`

A REPORT

ON

MODAL DECOMPOSITION OF FLUID FLOWS AND ITS APPLICATION TO COMPRESSOR ACOUSTIC CHARACTERISTICS

BY

Venugopal Ranganathan 2017A4PS0495G

AT

Mercedes-Benz Research and Development, India Pvt. Ltd.

A Practice School-II Station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

(June, 2021)

`

A REPORT

ON

MODAL DECOMPOSITION OF FLUID FLOWS AND ITS APPLICATION TO COMPRESSOR ACOUSTIC CHARACTERISTICS

BY

Venugopal Ranganathan 2017A4PS0495G

Prepared in partial fulfillment of the Practice School-II Course BITS F421

AT

Mercedes-Benz Research and Development, India Pvt. Ltd.

A Practice School-II Station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

(June, 2021)

`

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI
(RAJASTHAN)
Practice School Division

Station: Mercedes-Benz Research and Development, India Pvt. Ltd.
Center: Bangalore

Duration: Feb 1, 2021 – Jun 30, 2021 **Date of Start:** Feb 1, 2021

Date of Submission: June 30, 2021

Title of the Project: Modal decomposition of Fluid Flows and its application to compressor acoustic characteristics

ID No./ Names/ Discipline of the student: 2017A4PS0495G, Venugopal Ranganathan, B.E. Mechanical

Name(s) of the expert(s): Suresh Kumar Kannan, Anand Sadalge

Name(s) of the PS Faculty: Shashank Mohan Tiwari

Key words: Computational Fluid Mechanics, Modal Decomposition, Proper Orthogonal Decomposition, Dynamic Mode Decomposition, Single Value Decomposition.

Project Areas: Fluid mechanics, Data processing, Computational Fluid Dynamics, Modal decomposition

`

ABSTRACT:

Increase in the computational powers of present day computers has made it possible to model and simulate increasingly complex flows. Such simulations produce large amounts of data (of the order of TBs) which makes it difficult - not only to analyze the data and draw useful inferences from but also increases the burden on storage systems.

A nearest neighbor algorithm called *k*-d tree algorithm was implemented to achieve data reduction. Then, two modal decomposition techniques – Proper Orthogonal Decomposition (POD) and Dynamic Mode Decomposition (DMD) were applied on this reduced data to extract energetically or dynamically important features in the form of coherent structures. The results of the *k*—d tree algorithm and the two modal decomposition techniques are presented for two case studies – flow past a bluff body and air flow in a compressor.

The bluff body investigated was a cylinder. The geometry, mesh and other simulation parameters of this case study were taken from an aeroacoustics tutorial of a commercial CFD software, results of which such as pressure diagrams, FFT analysis were already available. So, this study was used to test the validity of the modal analysis outcome by testing both POD and DMD in their ability to capture the peak in pressure value at the vortex shedding frequency.

The second case study was airflow in a compressor. CFD simulation of airflow in an automobile compressor was performed and both POD and DMD were applied to the pressure data output from this simulation to obtain spatial-temporal patterns of pressure. These patterns assist in the understanding of compressor noise sources. Due to confidentiality reasons with regards to compressor geometry and CFD simulation parameters, the results of this study are NOT included in this report.

`

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to Mercedes-Benz Research & Development, India Pvt. Ltd. and the Practice School Division, BITS Pilani for providing me with the opportunity to do my PS-II at MBRDI.

I would like to express my gratitude to my supervisor, Suresh Kumar Kannan for providing me with the opportunity to work on this project. I would also like to thank him for his guidance and the knowledge he has shared with me during the course of this project.

I would also like to thank my manager, Mr. Anand Sadalge for his constant support and encouragement.

`

TABLE OF CONTENTS

`

LIST OF TABLES

`

LIST OF FIGURES

`

`

`

`

CHAPTER 1: INTRODUCTION

1.1 Turbochargers and compressor acoustics:

Noise in the car cabin, especially in the case of passenger vehicles is required to be kept to a minimum. This has become even more important in recent times due to the inclusion of features such as infotainment and online communication through the internet. This requires an extensive study on the range of frequencies of the noises produced and speeds (RPM) of engine operation at which these noises are most prominent. Much of the noise in the inlet section of automotive turbochargers have their source in the compressor. Therefore, a strong understanding of the noise sources in compressors is essential to suppressing noise produced by automotive turbochargers. POD and DMD assist in this pursuit by extracting meaningful information from the data output from CFD simulations which can be used to take measures to suppress the noise. A description of an automobile turbocharger and the types of noises produced is presented in the following text.

Turbocharger is a component seen in vehicle engines and is used to increase the engine's power output. It achieves this by "forced-induction" – using the kinetic energy of the exhaust gases to spin the compressor wheel, resulting in the compressor drawing in air through the intake manifold. The turbocharger

`

increases the volumetric efficiency of the engine by forcing in more air, effectively increasing the density of the intake air in the compressor.

Structurally, turbochargers generally consist of a compressor, turbine and a housing assembly. Its operation is described as follows: The exhaust gases turn the turbine wheel which is connected to the compressor wheel through a common shaft. The spinning turbine wheel turns the compressor wheel which draws in the intake air.

Other components seen in a turbocharger are:

1. Intercooler: used to cool down the intake air to increase the density and in doing so increase the volumetric efficiency and to reduce the occurrence of engine knocking.
2. Wastegate: Used to control the extent of turbocharging by regulating the amount of exhaust gas directed towards the turbine wheel.
3. Blow-off valve: This valve is used to vent the excess intake air. This is done to prevent excess pressure build-up in the compressor and backflow to the turbine which could cause damage.
4. Acoustics of turbochargers:

`

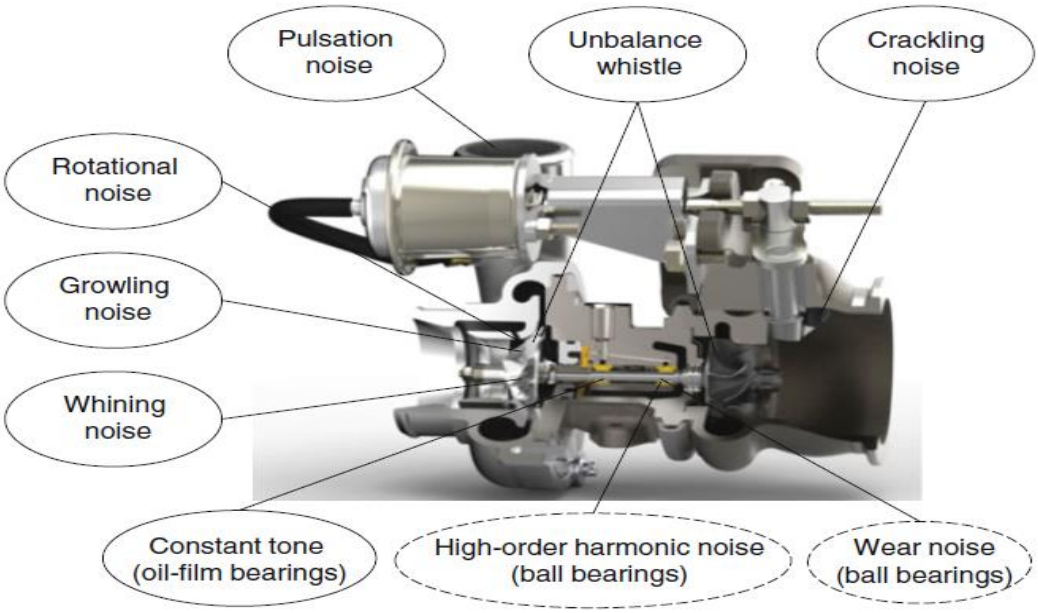


Figure 1.1: Induced noises of automotive turbochargers. Pulsation whistle, Rotational noise, Growling noise, Whining noise are classified under aerodynamic noise. Crackling noise, Unbalance whistle, Constant tone (howling), High-order harmonic noise and Wear noise are classified under rotordynamic noise. Taken from [1].

Different kinds of noises seen in automotive turbochargers are classified into two categories- aerodynamic noise and rotordynamic noise.

Aerodynamic noise includes - Pulsation whistle, Rotational noise, Growling noise and Whining noise.

Rotordynamic noise includes – Crackling noise, unbalance whistle, constant tone (howling), high-order harmonic noise and wear noise. These noise types – their sources, frequency and engine operation speed (RPM) at which they are observed are described in detail in [1].

`

1.2 Motivation for modal decomposition:

Aeroacoustics which is made of the words – “aerodynamics“ and “acoustics” is defined as the study of noise generation which is produced by turbulence in fluid motion or by the interaction of aerodynamic forces with surfaces. Due to the generation of noises from fluid motion, the field of aeroacoustics is closely linked to fluid mechanics. The study of fluid mechanics is carried out using Computational fluid dynamics (CFD) which is a tool used for simulating and analyzing fluid flows. It makes use of numerical schemes to solve partial differential equations based on the conservation of mass, momentum and energy. While increase in computational capacity has allowed simulations to be performed in a matter of a few minutes, modeling increasingly complex flow phenomenon such as turbulence poses a challenge in that the simulations carried out produce a lot of data (of the order of ~TBs).

The large data produced poses two problems. Firstly, storing of large data poses a problem in terms of memory usage of computer systems. Secondly, it makes the analysis of flow difficult – i.e., it is tough to sift through the large amount of data to make useful inferences about the nature of flow being investigated. Modal decomposition – defined as the extraction of energetically or dynamically important flow features is used as a tool to tackle both of these problems. It is

`

used to comb through the large data and create lower dimensional models that describe the dynamics of the flow and to do so as quickly and efficiently (in terms of memory usage and information captured by the low dimensional model) as possible.

`

CHAPTER 2: METHODOLOGY

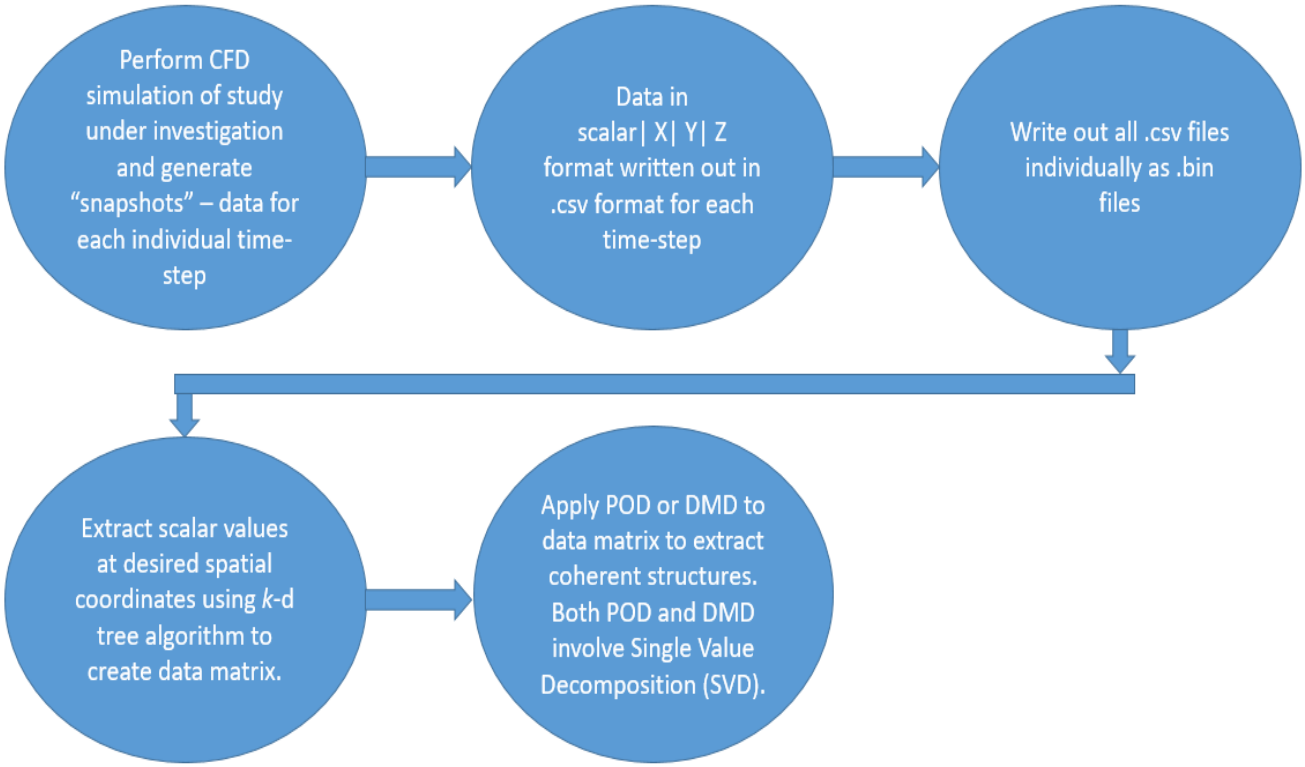


Figure 2.1: Working scheme adopted for the creation of the pressure data matrix on which POD and DMD are applied, starting from the set of .csv format files called "snapshots". The same working scheme is adopted for both the case studies.

2.1 Working scheme:

The working scheme adopted is described in the following steps:

1. CFD simulation is carried out for a particular case study. Post CFD simulation, data is output in the form of .csv files. Each of the .csv files contains data in the format

   Variable | Coordinates.

