URANS data set

In addition to the SAS data set the URANS data set with 200 snapshots is investigated. Two dominant frequencies are detected similar to the SAS data set but with slightly higher frequencies at $1340Hz$ and $2640Hz$. A main difference in the detection manner is, compared to the SAS data, that the average modes are not ranked at first place according to the energy criteria. Spatially the mode at $1340Hz$ as displayed in figure 4.11 is as distinctive as its pendant from the SAS data set. A similarity is the spatial peak

in oscillation in close proximity of the downstream facing cylinder surface. Furthermore, it is noticeable that the fluctuation in the cylinder wake is spatially decaying more rapidly with increasing distance downstream.

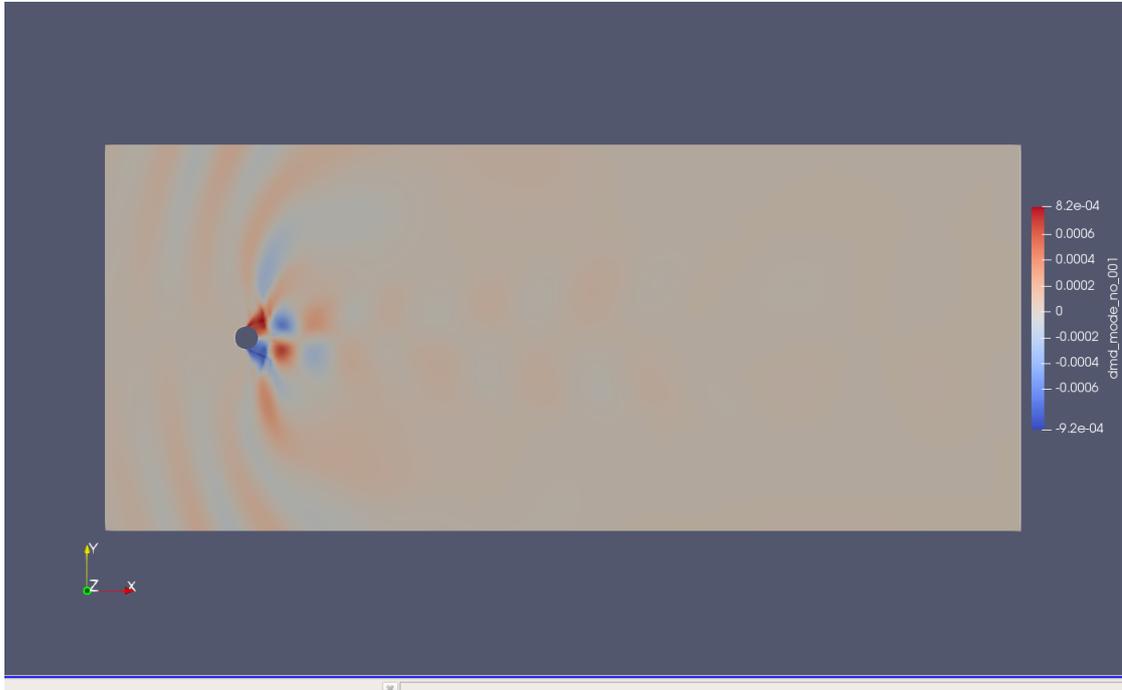


Figure 4.11: Slice of real part of dominant mode at 1340Hz of URANS data with 200 snapshots. Analysed quantity: pressure.

The less defined features of the oscillations is applicable for the harmonic frequency at 2640Hz . As displayed in figure 4.12, fluctuations in the cylinder wake are spatially similar to the harmonic mode detected in the SAS data set but less distinctive. In fact, their extend downstream along the wake decays faster with increasing distance from the cylinder. In addition, the diverging manner of the oscillations in respect of the x - z plane is not identifiable.

A closer look onto the analysis shows that high ranked modes also obtain higher absolute Ritzvalues at around 1.004. In the Ritzvalue plot, the analysis of URANS shows within each flow quantity the appearance of here referred to as zero Hertz, non-average modes. These modes have zero frequency but small Ritzvalues and their spatial appearance is not the average of all snapshots. See figure 4.13. The results from 200 snapshots are 199 modes in total of which 1 is the average mode, 98 are complex conjugated pairs and 2 are non-average zero frequency mode. The origin of these non-average zero Hertz modes