`

The “Variable” being the variable, information about whose distribution is to be extracted using POD & DMD. The “Coordinates” are the spatial coordinates comprising the gridspace. The variable is written out at each of the grid-points of the gridspace. Each of the .csv files is written out for each particular time-step of the simulation. These .csv files corresponding to successive time-steps are called “snapshots”. So, each snapshot contains data corresponding to an individual time-step arranged in columns. Column 1 contains the values of the “Variable” at the grid-points specified by the adjacent column (s) which comprise the “Coordinates”. In both the case studies – Flow past a bluff body and Compressor flow, the variable written out is static pressure (in Pascal for Flow past a bluff body and in mBar for Compressor flow) resulting in the format of data in each .csv file being:

Pressure | X-coordinate | Y-coordinate | Z-coordinate

2. The “snapshots”, i.e. each of the .csv files are then written out as binary (.bin) files. This is done to reduce the storage space occupied by the files. This is the first step taken to reduce the data size that is to be used to make useful inferences, thereby reducing the storage memory needed.
3. A nearest neighbor algorithm, *k*-d tree algorithm is implemented to achieve further data reduction. The algorithm selects a set of grid-points from the entire

`

grid-space containing all the grid-points. The pressure values at these select grid-points are written out for successive times in successive .bin files. The implementation of this algorithm is specified as follows:

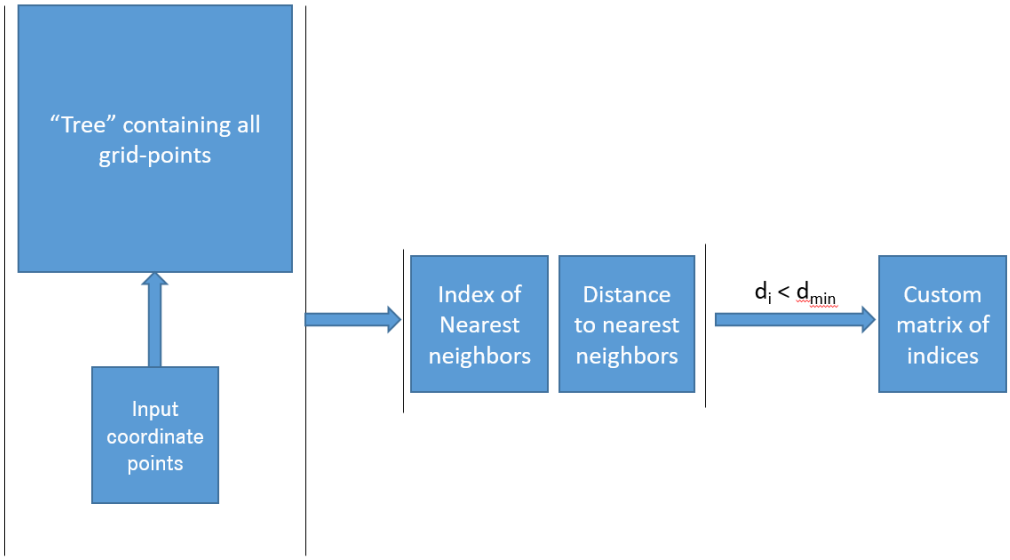


Figure 2.2: Pictorial representation of the *k*-d tree algorithm. Nearest neighbor coordinate points are found for each point in the input point set. The indices of all nearest neighbors satisfying the minimum distance criterion are stored. Subsequently, pressure values for all times are written out at the grid-points corresponding to these index values.

1. A set of point coordinates is supplied as input in a .bin format file. The coordinates of these input points are preserved as reference coordinates.
2. Each .bin format file from the previous step (step 2) is read and a "tree" of the (X,Y,Z) coordinates contained in each file is created, effectively a tree of all grid-points in the grid-space is created. So, each of the grid-points from the entire grid-space represents a "node" in the tree. The coordinates of the nearest neighbor

`

grid-point from the tree and the respective distance between each point in the input set and its nearest neighbor is found. This property of specifying a set of input points can be particularly useful as it allows the user to extract points from specific regions of their computational domain.

3. A minimum distance criterion is used to select the nearest neighbors – the indices of the grid-points from the tree that are at a distance less than the specified minimum distance from their corresponding input coordinate-point are preserved. The pressure values at these grid-points are written out for successive times in successive binary (.bin) files. Additionally, the coordinates of the chosen grid-points are preserved (these constitute the reduced grid-space). These coordinates constituting the reduced grid-space are used to “map” modes onto the geometry.
4. The binary (.bin) format files each containing pressure values at select grid-points are read and a data matrix is created. Each column of the data matrix contains pressure values at select grid-points with successive columns containing pressure values at successive times. Both modal decomposition techniques (POD & DMD) are applied on this data matrix. The *k*-d tree algorithm thus allows the user to write out a data matrix that contains values of the scalar under investigation at grid-points from a specific domain of interest.

`

2.2 Singular Value Decomposition (SVD):

Singular Value Decomposition is a mathematical technique that is integral to the implementation of both POD and DMD . SVD is a method of matrix factorization used to obtain low rank approximations to matrices. In many domains including fluid mechanics, complex systems generate data that is arranged in large matrices. For example, time-series data from a simulation or experiment may contain values of a variable of interest at different spatial coordinates arranged in a column, with each column containing values corresponding to a given time. Other examples include data sets of audio, image, weather, neural data from the brain, etc.

The point of interest in these data matrices is that the data produced is low-rank, meaning there are a few dominant patterns that explain the high-dimensional data. SVD is used to extract these patterns of data and is the foundation for the Proper Orthogonal Decompostion (POD) and Dynamic Mode Decomposition (DMD) techniques.

The low-dimensional approximations and the resulting dominant patterns are obtained directly from the data without any expert knowledge or intuition and the SVD is guaranteed to exist for any matrix.

`

Mathematics of the SVD:

The data matrices are of the form:

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \boldsymbol{x1} & \boldsymbol{x2} & \dots & \boldsymbol{xn} \\ | & | & & | \end{bmatrix}$$

with $X \in C^{mXn}$. The columns $x_k \in C^m$ may be measurements from simulations or experiments. For example, the column vectors may represent the state of a physical system that is evolving in time, such as the fluid velocity at a set of discrete points.

m and n are the dimensions of the data matrix. The columns are often called *snapshots*, and n is the number of snapshots in **X.**

The SVD performs the matrix decomposition as:

$$\mathbf{X = U\Sigma V^*}$$

where $U \in C^{mXm}$ and $V \in C^{nXn}$ are unitary matrices ($U^{-1} = U^*$, $V^{-1} = V^*$).

(* represents complex conjugate transpose; for real valued matrices, this is the same as regular matrix transpose, $X^* = X^T$)

$\Sigma \in R^{mXn}$ is a diagonal matrix with real, non-negative entries. The entries in Σ are non-negative and are ordered from largest to smallest so that

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0 \text{ where } p = \min(m,n).$$

The rank of **X** is equal to the number of nonzero singular values.

`

The implementation of SVD is done in two ways:

[U,S,V] = svd(X,'econ') %Economised or Reduced Singular Value Decomposition

[U,S,V] = svd(X) %*Singular Value Decomposition*

The difference between the two is represented pictorially:

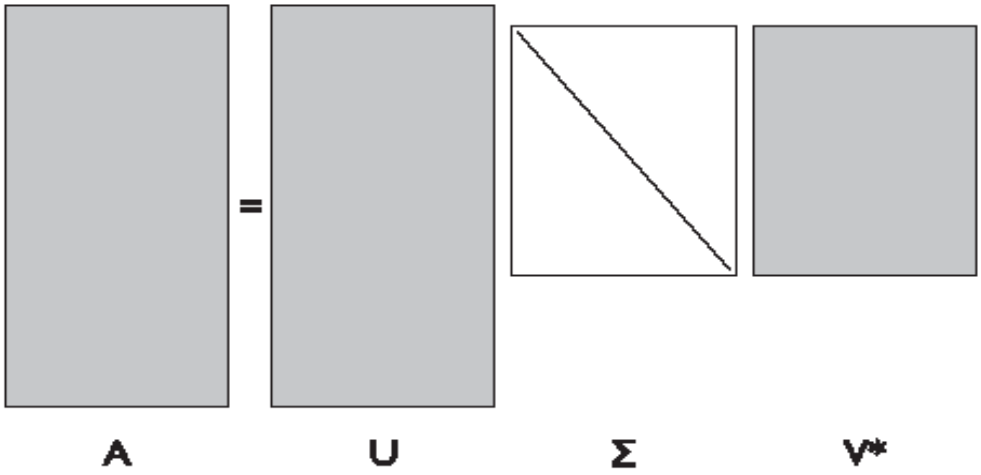


Figure 2.3: Description of the reduced SVD decomposition. Taken from [2].

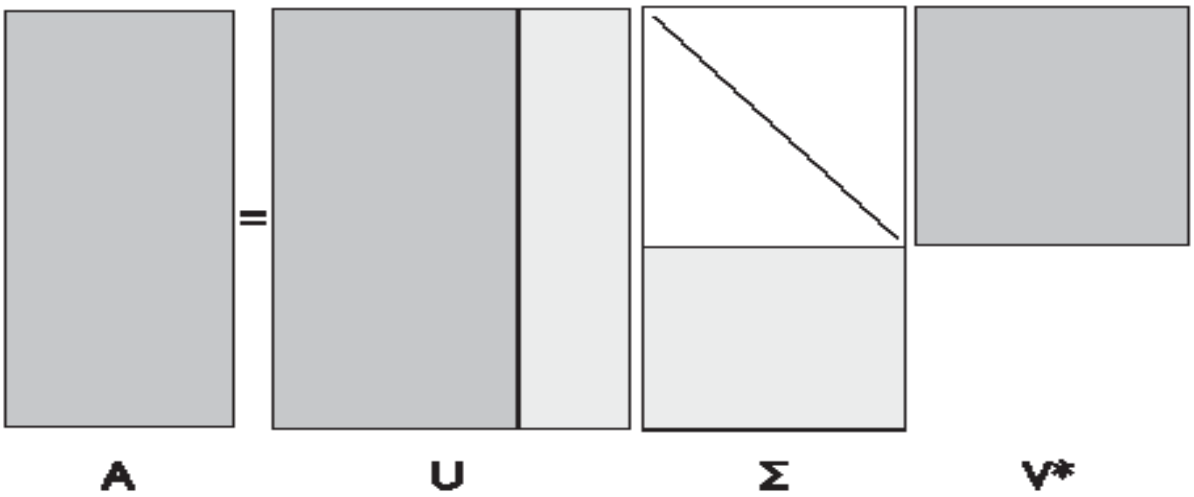


Figure 2.4: Graphical description of the full SVD decomposition. The light colored portions of U and Σ are silent rows and columns that are extended from the reduced SVD. Taken from [2].

`

If A is an *m*X*n* matrix then:

1. Reduced SVD: U is *m*X*n* , Σ is *n*X*n* and V is *n*X*n.*
2. Regular SVD: U is *m*X*m,* Σ is *m*X*n* and V is *n*X*n.*

The variation is that in going from reduced to regular SVD, an additional (*m* – *n*) columns are present in U and additional (*m* – *n*) rows of zeros are added to the Σ matrix. Due to the diagonal nature of Σ, each column of U is multiplied with corresponding element of Σ. This results in the additional columns of U being "silent". For non-square matrices A, the economy/reduced SVD is more efficient than the full SVD in terms of memory usage as the matrices U and Σ are smaller in size.

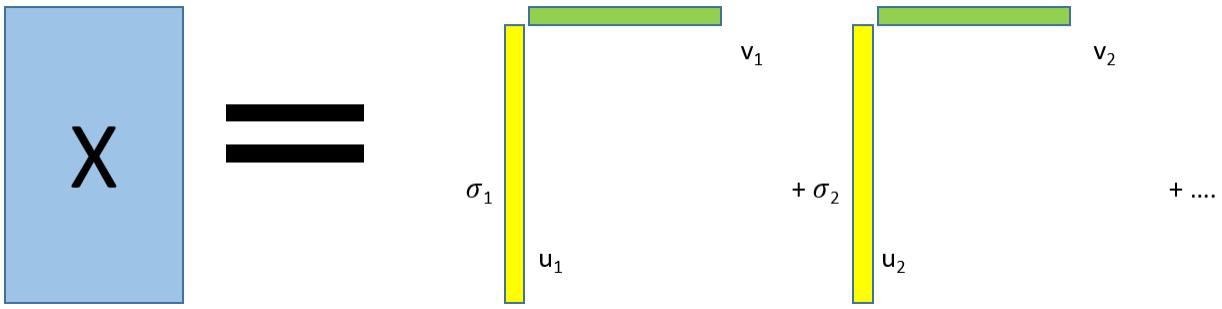


Figure 2.5: Pictorial representation of SVD, arising due to the diagonal nature of the Σ matrix. It expresses the data matrix, X as the sum of matrix products that can be truncated to obtain approximations to the data matrix.

Due to the diagonal nature of Σ and the singular values being present in descending order, *hierarchical* low-rank approximations can be found

`

conveniently by keeping the leading *r* singular values and vectors (columns of U and rows/ columns of V/V*) and discarding the rest.

The resulting approximated matrix $\widetilde{\mathbf{X}}$ is denoted by:

$$\widetilde{\mathbf{X}} = \widetilde{\mathbf{U}}\widetilde{\mathbf{\Sigma}}\widetilde{\mathbf{V}}^*$$

The rank-*r* SVD approximation is given by the sum of r distinct rank-1 matrices:

$$\widetilde{\mathbf{X}} = \sum_{k=1}^{r} \sigma_k \mathbf{u}_k \mathbf{v}_k^* = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^* + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^* + \ldots + \sigma_r \mathbf{u}_r \mathbf{v}_r^*.$$

Therefore, the original data matrix **X** can be described by a few dominant patterns given by the columns of $\widetilde{\mathrm{U}}$ and $\widetilde{\mathrm{V}}$:

$$\mathbf{X} \approx \widetilde{\mathbf{U}}\widetilde{\mathbf{\Sigma}}\widetilde{\mathbf{V}}^*$$

Dominant pattern inference:

From the SVD representation of X ($X = U\Sigma V^*$) and the unitary nature of the matrices U and V ($U^{-1} = U^*$, $V^{-1} = V^*$), the following can be obtained:

$$XX^*U = U\Sigma^2$$

$$X^*XV = V\Sigma^2$$

So, the columns of U are eigenvectors of the matrix XX* and the columns of V are the eigenvectors of the matrix X*X. The values of the singular values are arranged in descending order resulting in the columns of U being hierarchically ordered by the correlation they capture in the columns of X. Similarly, the columns of V capture the correlation in the rows of X.

`

Physically, if the columns of X are spatial measurements in time, then columns of U encode spatial patterns and columns of V encode temporal patterns in hierarchical order.

2.3 Proper Orthogonal Decomposition (POD):

Proper Orthogonal Decomposition (POD) is a technique for dimensionality reduction used to study complex systems which can often be characterized by low-dimensional dominant patterns. POD aims to capture these patterns to accurately model the full spatial-temporal evolution of the governing complex system. It is based on the SVD algorithm and so produces a set of *modes* that are *optimal* for representing the data matrix **X.** Using SVD, it allows reduction in the number of *modes* required to accurately describe behavior of the data.