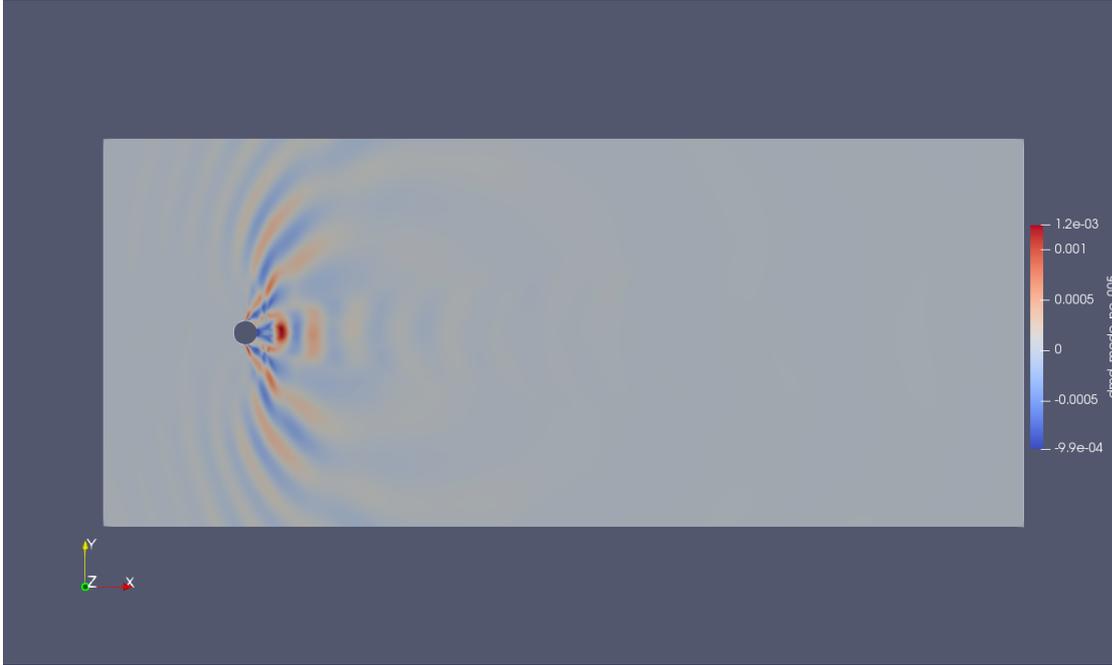


Figure 4.12: Slice of real part of dominant mode at 2640Hz of URANS data with 200 snapshots. Analysed quantity: pressure.

has not been clarified. It is unknown why the expectation of receiving one average mode and else only complex conjugated pairs is not met. If the number of used snapshots is even, the number of modes is uneven and with one single average mode all other modes can possibly appear in complex conjugated pairs. Consequently the use of an uneven number of snapshots results in an even number of modes and therefore the presence of the non-complex-conjugated average mode results in one extra mode that has no complex conjugated pair.

Compressed DMD

Since the SAS data set has 13 million spatial points an analysis with the compressed sensing DMD is not carried out due to the immense time consumption of the CoSaMP algorithm. Another bottleneck when applying this algorithm is the compression matrix \underline{C} . This compression matrix is of size $H \times M$ where M is the number of spatial points and H the number of compressed spatial points. Even if H is only 1% of M , the size of the compression matrix exceeds the size of any other matrix computed by DMD by far as long as $M \gg N$. For medium sized data sets, as the one from the 3D cylinder in a cross flow, this causes serious problems with random access memory RAM storage. If

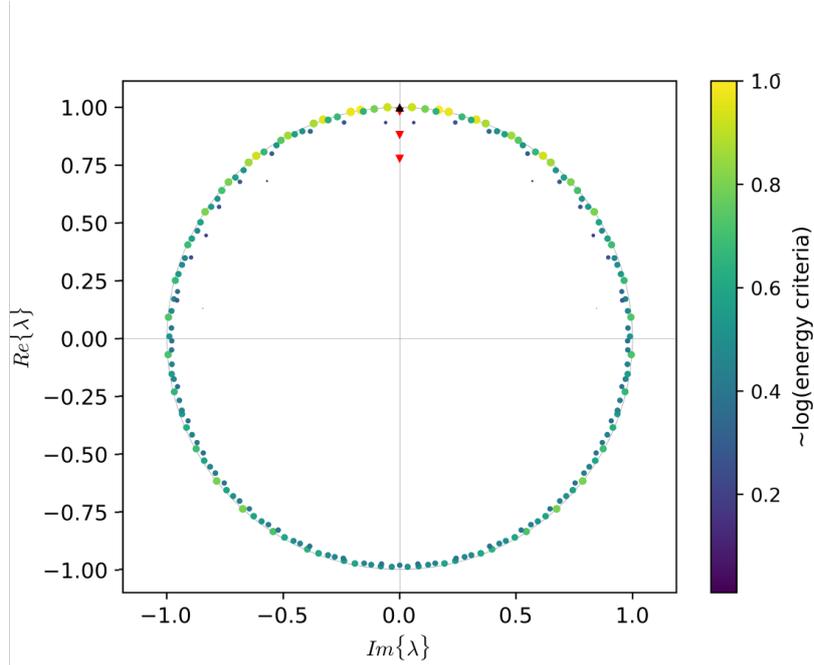


Figure 4.13: Ritzvalues of exact DMD from 200 snapshots of URANS data set. Analysed quantity: x -velocity. Black triangle indicates average mode, red triangles indicate non average zero frequency modes.

the compression matrix is constructed as random projection or as Gaussian random projection matrix, this matrix is dense. Therefore the two methods have not been executed within the analysis of the SAS data set. However the single pixel measurement compression matrix is sparse. Therefore the compressed DMD algorithm is adapted, reading the same single pixel from each snapshot when the snapshot data is read by the algorithm. With this avoidance of creating the compression matrix, memory size is no further issue and compressed DMD is executable. The analysis with single pixel measurement is executed with 3% and 1% of the original data. Both compressed analysis detected the two dominant frequencies similar to the non-compressed analysis. Compared to the exact DMD modes, the output from compressed DMD shows modes spatially less defined. The analysis of pressure for example shows the oscillations of the pressure field, which are not in the wake of the cylinder, even stronger. See figure 4.14. With higher compression, the intensity of the oscillations outside the wake become more dominant. In the modes of exact DMD this oscillation of the pressure field outside the wake are also visible but less intense compared to the results from compressed DMD.

From a memory usage point of view compressed DMD with single pixel compression has

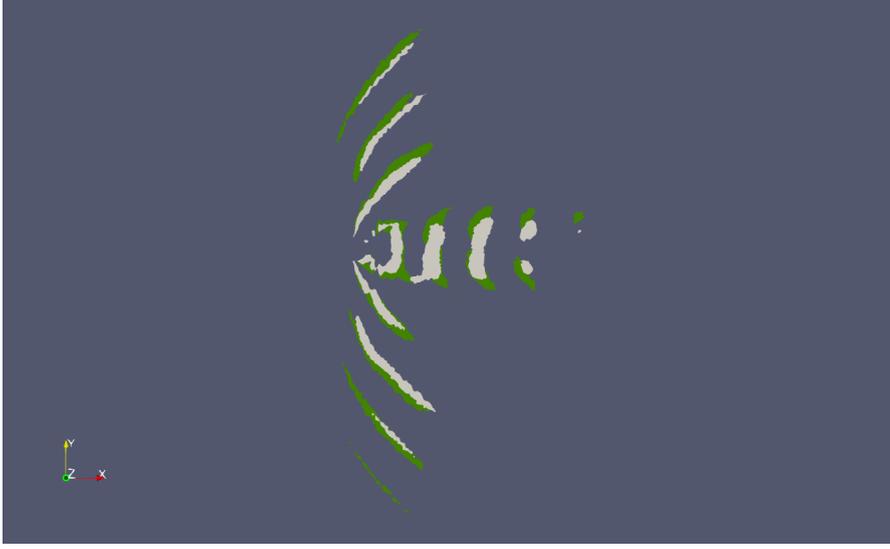


Figure 4.14: Second dominant modes at 2530Hz detected by compressed DMD. 2D cut of 1% compression (green) overlain with 3% compression (grey) at same data threshold.

not as many benefits regarding memory size. The data vectors need to be read completely since they are used for the full state mode reconstruction in (2.45). Nevertheless, compression is beneficial for the time consumption of the singularly value decomposition.

Data size

One of the general bottlenecks of the applied analysis with dynamic mode decomposition regarding the case size is the availability of RAM. Within the study of the 3D cylinder in a cross flow case, memory usage is investigated. In the procedure of calculating the DMD modes, largest matrices are of size $M \times (N - 1)$. In the case of the 3D cylinder there are $M = 13270339$ spatial points and $N - 1 = 199$ modes that are computed, resulting in a matrix size of 2.64 billion matrix entries. In the computation these matrices have a size of 21 gigabyte. The matrix of resulting modes has the same number of entries but is of complex size, saving two numbers per entry, therefore overall mode matrix size is 42 gigabyte. At the point of exact mode creation (2.39) both matrices, the modes $\underline{\Phi}$ and the second snapshot matrix \underline{X}_2 need to be saved in the memory. This results in an urgency of 63 gigabyte RAM. At this point other matrices e.g. $\underline{\Sigma}$ and V are stored in the memory as well but since their matrix dimensions do not include the spatial size M , their RAM usage is almost negligible. Holding enough memory to read the data and compute the modes is indispensable. Another important conspicuousness when working with the DMD is the time the algorithm needs to run through. For the 3D cylinder in

cross flow case it is clear, that reading the data and writing the modes onto the provided mesh is more time consuming than the actual DMD calculation itself.

4.3 Analysis of turbine rotor

In a third step of the identification of coherent structures through DMD the rotor of a two stage high pressure engine turbine is investigated.

Within this analysis all five flow variables, namely pressure, density and all three components of velocity are analysed by exact DMD. Regarding the spectral limit, it is notable that the maximum detected frequency is only 1.2% less than maximum resolvable frequency derived from Shannon-Nyquist criteria. In this case the algorithm as such is not exceeding the upper frequency detection limit for the given time step. Overall, a closer look onto the Ritzvalue plot revealed a more ordered distribution of eigenvalues compared to the cylinder in a cross flow. Here the eigenvalues are divisible into three groups. At first there are zero frequency modes, similar to the one detected in the 3D CCF URANS data set. This includes the average mode and non-average zero frequency modes. Next, there are the stable modes that are located on the unit circle with high energy criteria and finally there are decaying modes that are lying inside the unit circle. It seems that the latter one accompany the stable modes since each stable mode's eigenvalue has a decaying mode eigenvalue in close proximity. In the Ritzvalue plot in figure 4.15 it is also observable, that at low frequencies, these decaying eigenvalues almost lie exactly on each stable eigenvalue pendant. Another characterization in this plot is the equidistant distribution of the stable modes eigenvalues on the unit circle in azimuthal direction. Since the angle in the complex plane indicates the modes frequency, this repetitive distribution on the circle shows that all stable modes are integral multiplies from the lowest stable non-zero frequency. In table 4.3 the deviation from the lowest first three non-zero frequencies compared to the their harmonic pendant from the stator-rotor interaction frequency F_{SRI} are listed. It is notable that the lowest frequencies are all in close proximity of the stator-rotor interaction frequency.

As indicated by the colour and size of the dots in the Ritzvalue plot in figure 4.15, the low frequencies pointed out in table 4.3 do not own the largest energy criteria value. According to the criteria, modes with much higher frequency than the one listed in table 4.3 are ranked first. Also the average mode is in none of the analysis of the five quantities ranked first. Many of these high frequency, high energy criteria value show within the domain only oscillations at the edge of the interface plane between the stator mesh

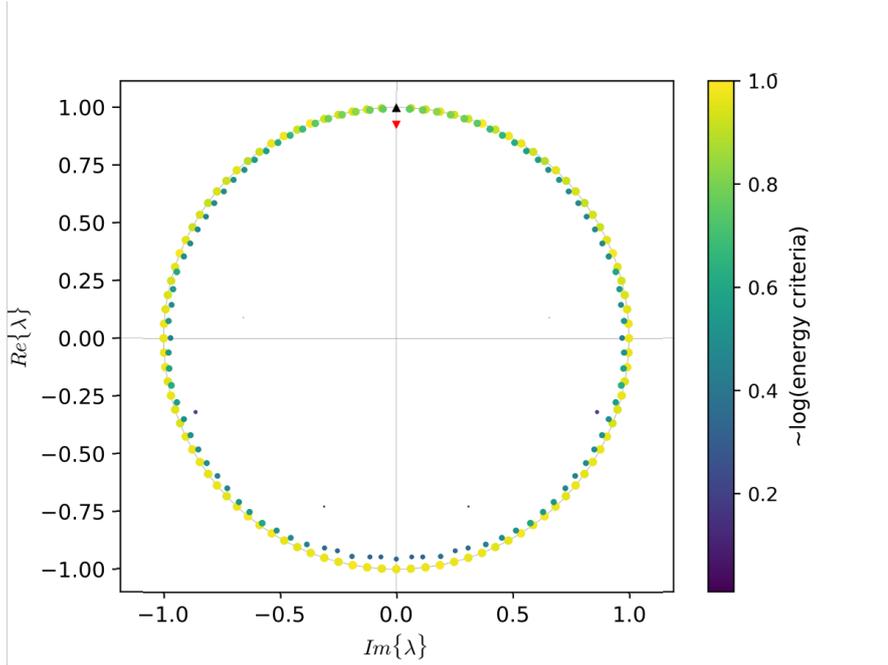


Figure 4.15: Ritzvalues of exact DMD from 200 snapshots of turbine rotor data set. Analysed quantity: pressure. Black triangle indicates average mode, red triangle indicates non-average zero frequency mode.