To start the construction of optimal POD modes, the dynamics of the system are captured in a data matrix **X** at a prescribed time interval:

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \boldsymbol{u1} & \boldsymbol{u2} & \dots & \boldsymbol{un} \\ | & | & & | \end{bmatrix}$$

As mentioned previously, SVD on the data matrix **X** results in the following formulation:

$$\mathbf{X = U\Sigma V^*}$$

$X \in C^{mXn}$

`

$U \in C^{mXm}$, $V \in C^{nXn}$ and are both unitary matrices

$\Sigma \in C^{mXn}$ is a diagonal matrix of non-negative values arranged in descending order called *singular values.*

If columns of X describe spatial distribution at a given time, then columns of U are the spatial POD modes that describe the dominant spatial patterns of the scalar listed in X. The columns of matrix V give the time-history of the modes and constitute the temporal POD modes that describe the dominant temporal patterns. The values in Σ give the weighting of each mode relative to the others. Extraction of the *dominant* modes is in-built in the POD algorithm as the POD modes produced are ranked on the basis of their relative contribution to the original data.

In this way, POD helps to reduce the memory usage (by allowing user to neglect modes that that have relatively less contribution to the original data) and make the study of flow physics more efficient (by capturing the most dominant spatial and temporal patterns seen in the data).

The total number of POD modes generated is equal to the number of snapshots *n* in the data matrix X. (where normally $m >> n$). The objective is to determine the minimal number of modes necessary to represent the data matrix X with sufficient accuracy.

`

This is achieved by performing rank-*r* approximation to the true dynamics where *r<n:*

$$\mathbf{X} \approx \widetilde{\mathbf{U}}\widetilde{\mathbf{\Sigma}}\widetilde{\mathbf{V}}^{*}$$

The optimal POD modes are then given by the columns of the truncated matrix $\widetilde{\mathbf{U}}$**:**

$$\widetilde{\mathbf{U}} = \mathbf{\Psi} = \begin{bmatrix} | & | & & | \\ \mathbf{\Psi 1} & \mathbf{\Psi 2} & \dots & \mathbf{\Psi} \boldsymbol{r} \\ | & | & & | \end{bmatrix}$$

where the truncation preserves the *r* most dominant modes. This method of extracting dominant patterns in the data is an equation-free method as it doesn't make any use of the governing equations. Thus, optimal representation of the dynamics doesn't require any knowledge of the complex system.

However, one disadvantage of the POD algorithm is that it fails to capture frequency specific dynamics. While it is possible to obtain the frequency spectra of the temporal POD modes (columns of matrix V) and hence obtain the dominant frequencies in each POD mode, each POD mode has peaks at multiple frequencies. In this manner, dynamical information corresponding to specific frequency becomes obscured. This shortcoming is overcome through the implementation of Dynamic Mode Decomposition (DMD).

`

2.4 Dynamic mode Decomposition (DMD):

DMD is a technique for performing modal decomposition that is used to extract dynamical information from high-dimensional data. It is an equation-free, data-driven method that extracts dynamical information in terms of spatial-temporal coherent structures that can also be used for short-term prediction of the future state of the system in terms of the data. The advantage of the DMD algorithm is that it produces frequency specific dynamical information. Each DMD mode generated through the algorithm has a single frequency and growth rate associated with it. Hence, it is possible to obtain relative spatial distributions and relative phase diagrams corresponding to a particular frequency and growth rate.

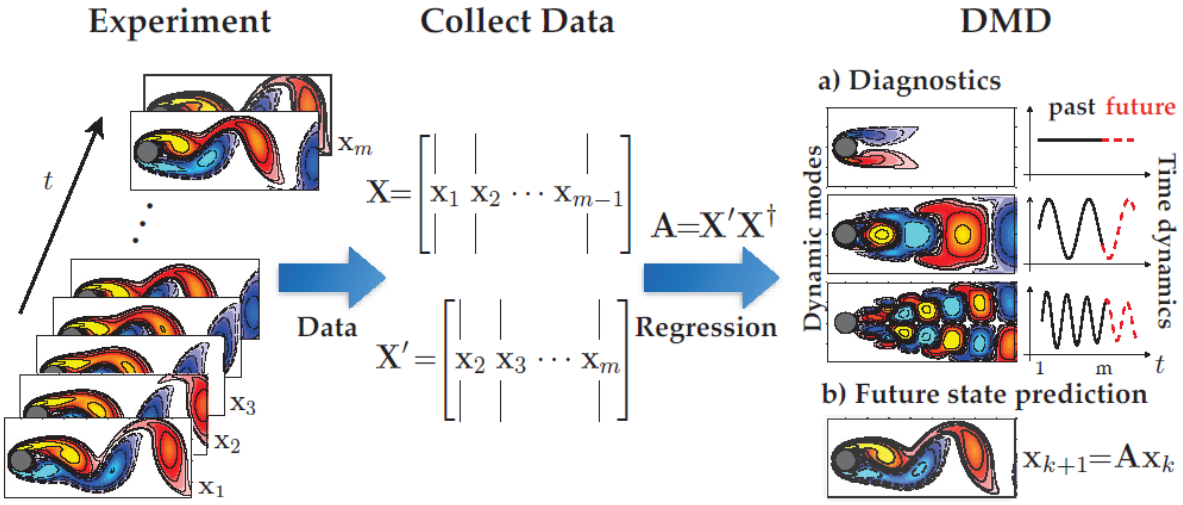


Figure 2.6: Schematic of DMD performed on a fluid flow example. The result consists of a set of DMD modes, each associated with an eigen value that is used to compute the frequency (Hz) and growth rate (Hz) of the modes. Taken from [3].

`

The input for DMD, like the POD is the data matrix **X (*mXn)*** given by:

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \boldsymbol{u1} & \boldsymbol{u2} & \dots & \boldsymbol{un} \\ | & | & & | \end{bmatrix}$$

The output produced is a set of DMD *modes* but differing from the POD modes in the fact that each DMD mode has an associated frequency of oscillation and growth/ decay rate.

DMD is algorithmically a regression of data onto locally linear dynamics given by:

$$\mathbf{x_{k+1} = Ax_k}$$

where **A** is chosen to minimize $|| x_{k+1} - Ax_k ||_2$ over the k = 1,2,3 … (n-1) snapshots. This method of modal decomposition is equation-free and makes no assumptions about the physics of the system. However, there is only a short time window in the future where predictions made using the DMD modes are accurate as the predictions of the state of a nonlinear dynamical system are made using a linear dynamical model. Additionally, DMD modes are not necessarily ordered hierarchically by the dynamical information they capture in the data. A metric called *normalized mode magnitude* has been used in both case studies of this report to quantify the contribution of DMD modes relative to each other. However, discrepancies are seen between the peak frequencies for the

`

dominant temporal POD modes and frequencies corresponding to the DMD modes having large normalized mode magnitude.

Mathematics of DMD:

Data is first collected from a dynamical system:

$$\frac{dx}{dt} = f(x, t{:}\mu)$$

where $x(t) \in R^m$ is a vector representing the state of the system at time t, μ contains parameters of the system and f() represents the dynamics.

Measurements of the system are made at times t_k from k = 1,2,…n for a total of n measurement times. These measurements are used to construct the data matrix **X** which is used to obtain the DMD modes that contain dynamical information about the system.

The DMD algorithm approximates the non-linear dynamical system as a linear dynamical model given by:

$$\frac{dx}{dt} = Dx$$

The solution of this equation is given as

$$x(t) \approx \sum_{k=1}^{r} \phi_k \exp(w_k t) b_k = \phi \exp(\Omega t) b$$

where ϕ_k and w_k are the eigenvectors and eigenvalues of the matrix A, and the coefficients b_k are the coordinates of x(0) in the eigenvector basis.

`

The approximated linear model is used to calculate values at successive time-steps as:

$$x_{k+1} = Ax_k$$

where

$$A = \exp(D\Delta t)$$

The objective then is to calculate the matrix A that approximates the nonlinear system as a linear system. This is done by minimizing the approximation error:

$$||x_{k+1} - Ax_k||_2$$

To minimize the approximation error, the data matrix **X** is arranged into two data matrices:

$$\mathbf{X_1} = \begin{bmatrix} | & | & & | \\ \boldsymbol{x1} & \boldsymbol{x2} & \dots & \mathbf{x}(\boldsymbol{n}-\mathbf{1}) \\ | & | & & | \end{bmatrix}$$

$$\mathbf{X_2} = \begin{bmatrix} | & | & & | \\ \boldsymbol{x2} & \boldsymbol{x3} & \dots & \mathbf{x}(\boldsymbol{n}) \\ | & | & & | \end{bmatrix}$$

Then, as described above, the locally linear approximation is written as:

$$X_2 = AX_1$$

$$\Rightarrow A = X_2X_1^p$$

(* here ‘p’ denotes the Moore-Penrose pseudoinverse).

`

The DMD modes are the eigenvectors of A and each DMD mode corresponds to a particular eigenvalue of A.

To calculate the eigenvalues and eigenvectors of A, the following steps are taken:

1. Singular Value Decomposition (SVD) of $\mathbf{X_1}$ is performed:

$$\mathbf{X_1} \approx \mathbf{U\Sigma V^*}$$

2. The *r*X*r* projection of the matrix **A** labelled as $\widetilde{\mathbf{A}}$ obtained as:

$$\widetilde{\mathbf{A}} = \mathbf{U^*AU} = \mathbf{U^*X_2V\Sigma^{-1}} \text{ (using } A = X_2X_1^p)$$

3. The eigen decomposition of $\widetilde{A}$ is performed to obtain its eigenvalues (λ) and eigenvectors (W).

4. The eigenvalues of A are the same as those of $\widetilde{A}$. The eigenvectors of A (DMD modes, labelled $\boldsymbol{\phi}$) can be calculated in two ways:

a. $\Phi = X_2V\Sigma^{-1}W$. In this case, the modes are called *exact DMD modes.*

b. $\Phi = UW$. In this case, the modes are called to as *projected DMD modes.*

As per [3] exact formulation should be used if a mode associated with zero eigenvalue is to be found. Otherwise projected formulation should be used.

For a detailed explanation of the theory, algorithm and motives, the reader is referred to [3].

Physical interpretation:

The values of the scalar at each time can be calculated using:

`

$$x(t) \approx \sum_{k=1}^{r} \phi_k \exp(w_k t) b_k = \phi \exp(\Omega t) b \quad \text{(E1)}$$

Here w_k are the scaled eigenvalues given by

$$w_k = \log(\lambda_k)/\Delta t$$

For real-valued data as is the case for both case studies, the eigenvalues of A are either complex-conjugate or real values. Calculating log of a complex valued expression results in the following formulation:

$$\log(z) = \log(a + ib) = \log(|z|) + 1i*\arg(z)$$

On the right hand side of the equation [E1], exponent of this expression times 't' is performed.

This results in the real(w) = magn(eigenvalue) quantifying the growth/decay rate while the imag(eigenvalue) gives the frequency of the respective DMD mode x(t).

The units of the growth rate and frequency of each mode are converted to Hertz (Hz) by dividing real and imaginary part of the eigen value of the mode by $2\pi dt$:

Frequency (Hz) = imag(log(eigenvalue))/ $2\pi dt$

Growth Rate (Hz) = real(log(eigenvalue)) / $2\pi dt$

Hence, DMD modes can be classified as growing, damped or stable depending on whether their eigenvalue magnitudes are greater, less than or equal to 1. Flow instabilities associated with modes having a positive growth rate amplify with

`

time while those associated with modes having a negative growth rate attenuate with time.

Explanation: The evaluation of x(t) involves the calculation of the exponent of a complex valued expression. $e^{(p + iq)t} = e^{pt}. e^{iqt} = e^{pt}. (\cos(qt) + i\sin(qt))$

The vector b is composed of the coefficients b_k which are the intial amplitudes of the DMD modes.

For each mode, the elements within the mode are complex value. Physically,

1. The magnitudes of each element represent their relative contribution in that mode. Hence, the mode diagram provides a pictorial representation of the values of the scalar (in this case pressure) in the domain at the frequency associated with a DMD mode.
2. The phases of the complex valued elements represent the relative phase of the elements in that particular mode. Spatial coordinates having the same values of relative phase peak in their scalar values (in this case pressure) at the same time. Spatial coordinates having positive values of relative phase contribute to harmonics that grow with time while those having negative phase values contribute to harmonics that attenuate with time. In this manner, the relative phase diagram provides a pictorial representation of the nature of wave propagation at the frequency associated with a DMD mode. This has profound

`

implications; for example in the allocation of medical resources (refer CH-11 of [3])

`

CHAPTER 3: FLOW PAST A BLUFF BODY

The CFD simulation parameters including geometry, mesh and solver settings for this case study of flow past a bluff body are taken from the aeroacoustics tutorial of a commercial CFD software. This case study is performed to test the accuracy of the two modal decomposition techniques, POD and DMD in their ability to capture dominant spatial-temporal features such as the dominant frequencies, which are already available for this study through FFT analysis done on the CFD software.

3.1 Computation Method:

The bluff body in this case study is a cylinder. The Reynolds number used is 63,800, which being in the subcritical region range of $300 \leq Re \leq 150{,}000$ results in a Strouhal number of 0.22. Strong and periodic vortex shedding is observed exhibiting Karman vortex street. The geometry as shown in figure 3.1 consists of a cylinder of 20 mm diameter, placed at the center of a three-dimensional circular computational domain, 0.6 m in dimeter. The free stream air velocity and temperature are set to 50 m/s and 300K, respectively resulting in a Mach number of 0.144 and a Reynolds number of 63,800.

`

A volumetric control is created for mesh refinement, to capture the separation behind the cylinder, and to capture the vortex shedding. The mesh generated is shown in figure 3.2.

K-Omega turbulence model is implemented with a turbulence under-relaxation factor of 0.8 and a turbulent viscosity under-relaxation factor of 1. Implicit unsteady solver is used with a time-step of 2.5E-5 seconds. Pressure (Pa) values are written out every 8 time-steps resulting in a sampling frequency of 5000 Hz. Consequently a total of 600 .csv format files each containing pressure values at 826,890 grid points are generated.