block and the rotor mesh block. These modes have high energy criteria value and are spatially very concentrated at the pointed out location. This is applicable in the interface plane upstream as well as downstream of the rotor. Still, these modes have frequencies that are a multiple integer of the stator-rotor interaction frequency. In addition to the investigation of the 200 snapshot data set, only the first 100 snapshots of this data set are analysed by exact DMD. The analyse quantity is pressure. Here the mode with highest energy criteria is the average mode at $0Hz$. Furthermore, the mode with a frequency closest to the stator-rotor interaction frequency F_{SRI} is ranked in fourth place regarding the energy criteria value. The detected stator-rotor interaction frequency deviates 6.2% of the original frequency determined in (3.1). Its second harmonic is at 1.985 times the detected baseline modes frequency and is ranked fifth in the energy criteria.

Figure 4.16 shows a schematic contour plot slice through the mode with closest frequency to the stator-rotor interaction frequency. Upstream the rotor blades, close to the interface plane large areas where the pressure field oscillates are clearly visible. This emerges from the flow field wake of the preceding stator blade which the rotor passes through repeatedly with the stator-rotor interaction frequency. It is also noticeable that these oscillations

Deviation from F_{SRI} harmonic	pressure		density		x-velocity	
	stable	unstable	stable	unstable	stable	unstable
#1	6.7	-3.0	5.8	4.3	5.8	-7.8
#2	7.0	-3.5	7.4	-2.6	6.1	-9.7
#3	6.7	-0.9	7.1	-0.2	6.3	-0.8

Table 4.3: Percentaged deviation of smallest three non-zero mode frequency of stable and unstable branch from corresponding F_{SRI} harmonic frequency. Abstract of the analyse of pressure, density and x-velocity.

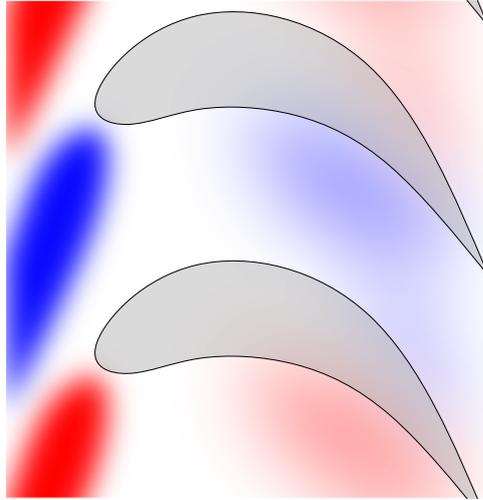


Figure 4.16: Scheme of azimuthal section through mode with frequency closest to F_{SRI} . Colours indicate mode magnitude.

reoccur on pressure and suction side of the stator. Focusing onto the blades contour it is observable that the oscillations in the flow field nearby the blade emerge from the cooling ducts in the blade. Similar observations are made at the mode ranked second in the energy criteria. This mode has three significant areas of fluctuations. The first one is in the domain upstream of the rotor blades, with its location similar to the stator-rotor interaction. This can be noted as harmonic oscillations from the stator-rotor interaction since the frequency is 4.96 times the frequency from the mode displayed in figure 4.16. The second location is at the trailing edge and in the wake of the stator blades. At last, the oscillation at this specified frequency occur as a fluctuation along the stator pressure side, induced by the outflow of the cooling duct.

A result of the investigation is the detection of the stator-rotor interaction frequency with a deviation of +6.2%. The spectral detection concludes all five flow quantities as

well as the analysis of pressure with 100 snapshots only. Modes of analysed pressure data clearly show the oscillations of the pressure field in the wake of the stator blades. The harmonic oscillation of the flow field is due to the rotor passing through wakes from its preceding turbine stators. This however is not considered a coherent structure since the oscillation occurs from forced mechanism that originate through a harmonic change of boundary conditions and not through large scale structures evoking in turbulent shear layers. Nevertheless DMD identifies this oscillation including higher harmonics. In their spatial extend the modes show how the stator-rotor interaction frequency impacts flow around the rotor blades. The stator-rotor interaction frequency imposes oscillations on trailing edges as well as on outflow of cooling ducts. Other modes show so far not clarified oscillations on interface edges between rotor and stator mesh. The equidistant distribution of Ritzvalues along the unit circle indicate that there are only stable modes with harmonic frequencies of F_{SRI} . A comparison with this flow field calculated with SAS turbulence model would be interesting to see whether this strong harmonic composition of the modes is retraceable or not.

CONCLUSION

The overall objective of the study at hand is to detect coherent structures in in-field CFD data of jet engine components. This has been tackled by investigating the implementation framework of four variations of DMD on a predetermined data field which successfully demonstrated the modal detection and its characterization. Furthermore a generic flow case has effectively shown the identification of coherent structures in analogy to an experiment. In a final step, the examination of jet engine flow field data was carried out, capturing dominant flow dynamics.

The artificial flow field proves explicitly the decomposition of data into a model in which the spatial modes including their oscillating frequency as well their stability behaviour are captured. The four outlined varieties of DMD demonstrate their abilities in detection where compressed data and sub sampled data lead to approvable results comparable to the original methods of standard and exact DMD. Possible inaccuracy in spatial detection of low amplitude oscillations is revealed causing errors in correct mode representation or even failing to detect the corresponding frequency. Spectral limitations exhibit the confirmation of Nyquist-Shannon sampling theorem as well as mode frequencies detection below fundamental frequency. Compressed sensing DMD displayed its ability to reconstruct full size modes from sub sampled data while suffering from immense increase in run time. Consequently, this method can only be recommended when full state data is not at hand. From the remaining pool of DMD alternatives, standard DMD results are analog to the ones from exact DMD. Thus exact and compressed DMD are found to be better suited to investigate real flow solutions.

The application of exact and compressed DMD on real flow field data met the expect-

tations of revealing important modes in the cylinder wake. In fact, resulting modes distinguish two dominant frequencies. Their composition is a fundamental mode and their first harmonic. Both detected modes' frequency is in close proximity of the vortex shedding frequency from the experiment the simulation set up is based on. Due to the solely analysis of primitive flow variables only, physical important modes show an influence on the potential flow-field which is not assignable to the turbulent coherent structure. Artifacts of such are particular striking when applying the compressed DMD algorithm. In addition, its application bears memory issues that can be overcome by using single pixel compression only. Moreover, the level of detail in the used CFD turbulence model is visible in the spatial resolution of the modes. The SAS data where less of the turbulent spectrum is resolved by RANS, result in enhanced distinction of the modes spatially. The proper ranking of the modes turns out to be an important component in order to detect physically important flow features especially when dominant frequencies of the investigated system are unknown beforehand. Here the energy criteria ranking system suggested by Tissot et al. [2014] exhibits good results particularly on the SAS data set, ranking average modes and the dominant modes in the wake at the top.

Run time consumption is dominated by reading data and writing modes onto the mesh. Hence the beneficial feature of compressed DMD, using less spatial points computing the SVD exposed to be almost negligible. Consequently exact DMD is the most adequate method combining time consumption, memory usage and accuracy. In the turbine case DMD analysis failed to detect coherent structures definitely. Though dominant modes show good approximation with expected spatial and spectral stator-rotor interaction characteristics, the spectra reveals no other non related frequency.

In general, the investigation of flow field data through dynamic mode decomposition approves its capabilities to determine essential flow field features. Further research on the here presented engine component could focus on evaluating each rotor and stator component individually to receive a more precise insight into the flow dynamics and their origin. Hence mode oscillations in between rotating and fixed mesh interfaces can be further investigated to determine their physical relevance. Moreover, the analysis of the flow field can be extended by examining fields of vorticity in order to detect coherent structures more detailed avoiding the acquisition of oscillations in the non turbulent flow field or such of impressed fluctuations due to movable geometry. The within the study implemented tool to analyse CFD data will help engineers to gain an insight in the dynamics of engine component flow in order to detect root causes of oscillations and consequently improve overall engine performance.

BIBLIOGRAPHY

- J. R. Ackerman, J. P. Gostelow, A. Rona, and W. E. Carscallen. Measurements of fluctuating pressures on a circular cylinder in subsonic crossflow. *AIAA Journal*, 47(9):2121–2131, 2009.
- W. E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics*, 9:17–29, 4 1951.
- L. Bernal, R. Breidenthal, G. Brown, J. Konrad, and A. Roshko. On the development of three dimensional small scales in turbulent mixing layers. *stsf*, pages 8–1, 1979.
- W. B. Boyce and R. C. DiParma. *Elementary Differential Equations and Boundary Value Problems*. John Wiley Sons Inc., 10 edition, 2012.
- S. L. Brunton, J. L. Proctor, J. H. Tu, and J. N. Kutz. Compressed sensing and dynamic mode decomposition. *Journal of computational dynamics*, 2(2):165, 2015.
- N. B. Erichson, L. Mathelin, J. N. Kutz, and S. L. Brunton. Randomized dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 18(4):1867–1891, 2019.
- R. Feynman, R. Leighton, and M. Sands. *The Feynman Lectures on Physics*, volume 1. California Institute of Technology, new millennium edition, 1962.
- P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge, 2 edition, 2012.
- C. Karpfinger. *Höhere Mathematik in Rezepten (German) / Higher mathematics in recipes*. Springer-Verlag Heidelberg, 2014.
- J. Kutz, S. Brunton, B. Brunton, and J. Proctor. *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*. SIAM, 2016a.

- J. N. Kutz, X. Fu, and S. L. Brunton. Multiresolution dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 15(2):713–735, 2016b.
- F. Menter, M. Kuntz, and R. Bender. A scale-adaptive simulation model for turbulent flow predictions. 2012.
- D. Needell and J. A. Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and computational harmonic analysis*, 26(3):301–321, 2009.
- J. L. Proctor, S. L. Brunton, and J. N. Kutz. Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, 15(1):142–161, 2016.
- C. Rowley, I. Mezic, S. Bagheri, P. Schlatter, and D. Henningson. Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics*, 641:115 – 127, 12 2009.
- A. Ruhe. Rational krylov sequence methods for eigenvalue computation. *Linear Algebra and its Applications*, 58:391–405, 04 1984.
- P. J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.
- K. Schreiber. Validierung des sas-modells für die simulation von strömungen durch schubumkehrer / validation of sas-models for validating flow simulation through thrust reverse unit. Bachelors thesis at Technical University of Berlin, 2019.
- J. Stoer and R. Bulirsch. *Numerische Mathematik 2*. Springer-Verlag Heidelberg, 5 edition, 2005.
- K. Taira, S. L. Brunton, S. T. 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):4013–4041, 2017.
- G. Tissot, L. Cordier, N. Benard, and B. R. Noack. Model reduction using Dynamic Mode Decomposition. *Comptes Rendus - Mecanique*, 342(6-7):410–416, 2014.
- J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz. On dynamic mode decomposition: Theory and applications. *Journal of Computational Dynamics*, 1(2):391–421, 2014.