3.2 Geometry:

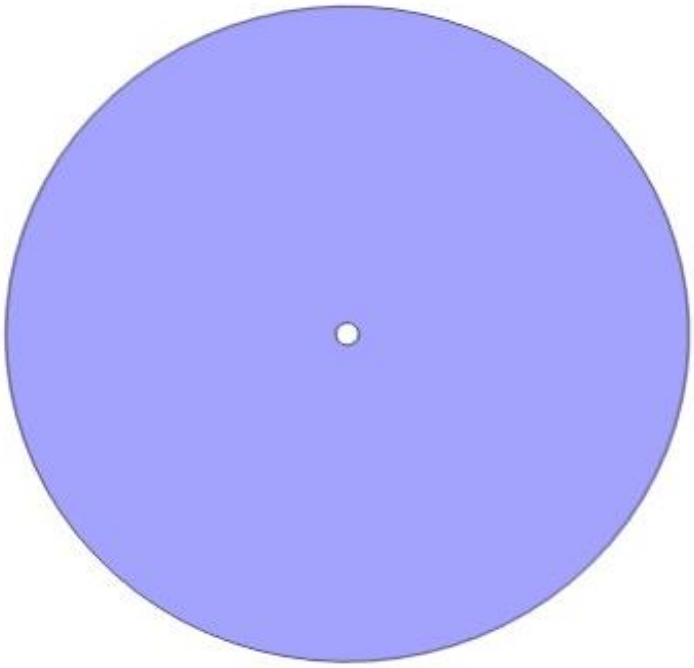


Figure 3.1: Geometry of the case study - flow past a bluff body. The bluff body in this case is a cylinder of diameter 20 mm.

`

3.3 Mesh:

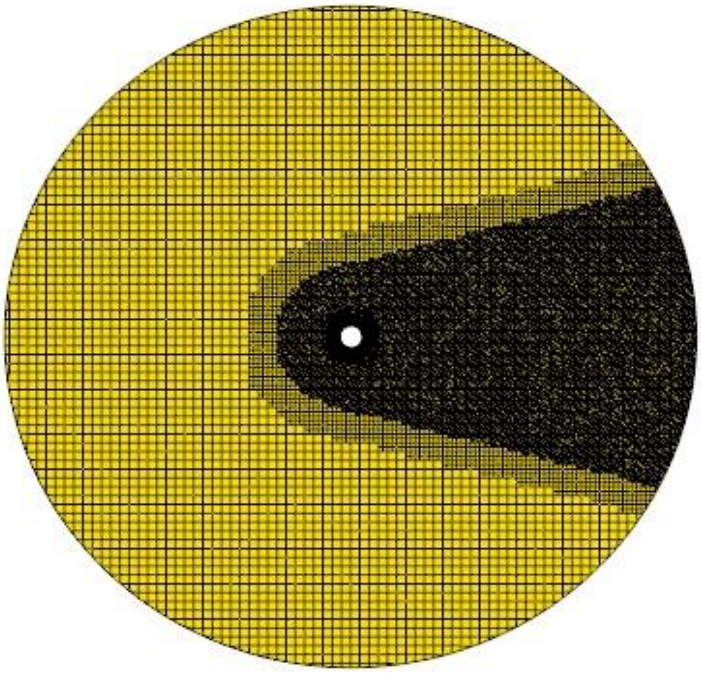


Figure 3.2: Mesh generated for simulating flow past a cylinder. A volumetric control is setup for mesh refinement to capture separation behind the cylinder and the vortex shedding.

3.4 Creation of the data matrix:

1. 600 .csv files, each corresponding to an individual time step contain information

   in the following format:

   P(in Pa) | X | Y | Z

   These files total a size of approximately 35 GB and are written out from the CFD

   simulation.

`

2. Each of the “snapshot” matrices are written out in .bin format using code. This is done to reduce the memory usage and make subsequent operation on the snapshot matrices faster. Figure 3.3 shows the data reduction that has been achieved for this case study of flow past a bluff body by converting all .csv format snapshots into individual .bin format files.

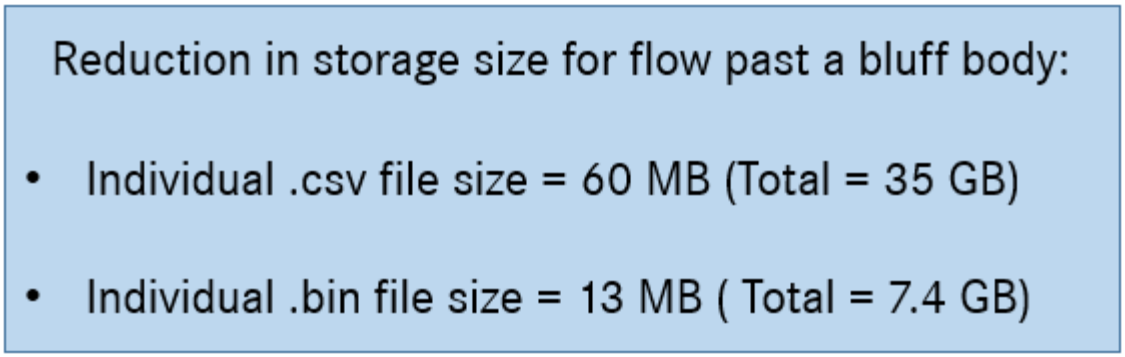


Figure 3.3: Reduction in storage size achieved for the case of flow past a bluff body through conversion of .csv format snapshots to .bin format

3. The *k*-d tree algorithm is implemented and the input set of coordinate points are 10,000 points randomly selected from the grid-space itself. As a result, as per the algorithm, the nearest neighbor for each point in the input set are the points themselves as the same points are also present in the “tree” created for all points in the grid-space. Consequently, pressure values for all times at these 10,000 randomly selected points are written out in individual .bin format files. Additionally, the coordinates of the chosen grid-points are preserved (these constitute the reduced grid-space). As can be observed from figure 3.4,

`

inferences about the spatial-temporal features of the flow being investigated are made using only 32 MB of data instead of the original 35 GB.

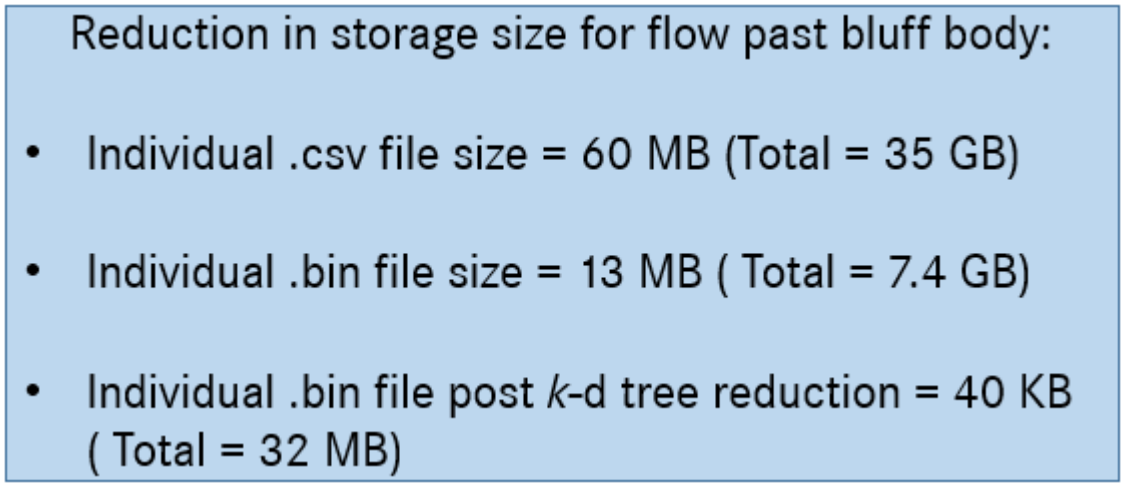


Figure 3.4: Reduction in storage size achieved for the case of flow past a bluff body. through the implementation of *k*-d tree algorithm.

The size of each of the .bin format files containing pressure values is 40 KB. A total of 600 such files are written out having a total size of 23.4 MB. The file containing the coordinates of the points in reduced grid-space containing 10,000 grid-points is 8.6 MB.

4. These binary (.bin) format files each containing pressure values at select grid-points are read and a data matrix is created. Each column of the data matrix contains pressure values at select grid-points with successive columns containing pressure values at successive times.

`

3.5 Results of POD:

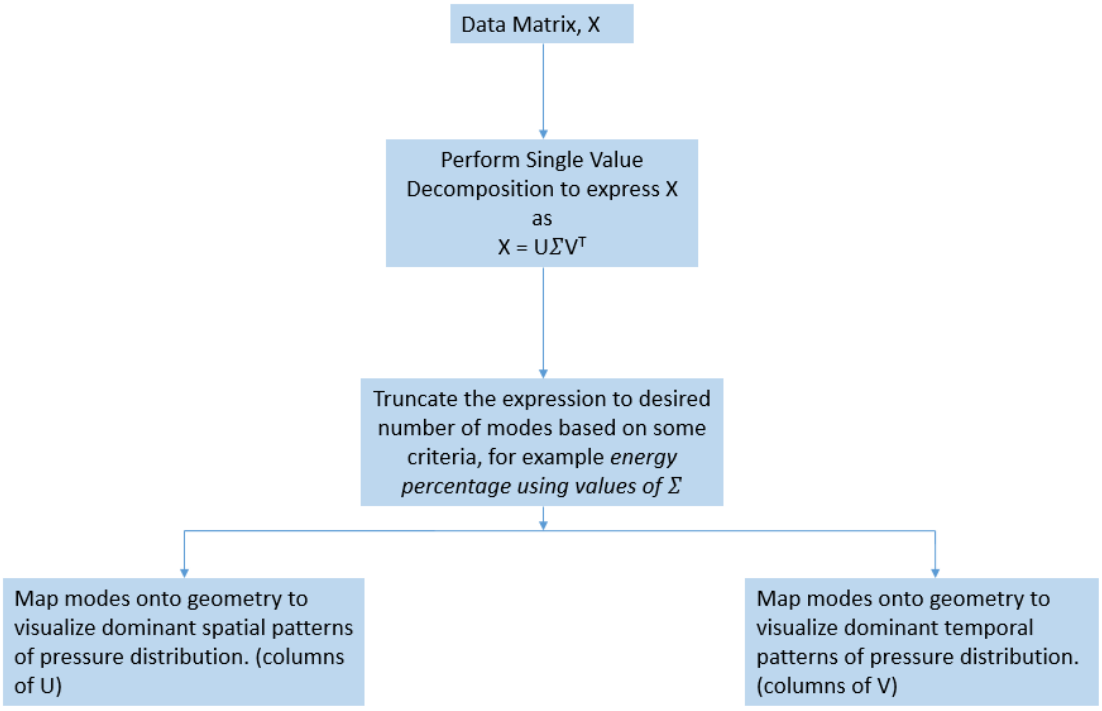


Figure 3.5: Workflow of the POD algorithm. SVD is applied to the data matrix created. The resulting columns of the matrix U (left singular vectors) are the spatial POD modes while the columns of the matrix V (right singular vectors) are the temporal POD modes.

The POD algorithm as described in the introductory text is applied to the data matrix created for the transient pressure data of flow around cylinder. Data is sampled from 826,890 grid points over 600 timesteps.

The results of POD analysis on this case study are presented as follows:

`

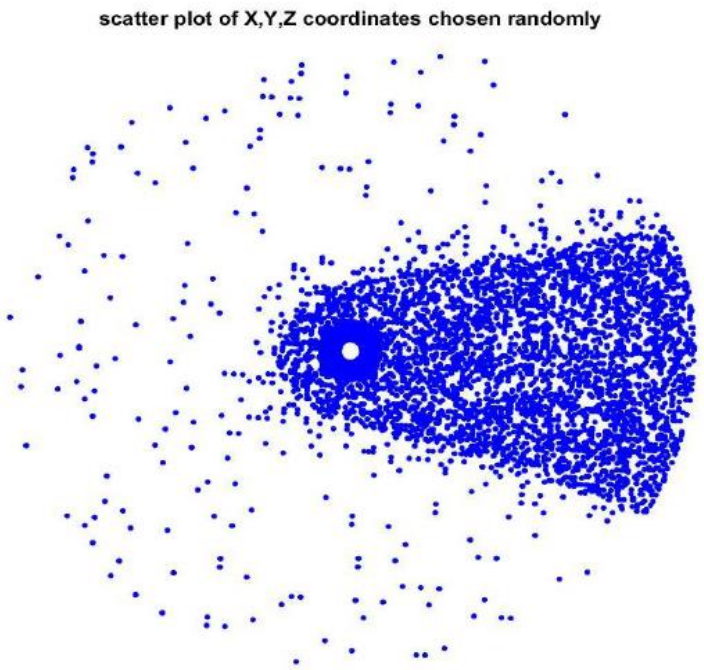


Figure 3.6: 10,000 coordinate points (X,Y,Z) were chosen randomly from the grid of 826,890 grid points using *k*-d tree algorithm resulting in a data reduction from approximately 35 GB to 32 MB.

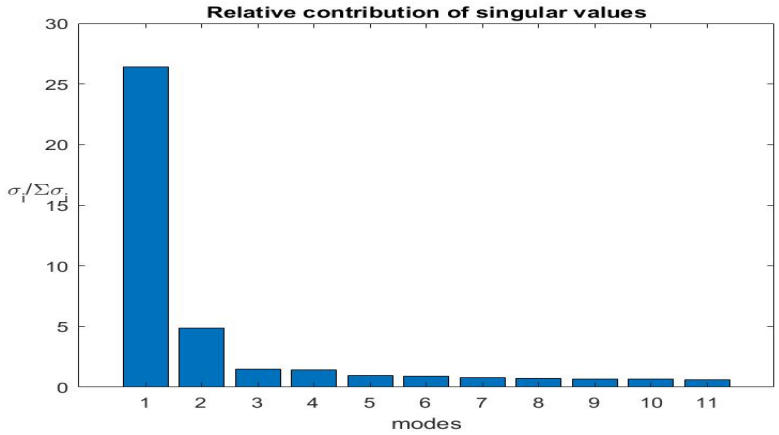


Figure 3.7: Relative contributions of singular values (values of Σ) can be used as a measure to decide appropriate truncation point to approximate data matrix X i.e the sum

$\widetilde{\mathbf{X}} = \sum_{k=1}^{r} \sigma_k u_k v_k^* = \sigma_1 u_1 v_1^* + \sigma_2 u_2 v_2^* + \ldots + \sigma_r u_r v_r^*$.

`

In this case, relative contributions are: σ_1 = 26.42, σ_2 = 4.87, σ_3 = 1.47, σ_4 = 1.44. The first mode (spatial & temporal POD mode) capture approximately 26 % of the pressure data in the data matrix in terms of singular values.