APPENDIX

A.1 Notes on the Pseudo-Inverse

Assume we have a linear equation system with a non-square matrix $\underline{\underline{A}} \in \mathbb{R}^{n \times m}$, $N \neq M$ the unknown variable $\underline{x} \in \mathbb{R}^{M \times 1}$ and the known, right hand side $\underline{b} \in \mathbb{R}^{N \times 1}$:

$$\underline{\underline{A}} \underline{x} = \underline{b}.$$

Since we can not compute the inverse of $\underline{\underline{A}}$ we can use the following workaround. Create the square matrix $\underline{\underline{A}}^T \underline{\underline{A}}$

$$\underbrace{\underline{\underline{A}}^T \underline{\underline{A}}}_{m \times m} \underline{x} = \underline{\underline{A}}^T \underline{b}$$

and if non-singular, meaning that a regular inverse exists, we can invert it. Therefore

$$\underline{x} = \underbrace{(\underline{\underline{A}}^T \underline{\underline{A}})^{-1} \underline{\underline{A}}^T}_{\text{pseudo-inverse}} \underline{b}$$

$$\underline{x} = \underline{\underline{A}}^\dagger \underline{b}$$

Depending on the determination of the system the equation has two different meanings:

1. If $\mathbf{N} < \mathbf{M}$ the system is over determined. There are more equations than variables. In general there are infinite solutions. In this case the pseudo-inverse solves for the \underline{x} with minimum L_2 norm

2. On the other hand side if $\mathbf{N} > \mathbf{M}$ the system is under-determined. In general there are zero solutions. The pseudo inverse solves for the least square solution by minimizing $\min \| \underline{b} - \underline{A}^\dagger \underline{x} \|^2$.

See also [Rowley et al., 2009, p.3].

A.2 The Arnoldi algorithm as basic idea to approximate eigenvalues

The Arnoldi algorithm¹ is an algorithm to compute a set of leading eigenvalues of a matrix [Arnoldi, 1951, p.1]. The subset of eigenvalues of a matrix \underline{A} is received by transforming it into a smaller matrix, then computing it's eigenvalues which approximate the leading eigenvalues of the initial. This matrix transformation is achieved by projecting \underline{A} onto a basis of its Krylov subspace [Ruhe, 1984, p.391]. There are several different bases that can be found. Three of these will be explained in the following section, the first one to give a general idea of the method, the second one which is the Arnoldi method and the last one which is used by DMD.

For a linear equation system such as $\underline{x}_{j+1} = \underline{A} \underline{x}_j$ where $j \in \mathbb{N}$ and $\underline{A} \in \mathbb{R}^{M \times M}$ the Krylov subspace of \underline{A} , denoted by \mathcal{K} is defined by

$$\mathcal{K}_j(\underline{A}, \underline{x}_1) = \{ \underline{x}_1, \underline{A} \underline{x}_1, \underline{A}^2 \underline{x}_1, \dots, \underline{A}^{j-1} \underline{x}_1 \}$$

with the initial given vector \underline{x}_1 and the system operator \underline{A} as function inputs. From the linear system equation we can derive that each column vector of the subspace represents one state vector at a discrete time

$$\mathcal{K}_j = \{ \underline{x}_1, \underline{x}_2, \dots, \underline{x}_j \}.$$

A matrix \underline{X}_j shall be defined that spans the subspace above, for simplicity referred to as *Krylov matrix*

$$\underline{X}_j = [\underline{x}_1, \underline{x}_2, \dots, \underline{x}_j] \in \mathcal{K}_j. \tag{A.1}$$

Now the general method to find the leading eigenvalues of \underline{A} is constructed by assuming that with a large enough sequence j the vectors in the Krylov matrix become linearly dependent, meaning that any future state vector is representable by a linear combination

¹Introduced by American engineer Walter Edwin Arnoldi in 1951.

of all previous vectors [Ruhe, 1984, p.392]

$$\begin{aligned}
 \underline{x}_{j+1} &= \underline{A} \underline{x}_j & (A.2) \\
 &= \underline{x}_1 c_1 + \underline{x}_2 c_2 + \dots + \underline{x}_j c_j \\
 &= \underline{X}_j \underline{c}
 \end{aligned}$$

where c_1, \dots, c_j are the coefficients, retrievable in the coefficient vector \underline{c} . Note that the size of the Krylov column subspace j is by definition less than or equal to the size of column subspace M of the system operator \underline{A} . Further we can multiply the system matrix by the entire Krylov matrix instead of just a single vector (compare (A.2))

$$\begin{aligned}
 \underline{A} \underline{X}_j &= [\underline{A} \underline{x}_1, \dots, \underline{A} \underline{x}_j,] \\
 &= [\underline{x}_2, \underline{x}_3, \dots, \underline{X}_j \underline{c}] \\
 \underline{A} \underline{X}_j &= \underline{X}_j \underline{F}. & (A.3)
 \end{aligned}$$

This results in a sparse matrix \underline{F} on the right hand side, containing only ones on the subdiagonal and the coefficients c_1, \dots, c_j in its last column. The matrix \underline{F} is of so called *companion type*² and it's eigenvalues are a subset of the eigenvalues of \underline{A} [Ruhe, 1984, p.3]. Note that $j \leq M$ and therefore the resulting number of eigenvalues are a subset of the full M eigenvalues.