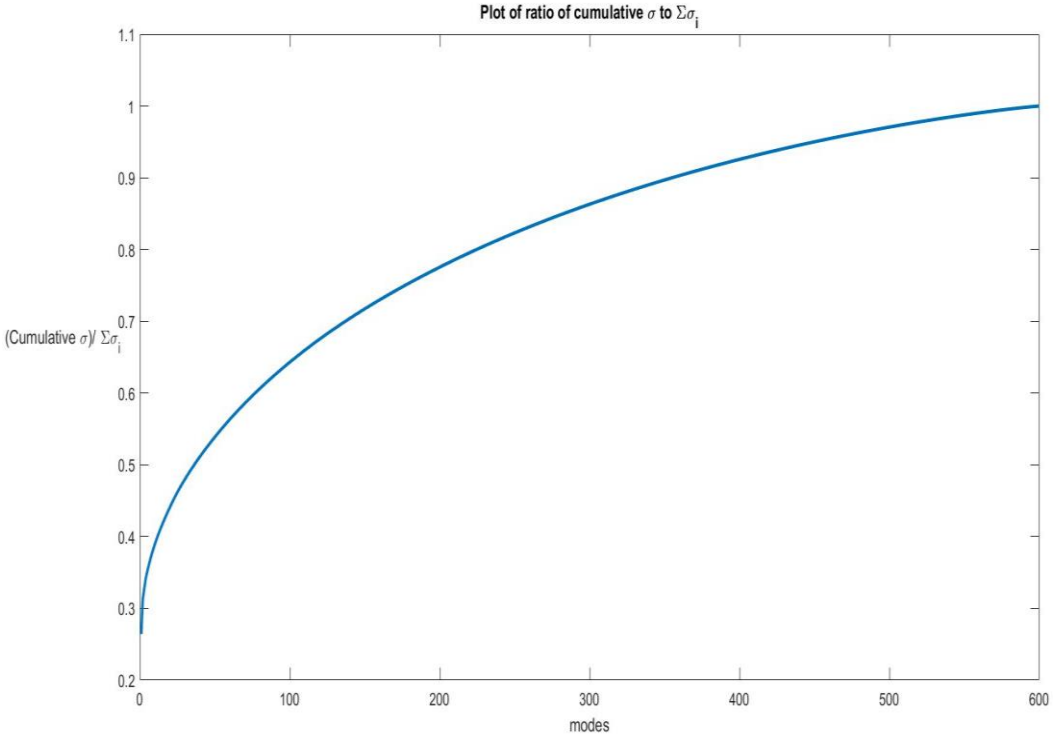


Figure 3.8: 450 modes are needed to capture 95% of the energy in terms of the singular values. A total of 600 snapshots result in 600 columns in U and V matrix i.e. 600 spatial modes and 600 temporal modes. The last 25% i.e., 150 columns of U and V matrix capture only 5% of the features in the data. Hence, these modes can be neglected resulting in lesser data that has to be focused on. This makes drawing inferences from the data easier.

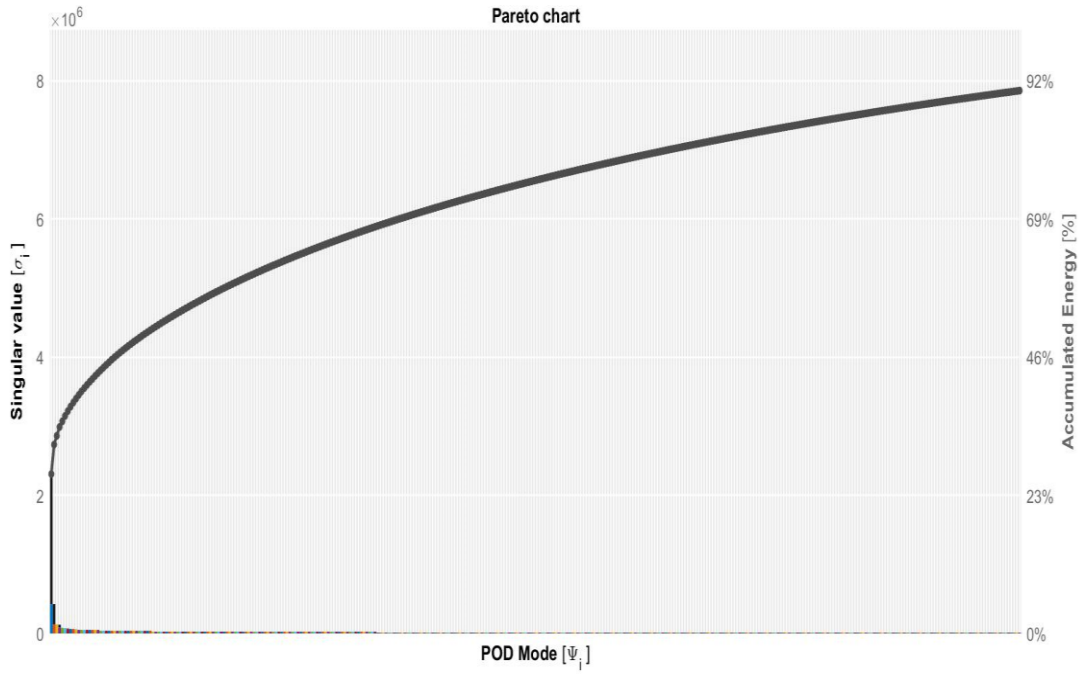


Figure 3.9: Pareto chart of singular values with left axis denoting absolute values and right axis denoting relative contributions. The bar graph is the graph of values of singular values and the line graph is the graph of cumulative values of singular values.

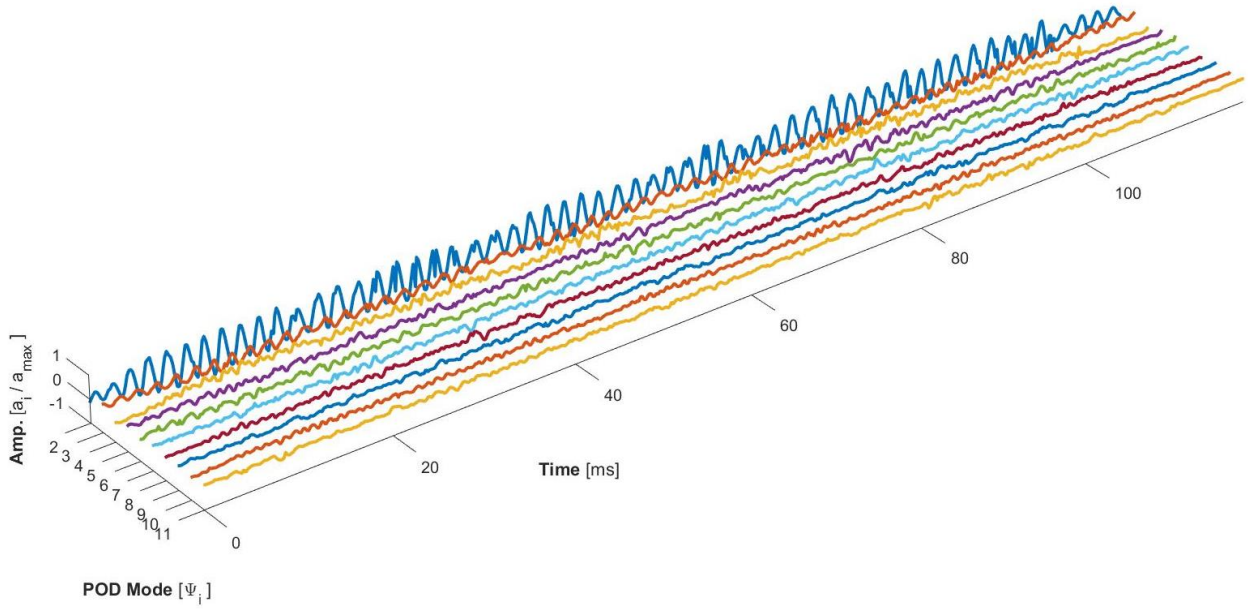


Figure 3.10: Temporal variation of POD modes 2 through 11 with scaled amplitudes. These modes are used to create the frequency spectra, which gives the measure of the dominant frequencies present in the data. Power Spectral

`

Density (PSD) computations for each temporal POD mode are carried out using Welch's method to compute the peak frequencies for each POD mode.

As mentioned, the columns of matrix U and V denote the spatial and temporal POD modes, respectively. Thus, each spatial POD mode is a column vector with each value in the vector corresponding to a spatial coordinate preserved in a file. This is used to "map" the spatial modes onto the geometry.

Similarly, each temporal POD mode (columns of matrix V) is a column vector with each value corresponding to an instant of time. Thus, each temporal POD mode is a time-signal on which a spectral analysis can be performed to determine the dominant frequencies for each temporal POD mode. The first 4 spatial and temporal POD modes are plotted in figures 3.11 to 3.14. Modes 5 through 11 are plotted in figure A.1 to A.8 in the Appendix.

`

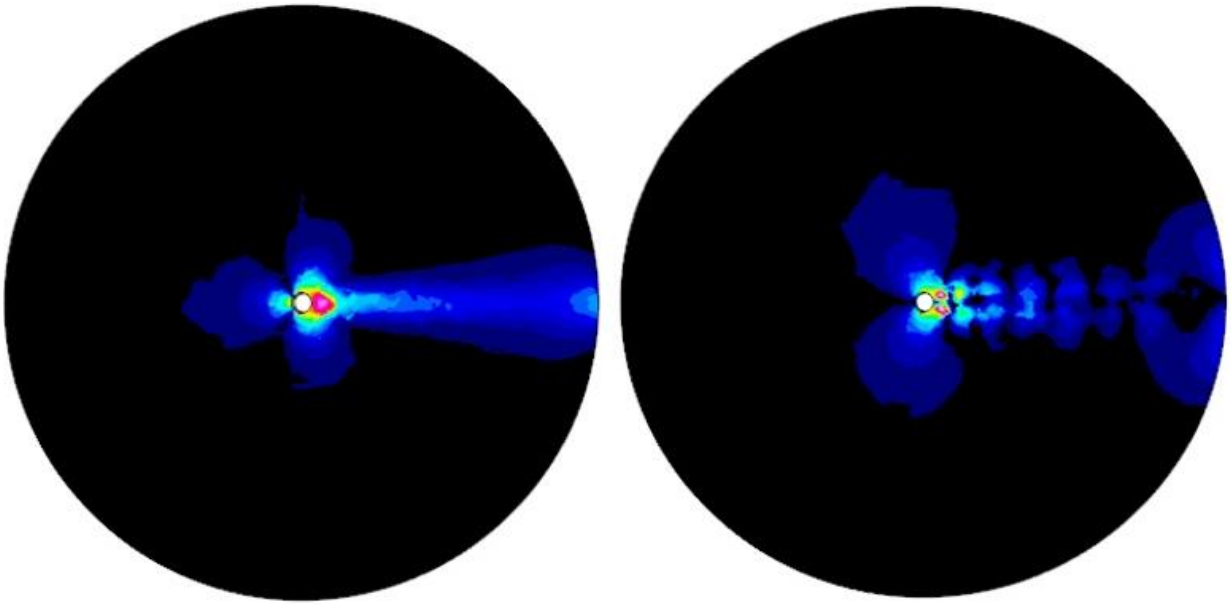


Figure 3.11: Spatial POD modes 1 (left) and 2 (right). They represent the most dominant spatial patterns of pressure seen in the data matrix X. The first spatial POD mode is the “mean” mode i.e. the first mode shows the mean pressure distribution across time. The modes are normalized using the maximum numerical value in the respective mode. Hence, the scale is from 0 to 1.

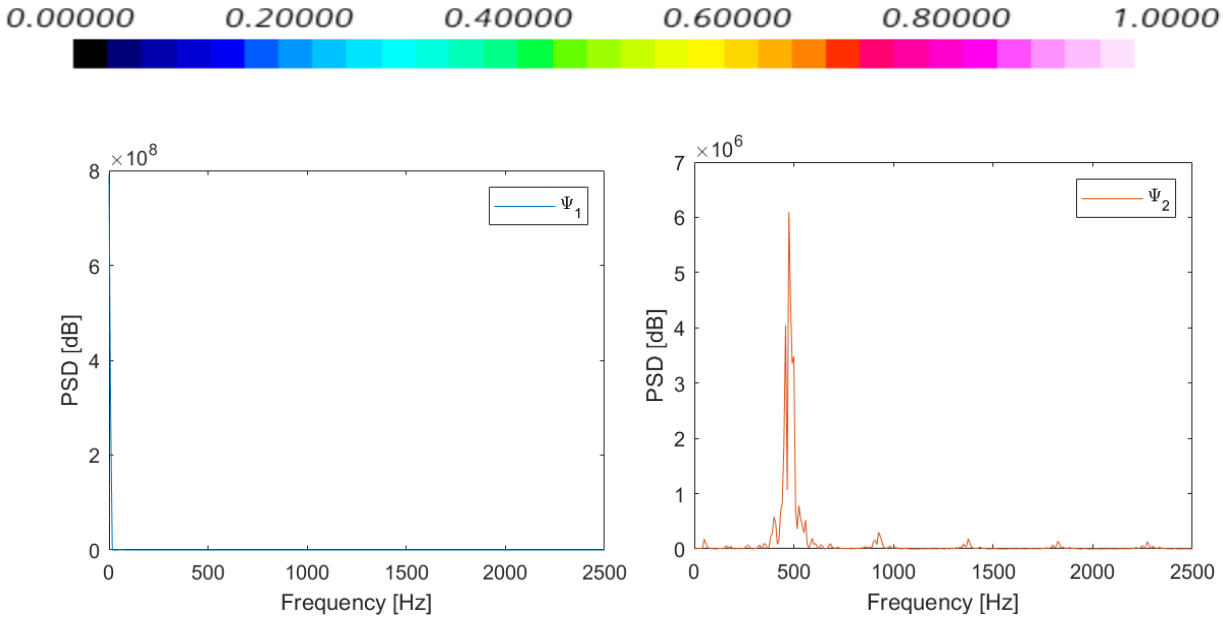


Figure 3.12: Temporal POD modes 1 (left) and 2 (right). They represent the most dominant temporal patterns of pressure seen in the data matrix X i.e. the most dominant frequencies seen in the data. The first mode has a peak at 0 Hz as it is a

background mode (representing the mean temporal variation). Mode 2 has a peak at 475 Hz.

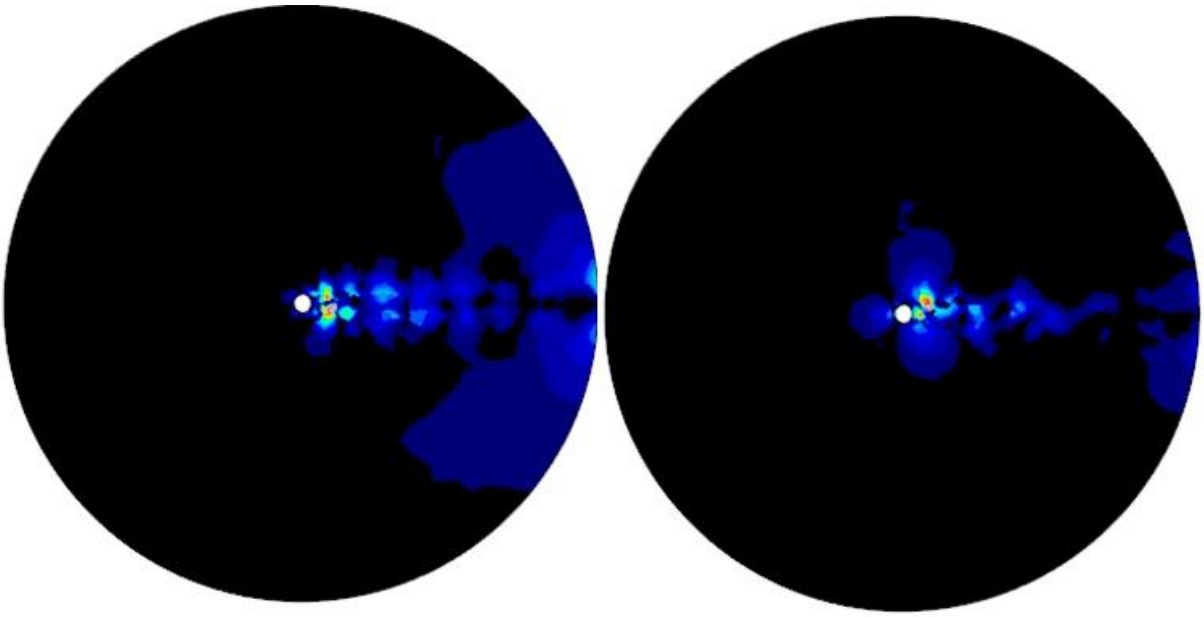


Figure 3.13: Spatial POD modes 3(left) and 4 (right), showing the relative contributions of spatial coordinates.

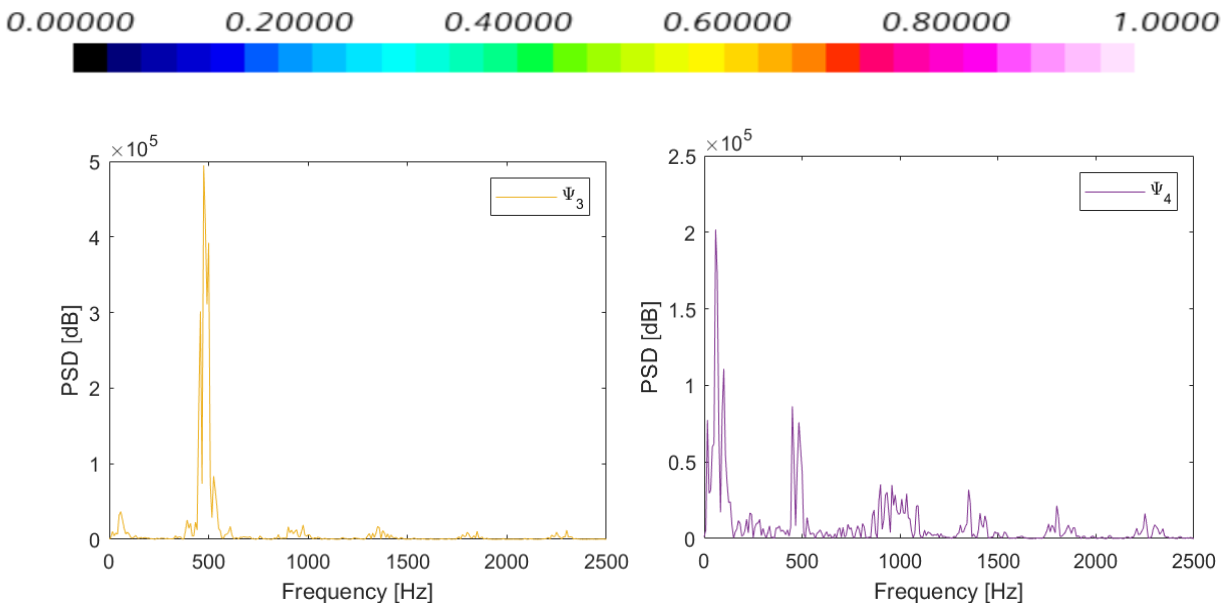


Figure 3.14: Temporal POD modes 3 (left) and 4 (right). Mode 3 has a peak at 475 Hz, mode 4 has peaks at 58.33 Hz, 100 Hz, 450 Hz, 483.33 Hz. As can be observed, mode 4 has peaks at multiple frequencies. Hence, dynamical information

corresponding to individual frequencies becomes obscured. This shortcoming of POD is overcome through the use of DMD.

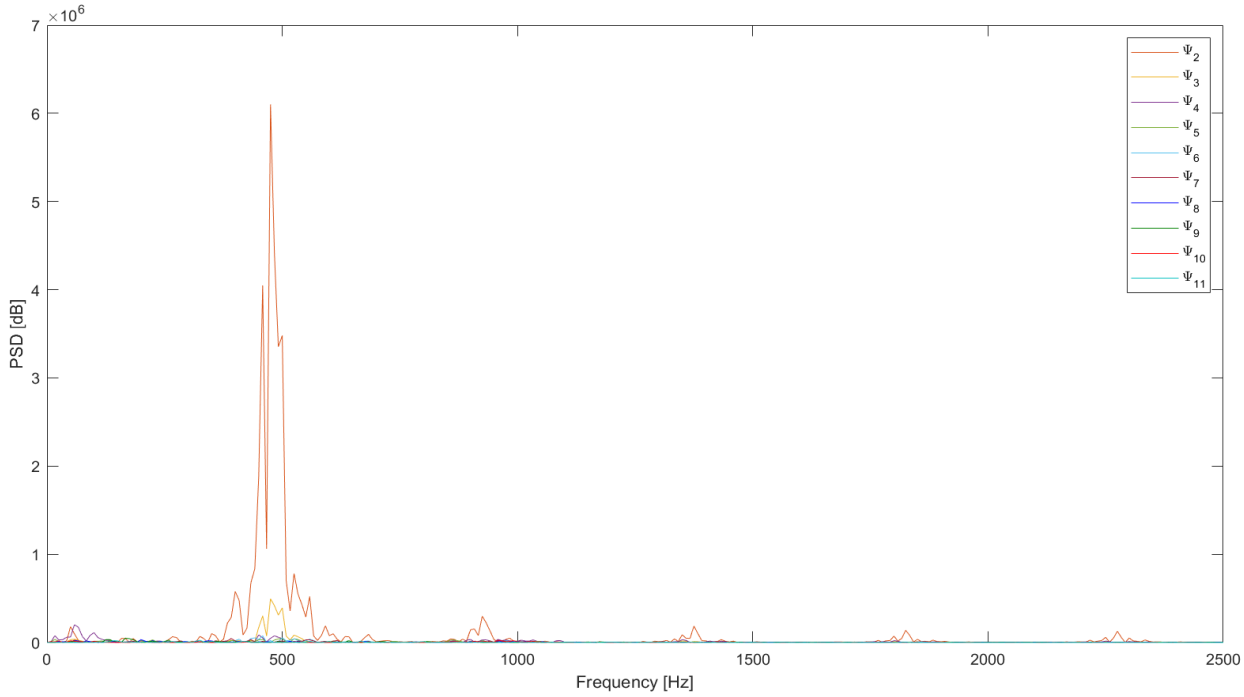


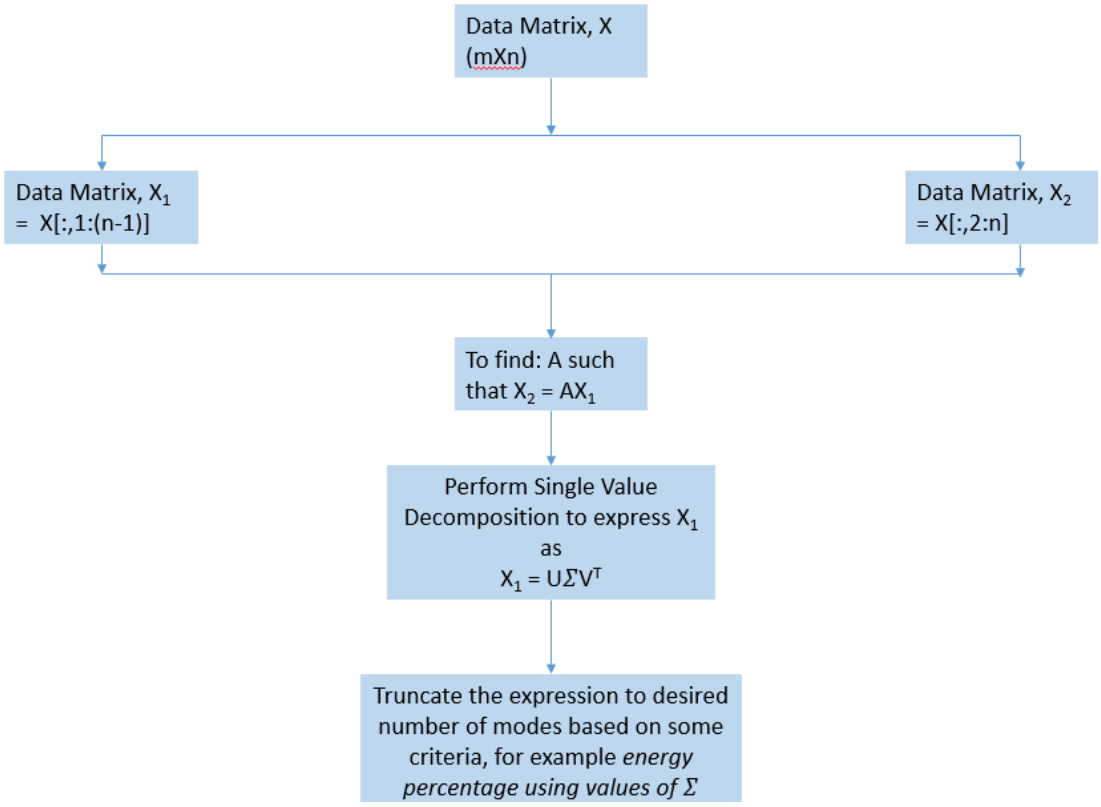
Figure 3.15: Power Spectral Density (PSD) curves of POD modes 2 through 11 give information about the dominant frequencies present in the data. There is a peak at frequency close to 500 Hz (exact value is 475 Hz) and another close to 1000 Hz (exact value is 925 Hz). This is in line with the results obtained using FFT analysis for the aeroacoustics tutorial on the CFD software. Hence, POD is able to successfully capture the peak frequencies whilst also being applied on significantly lesser data (32 MB instead of 35 GB).

As mentioned, POD modes have peaks at multiple frequencies. Hence, it is not possible to obtain relative spatial distribution at individual frequencies. This shortcoming serves as the motivation for performing Dynamic Mode Decomposition (DMD). Using DMD it is possible to obtain spectral information

such as mode frequency and phase, i.e. modal information at individual frequencies.

3.6 Results of DMD:

The DMD algorithm as described in the introductory text is applied to the data matrix created for the transient pressure data of flow over cylinder. Data is sampled from 826,890 grid points over 600 timesteps.



`

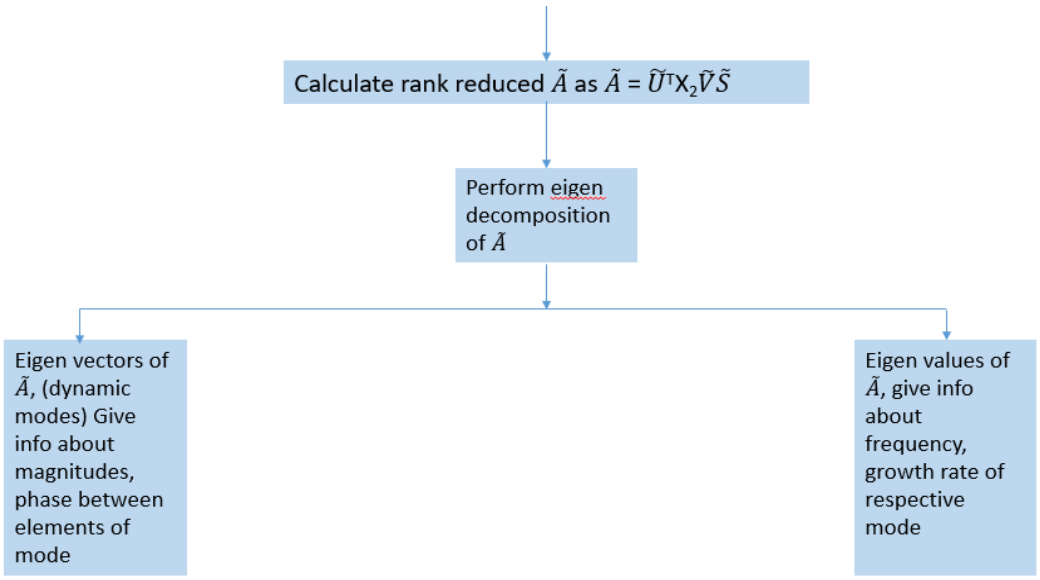


Figure 3.16: Workflow of the DMD algorithm. The data matrix X is used to create two matrices X_1 and X_2 to construct a locally linear dynamical system. SVD is applied to the matrix X_1 and the eigen vectors and eigenvalues of the rank reduced matrix $\widetilde{A}$ are used to obtain the modes and their associated frequency and growth rate.