The linear dependence with increased j is not an abrupt happening but rather a successive approach. Hence we can not stick to the assumption of the entirely linear dependence of the vectors and need to modify equation (A.3) to

$$\underline{A} \underline{X}_j = \underline{X}_j \underline{F} + \underline{R} \tag{A.4}$$

$$\underline{A} \underline{X}_j \approx \underline{X}_j \underline{F} \tag{A.5}$$

with some residuum \underline{R} . Hence the eigenvalues of \underline{F} do not contain the exact but an approximation of the subset of the eigenvalues of \underline{A} . This residuum is kept small by keeping its direction orthogonal to the Krylov subspace. This minimizes the residuum in (A.4) $\underline{R} = \|\underline{A} \underline{X}_j - \underline{X}_j \underline{F}\| = \min$. Rearranging (A.5) we receive

$$\underline{F} = \underline{X}_j^\dagger \underline{A} \underline{X}_j. \tag{A.6}$$

²Matrix of companion type is a sparse square matrix. It contains only entries in it's last column and ones in the subdiagonal. All other values are zero.

where \dagger denotes the pseudo inverse of $\underline{\underline{X}}_j$ that needs to be applied since the non square matrix has no regular inverse. More details on the pseudo inverse will be covered in the next section 2.4.1. The equation points out that the projection of $\underline{\underline{A}}$ onto its Krylov subspace \mathcal{K}_j via the basis $\underline{\underline{X}}_j$ results in the matrix $\underline{\underline{F}}$. The final step is to compute it's eigenvalues. Due to it's construction, this matrix of companion type is ill conditioned for implementations and therefore not used in practice [Schmid, 2010, p.9].

The essence of this method is that the approximation of the eigenvalues is only depending on the chosen subspace of \mathcal{K}_j and the direction of the residual [Ruhe, 1984, p.5]. Therefore other basis than $\underline{\underline{X}}_j$ of the Krylov subspace can be found. This matter of fact is used by the Arnoldi method where a different base is chosen. The algorithm computes via *Gram-Schmidt orthogonalization method* an orthonormal basis of \mathcal{K}_j . This basis is spanned by the column subspace of the matrix $\underline{\underline{Q}}_j$ that projects the system operator $\underline{\underline{A}}$ onto the Krylov subspace. The projection results in an upper Hessenberg matrix³

$$\underline{\underline{H}} = \underline{\underline{Q}}_j^T \underline{\underline{A}} \underline{\underline{Q}}_j \tag{A.7}$$

that again approximates leading eigenvalues of $\underline{\underline{A}}$ [Schmid, 2010, p.8] and is well conditioned to be solved for eigenpairs with traditional eigenvalue algorithms [Stoer and Bulirsch, 2005, p.2]. For these reasons this particular method is highly popular in numerical linear algebra. Reviewing (A.7) and (A.6) it is notable that when projecting $\underline{\underline{A}}$ onto it's Krylov subspace, the resulting matrix approximates leading eigenvalues of $\underline{\underline{A}}$ itself while obtaining the size of just $j \times j$ instead of $M \times M$.

In practical applications, the Arnoldi method is a robust workhorse to approximate leading eigenvalues - presupposed the system matrix $\underline{\underline{A}}$ is known [Schmid, 2010, p.8]. When analysing data with DMD, without necessarily knowing the underlying equations, $\underline{\underline{A}}$ is unknown. Hence, another method is deduced that finds an alternative basis of the Krylov subspace \mathcal{K}_j . This is achieved by the SVD of the Krylov matrix $\underline{\underline{X}}_j$

$$\underline{\underline{X}}_j = \underline{\underline{U}} \underline{\underline{\Sigma}} \underline{\underline{V}}^T \tag{A.8}$$

resulting in a matrix $\underline{\underline{\Sigma}}$ containing all singular values and two orthogonal matrices $\underline{\underline{U}}$ and $\underline{\underline{V}}$. The first $r \in \mathbb{N} \setminus \{0\}$ columns of matrix $\underline{\underline{U}}$ form a orthonormal base of $\underline{\underline{X}}_j$, consequently spanning the Krylov subspace \mathcal{K}_j . The value of r is equal to the rank of the Krylov matrix [Karpfinger, 2014, p.394] which is equal to j , the number of columns

³Upper Hessenberg matrix is a upper tridiagonal matrix with additional entries on its subdiagonal.

in \underline{X}_j ⁴. In the same manner as in the methods described previously, the received base \underline{U}_j is used to project the initial matrix onto its Krylov subspace

$$\underline{U}_j^T \underline{A} \underline{U}_j = \tilde{\underline{A}} \quad (\text{A.9})$$

resulting in an dense matrix $\tilde{\underline{A}}$ that again approximates the leading eigenvalues of the initial matrix \underline{A} .

Method enumeration	1	2	3
Initial matrix	\underline{A}		
Projection basis	\underline{X}_j	\underline{Q}_j	\underline{U}_j
Derived from	\mathcal{K}_j itself	Orthonormalization of \mathcal{K}_j	eigenvectors of $(\underline{X}_j, \underline{X}_j^T)$ via SVD
Link to subspace	identical	orthonormal basis	orthonormal basis
Projection result of type	\underline{F} Companion	\underline{H} Hessenberg	\underline{A} Regular, dense

Table A.1: Overview of the three given examples of creating a reduced order matrix that approximates leading eigenvalues. Method 1: Based on Krylov subspace with linear coefficients. Method 2: The Arnoldi algorithm. Method 3: Used by DMD.

An overview of the presented methods is given in table A.1. Each method provides a way to transform the initial matrix into a projection that approximates leading eigenvalues. An important note is that in all methods, the resulting projected matrix is of size $j \times j$ instead of $M \times M$. From numerical point of view this is desirable to reduce the computational effort to actually calculate leading eigenvalues. The computational benefit grows with the increasing ratio of $\frac{M}{j}$ which is the case in the applications of DMD.