The results of DMD analysis on this case study are presented as follows:

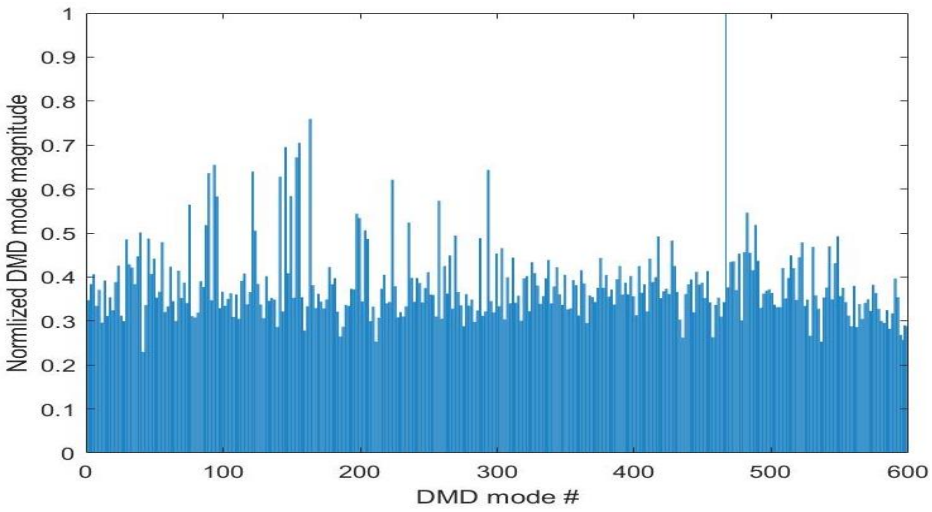


Figure 3.17: Each DMD mode (represented by a column) is normalized using the maximum numerical value in the respective mode. The magnitudes i.e. norms of each of these column vectors are computed which are further normalized using

`

the maximum norm value. This is done to quantify the relative contribution of each DMD mode in capturing the dynamical information in data matrix X. Hence, *normalized DMD mode magnitude* is a metric to hierarchically order the modes by the importance of dynamical information contained in the mode.

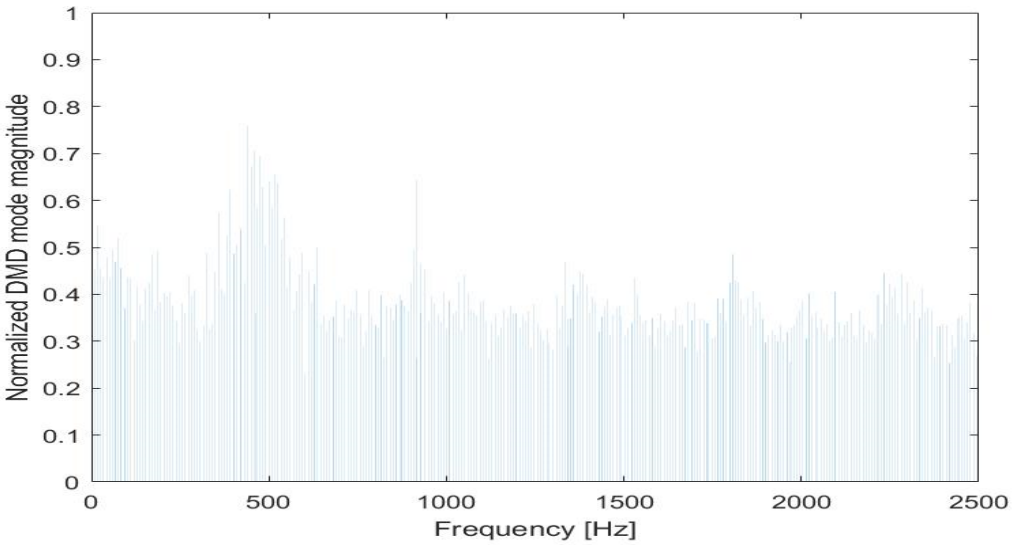


Figure 3.18: Spectra of DMD analysis exhibits the magnitudes associated with the individual modes corresponding to discrete frequencies. Peaks are observed at frequencies close to 500 Hz (exact peak at 440.34 Hz) and 1000 Hz (exact peak at 916.41 Hz). This is in line with the results obtained using FFT analysis for the aeroacoustics tutorial and similar to the results of the spectral information of the temporal POD modes 2 through 11 shown in figure 3.15.

Mode Number	Frequency [Hz]	Growth Rate [Hz]	Eigen Value	Normalized Mode Magnitude
467	0	0	1.0000 + 0.0000i	1
163	440.34	-0.51	0.8502 + 0.5252i	0.7596
155	459.19	0.32	0.8384 + 0.5458i	0.7051
145	473.86	0.02	0.8279 + 0.5609i	0.6952
153	449.52	0.45	0.8451 + 0.5356i	0.6721
93	516.8	-0.59	0.7958 + 0.6043i	0.6551
293	916.41	-1.7	0.4062 + 0.9115i	0.6435
89	524.52	-1.68	0.7888 + 0.6111i	0.6361
141	482.09	-0.63	0.8214 + 0.5690i	0.6282
548	57.77	-0.29	0.9970 + 0.0725i	0.4928
472	101.15	-0.39	0.9914 + 0.1267i	0.4356
167	807.51	0.04	0.5279 + 0.8494i	0.3293

Table 3.1: Data of select few modes sorted in descending order of normalized mode magnitude. There are a total of 599 DMD modes (extracted from matrix X_1), each associated with a single frequency and growth rate which as explained,

`

are computed using the eigen value associated with the mode. Mode numbered 467 has a frequency of 0 Hz, implying that it a *background mode* whose spatial distribution would give the mean distribution of pressure over time, identical to POD mode 1.

Further, flow instabilities induced at frequencies associated with modes having a positive growth rate amplify with time while the modes having negative growth rate are stable in time and their amplitudes decay.

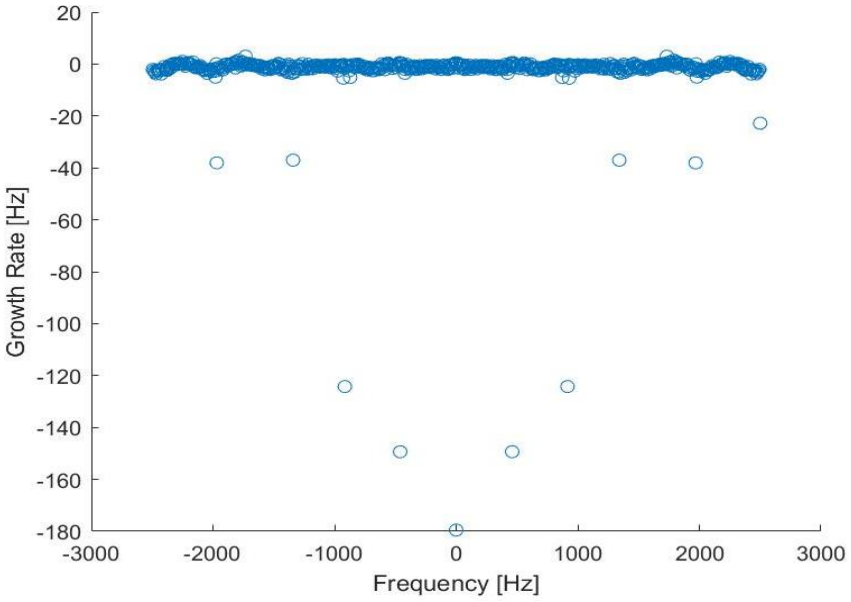


Figure 3.19: 79 out of the total 599 modes have a positive growth rate and so the fluctuations at the frequencies associated with these modes grow with time. The negative frequencies in the figure are due to the fact that eigen values of a real valued data matrix are either real or complex conjugates. Complex conjugates result in frequencies being negative for nearly half the modes.

`

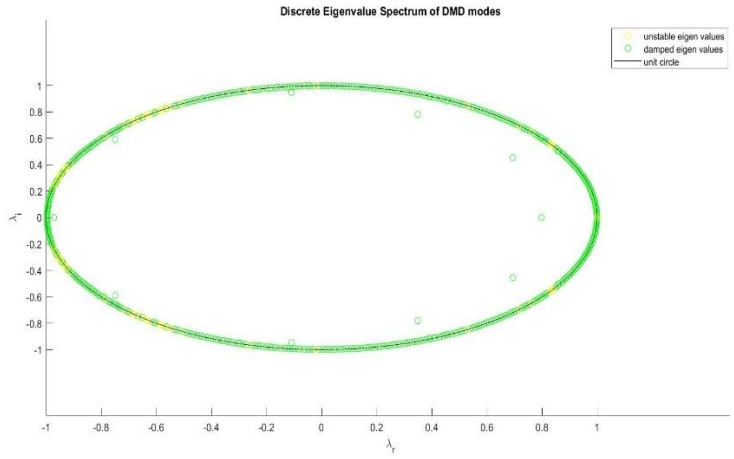


Figure 3.20: Discrete Eigen value spectrum of DMD modes. Eigen values lying outside the unit circle are unstable as the modes associated with them have positive growth rate. Growth rate of the modes is given by the expression
Growth Rate (Hz) = real(log(eigenvalue)) / 2πdt
The real part of the logarithm of the complex valued eigen values is the logarithm of the magnitude of the eigen value. This results in growth rate approaching 0 Hz for modes having an eigen value magnitude close to unity.

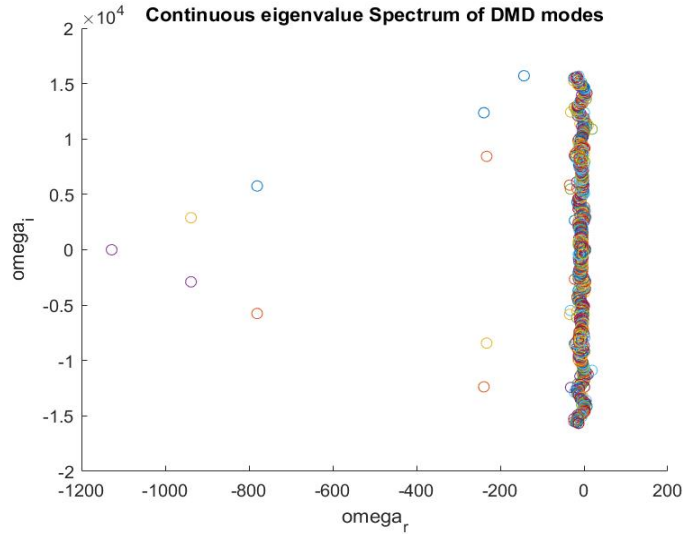


Figure 3.21: Continuous Eigen value spectrum of DMD modes. Ω is given by the expression *logarithm(eigen value)/d*t. Ω_{real} is thus the logarithm of the magnitude of the eigen value divided by the time-step at which the snapshots are written out. Ω_{real} being close to zero implies the eigen value being close to unity which results in growth rate being close to 0 as explained in figure 3.20. Thus, Ω_{real} gives

`

a measure of the growth rate of the modes. The closer the values of Ω_{real} are to 0, the more stable are the modes.

Mode Number	467
Frequency [Hz]	0
Growth Rate [Hz]	0
Eigen Value	1.0000 + 0.0000i
Normalized Mode Magnitude	1

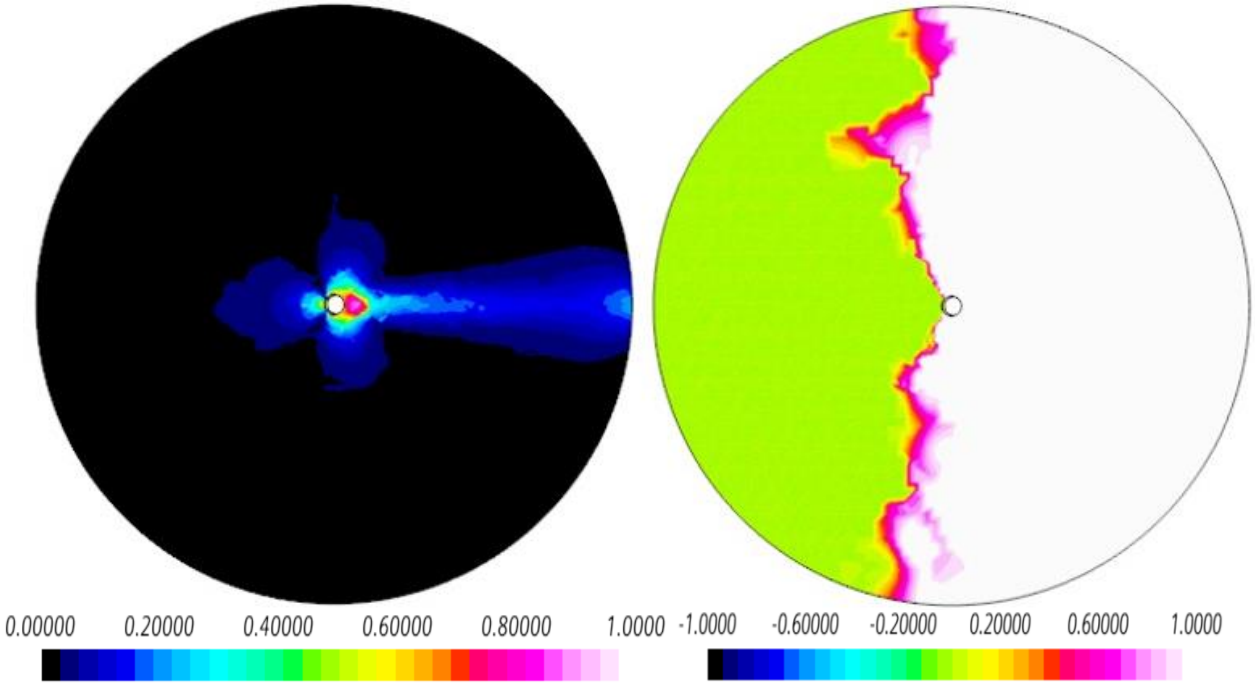


Figure 3.22: DMD mode 467 (left) and relative phase of elements in mode 467 (right).The mode is associated with a frequency of 0 Hz and is a *background mode.* The values of the element provide a measure of the spatial location's participation for the mode. The mode shape is very similar to POD mode 1 which exhibits the mean pressure distribution. The figure on the right shows the phase of the spatial elements relative to the others oscillating at the frequency associated with this mode. The spatial locations which have the same values in phase denote locations where pressure values peak at the same time. Further, spatial coordinates having a positive relative phase contribution represent harmonics that grow while those having a negative relative phase value represent harmonics that attenuate for the wave at the mode frequency. It gives a visual representation of the wave propagation at the mode frequency.