A.3 Artificial flow field

Additional information on data set #2

Properties:

- Number of snapshots: 101
- Time delta: 0.025s
- Number of points in x -direction: 80

⁴The Krylov Matrix \underline{X}_j is of size $\mathbb{R}^{M \times j}$. For the practical case of applying DMD we can assume that $M \gg j$ and therefore $\text{rank}(\underline{X}_j) = j$. More details are provided below in section 2.4.1.

- Number of points in y -direction: 80
- Domain range in x -direction: $[-\pi, \pi]$
- Domain range in y -direction: $[-\pi, \pi]$

	α	β	κ	ω	\mathbf{F}	\mathbf{v}
	[m]	[-]	[1/m]	[1/s]	[Hz]	[m/s]
$w_{n,1}$	5	1	1	3.142	0.5	3.142
$w_{n,2}$	0.1	1	10	12.566	2	1.257
$w_{n,3}$	20	1	5	31.416	5	6.283
$w_{n,4}$	4	1	19	100.531	16	5.291
$w_{n,5}$	1	1	5	113.097	18	22.619

Table A.2: Wave parameters of second data set of the artificial flow field. Frequency and velocity are derived from wave-number and angular frequency.

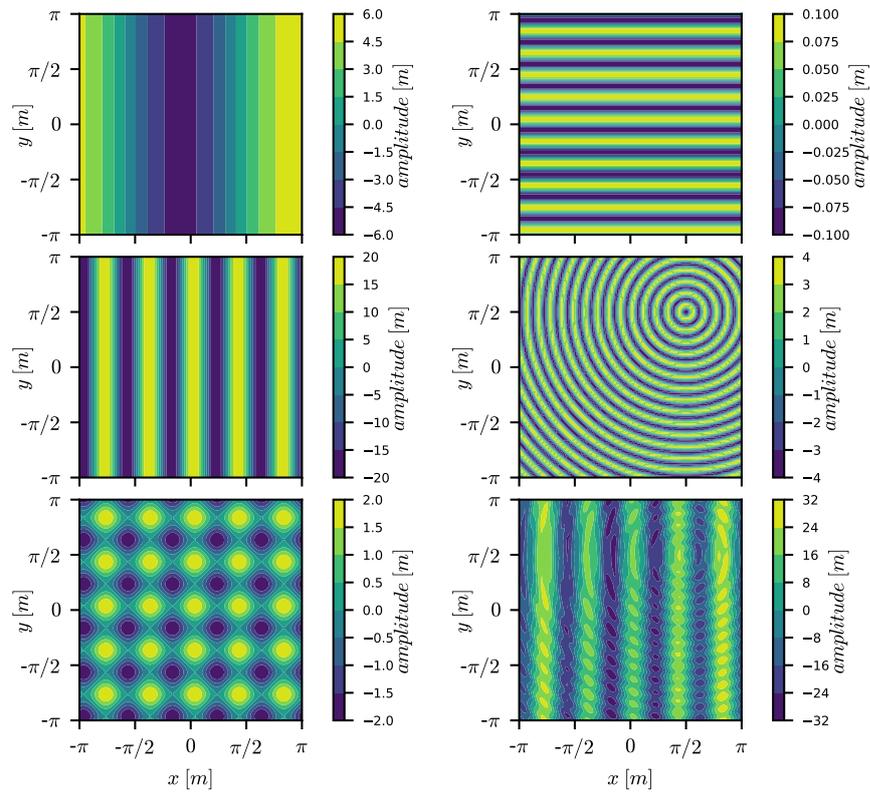


Figure A.1: Overview of data set #2. Top to bottom, left to right: Wave $w_{0,1}$, wave $w_{0,2}$, wave $w_{0,3}$, wave $w_{0,4}$, wave $w_{0,5}$ and the superposition of all including noise at $t = 0s$.

A.4 3D cylinder in cross flow with less snapshots

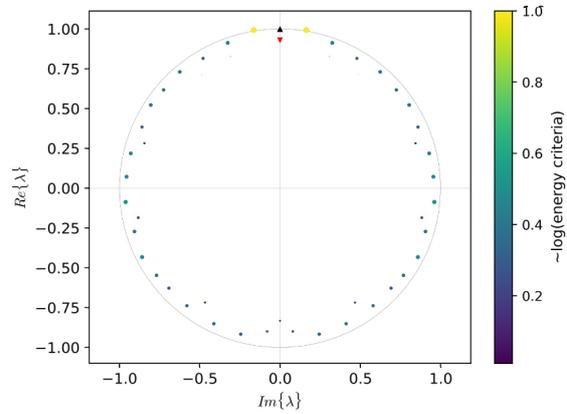


Figure A.2: Ritzvalues of exact DMD from 50 snapshots of SAS data set. Analysed quantity: pressure. Black triangle indicates average mode, red triangle indicates non average, zero frequency mode.

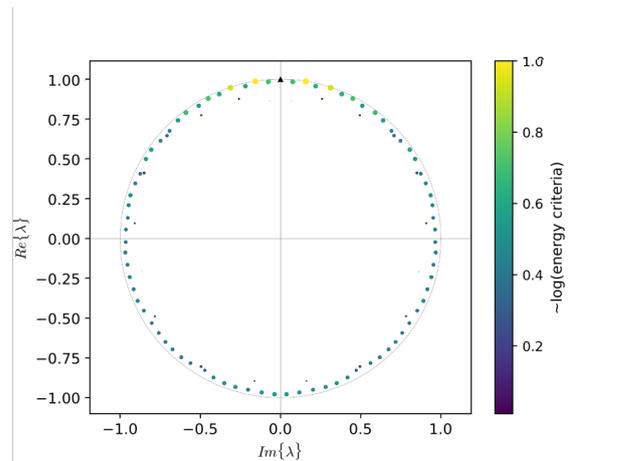


Figure A.3: Ritzvalues of exact DMD from 100 snapshots of SAS data set. Analysed quantity: pressure. Black triangle indicates average mode.