`

Mode Number	155
Frequency [Hz]	459.19
Growth Rate [Hz]	0.32
Eigen Value	0.8384 + 0.5458i
Normalized Mode Magnitude	0.7051

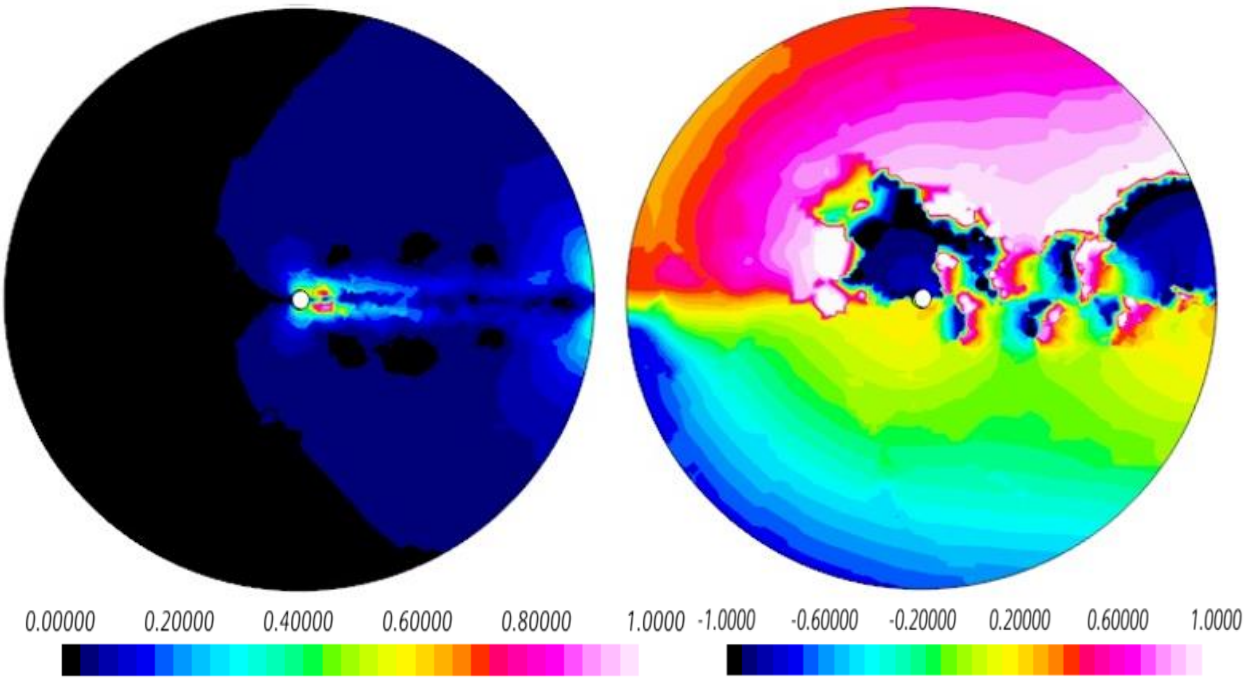


Figure 3.23: DMD mode 155 (left) and relative phase of elements in mode 155 (right). The mode is associated with a frequency of 459.19 Hz close to the peak frequency of 500 Hz. This mode is associated with a positive growth rate and hence flow instabilities associated with this mode grow.

`

3.7 Comparison of POD and DMD:

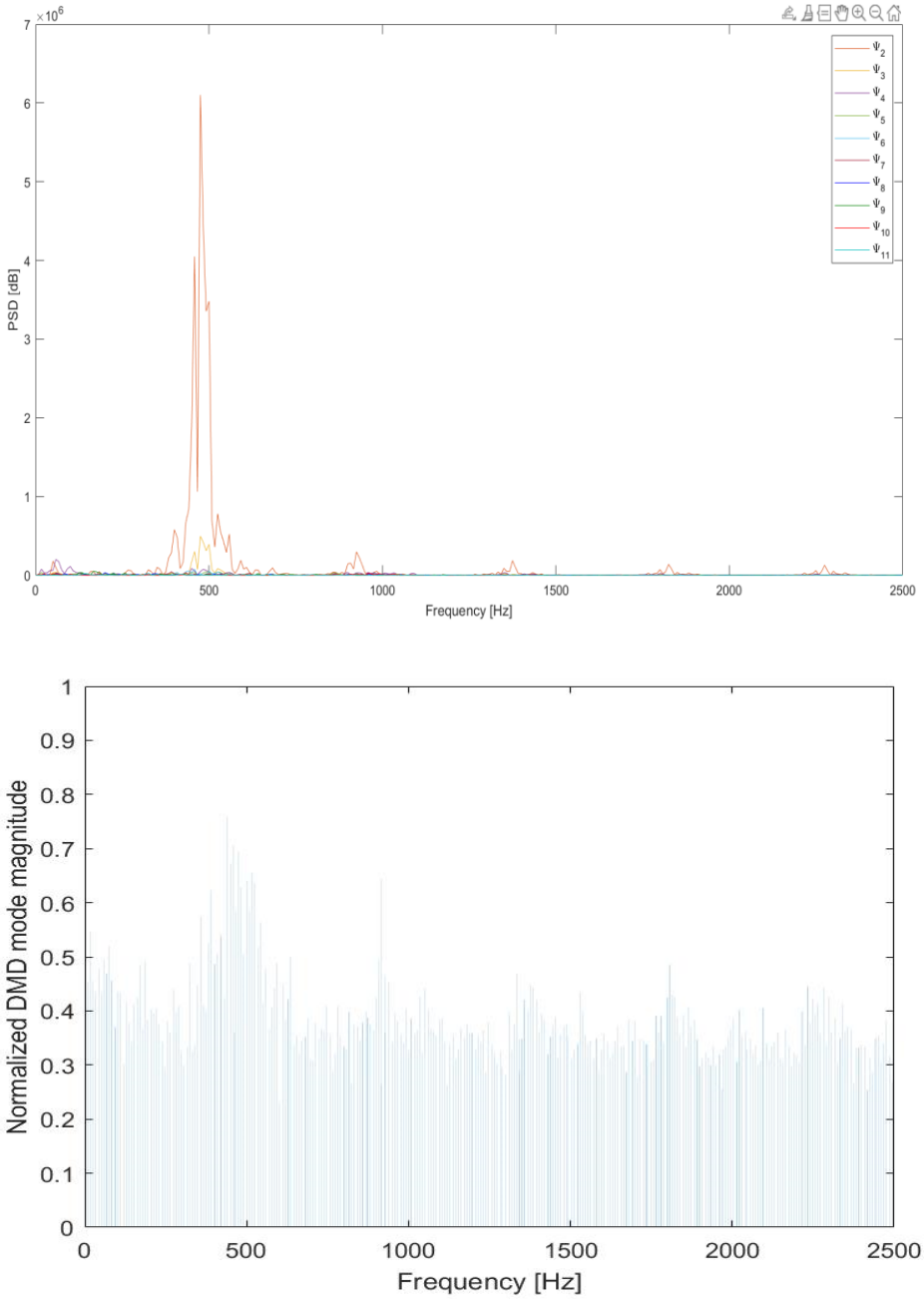


Figure 3.24: Frequency spectral distribution of the temporal POD modes (left) compared to the frequency spectra distribution of DMD modes (right) shows peaks at similar frequencies of close to 500 Hz and 1000 Hz. Both these peak frequencies are also consistent with the results of FFT analysis done. Hence, modal analysis outcome of both modal decomposition techniques are verified.

`

CHAPTER 5: CONCLUSION

Rapid advancements in the field of fluid mechanics have made it possible to simulate increasingly complex flows. This has also led to the production of large volumes of data. Such data is not only difficult to analyze and draw useful inferences from, it also takes up a large amount of storage memory.

Both of these challenges posed by large data are tackled by use of methods that can extract energetically or dynamically important features from less quantity of data, thus also reducing the size of data that is to be stored. This work demonstrates the usefulness of Proper Orthogonal Decomposition (POD) and Dynamic Mode Decomposition (DMD) in their ability to analyze large data sets that are written out from CFD simulations. Additionally, a nearest-neighbor algorithm : *k*-d tree algorithm was implemented to reduce the data before POD and DMD are applied on the reduced data set.

The first case study investigated was that of flow past a bluff body, the bluff body in this study was a cylinder. Pressure data was written out for 600 time-steps at 826,890 grid-points in .csv format files called “snapshots”. These files totaling a size of 35 GB were first converted from .csv format to .bin format files. These binary format files having a total size of 7.4 GB were fed into the *k*-d tree algorithm. 10,000 points were randomly selected and the pressure values at

`

these points for all times were written out in .bin format files. These reduced data files having a total size of 32 MB were used to create the data matrix on which POD and DMD were applied. POD gives the dominant-spatial temporal patterns of pressure in the data in hierarchical order of the “energy percentage”, measured using the singular values obtained through the SVD of the pressure data matrix. The spatial POD modes give the dominant patterns of pressure seen in the data. Using Welch’s power spectral density estimate, the temporal POD modes successfully capture the peak at the vortex shedding frequency of 500 Hz and at 1000 Hz. This is found to be consistent with the FFT analysis of this case study. A shortcoming of the POD method is that the modes are associated with peaks at multiple frequencies. Thus, it is not possible to extract dynamical information such as frequency, phase information, spatial patterns corresponding to individual frequencies using POD. DMD was used to overcome this disadvantage of the POD algorithm. DMD successfully produces mode and phase diagrams corresponding to individual frequencies. A metric called *normalized mode magnitude* was used to compute the relative importance of the DMD modes in a bid to hierarchically order the modes in a similar fashion as seen for the POD modes. On doing this for this case of flow past a cylinder, peak magnitudes were seen near the vortex shedding frequency of 500 Hz and near

`

1000 Hz. Hence, the results of modal analysis were verified for this case study as both POD and DMD capture the peaks at frequencies of 500 Hz and 1000 Hz, consistent with the results found from the FFT analysis done for this aeroacoustic study on a commerical CFD software.

The second case study investigated was that of air flow in a automobile compressor. The motivation for this case study was to find spatial-temporal features at discrete frequencies, especially those in the human auditory range of sound upto 16 kHz. The pressure patterns observed using POD and DMD could be used by engineers to take measures to suppress noise at these frequencies. Pressure data was written out for [redacted] time-steps at [redacted] grid-points. These [redacted] grid-points comprise only the non-rotating region of the compressor geometry. This was done to avoid the complexity of working with a rotating mesh, hence simplifying the study. The compressor consisting of [redacted] blades was run for [redacted] revolutions at an rpm of [redacted] rpm and pressure data was written out for the last [redacted] revolutions. The [redacted] .csv format snapshots totaling a size of [redacted] GB were first converted from .csv format to .bin format files. These binary format files having a total size of [redacted] GB were fed into the *k*-d tree algorithm. The focus of this study was on observing the spatial-temporal patterns of pressure seen in the spriale domain of the compressor because this comprises the areas where

`

most of the dynamics or high gradient phenomena are located. The *k*-d tree algorithm is implemented and the input set of coordinate points are the [redacted] coordinate points of the spirale domain. This input set of coordinates was supplied as a .bin format file containing these coordinates. 10,000 coordinates were selected at random from this sub-gridspace and the pressure values at these points for all times were written out in .bin format files. These reduced data files having a total size of [redacted] MB were used to create the data matrix on which POD and DMD were applied.

Relative phase diagrams of modes at the frequencies of the 5RO (rotating order), 6RO and BPF (Blade Pass Frequency) reveal the nature of wave propagation at these frequencies which can be used to draw dynamical inferences about the harmonics that amplify/attenuate at discrete frequencies. However, the peak frequencies captured by the temporal POD modes are not consistent with the peak frequencies found using normalized mode magnitude. For example, POD mode 3 has a peak at [redacted] Hz. However, as per the normalized mode magnitude, the mode corresponding to a frequency of [redacted] Hz is ranked 296 out of a total [redacted] DMD modes. The inference drawn is that POD is better at hierarchically ordering the modes but does not capture dynamical information, especially at particular frequencies. On the other hand, DMD is better at

`

capturing dynamical information but is not effective in hierarchically ordering modes. Thus, simultaneous use of POD – to determine dominant frequencies and DMD – to obtain mode, relative phase diagrams at those frequencies may produce best results. The second important inference from this work is that all results of POD and DMD are found from data sets vastly smaller than the original data sets. For flow past a cylinder, the size of data analysed is 32 MB instead of the original 35 GB. For air flow in a compressor, inferences were drawn from [redacted] MB of data instead of the original [redacted] GB. Finally, both POD and DMD are purely-data driven, their use does not require knowledge about the governing equations and therefore, can be easily applied to draw inferences from any large data set such as neuroscience, disease control.[3]

5.1 Recommendations for Further Work:

In this project, POD and DMD have been applied to static pressure data written out from CFD simulations for two case studies. The application of the modal decomposition techniques can be extended to other quantities of interest such as velocity. Additionally, these two techniques can serve as a foundation for the implementation of Reduced Order Models (ROMs). The idea behind ROMs is that while POD and DMD are applied to data output *post* CFD simulations, i.e. their application in this project is limited to post-processing section of CFD, ROMs

`

could extend to the solver section of CFD. The literature that has been surveyed, for example [4], demonstrates how POD can be implemented to draw inferences on multiple “training” input data sets. Using the outputs obtained drawn for the training sets, a ROM would be able to predict output for a “test” input data set.

`

REFERENCES

1. Aero and Vibroacoustics of Automotive Turbochargers, Hung Nguyen-Schafer, Springer, 2013

2. Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control, Steven Brunton, J. Nathan Kutz

3. Dynamic Mode Decomposition: Data-driven Modeling of Complex Systems" by Kutz, S. Brunton, B. Brunton, Proctor

4. Analysis and Compression of Large CFD Data Sets Using Proper Orthogonal Decomposition, T.J. Blanc, 2014

Some other literature that was read to build an understanding of modal decomposition is listed as follows:

1. Modal Analysis of Fluid Flows: An Overview, Taira Et al, AIAA Journal, 2017

2. Modeling of flow generated sound in a constricted duct at low Mach number flow, Gamage, University of Central Florida, 2017

3. Acoustic characteristics of a ported shroud turbocompressor operating at design conditions, Sharma Et al, 2018

4. "Dynamic Mode Decomposition: Data-driven Modeling of Complex Systems" by Kutz, S. Brunton, B. Brunton, Proctor

5. Analysis of High Fidelity Turbomachinery CFD using Proper Orthogonal Decomposition, Spencer, March 2016

6. Dynamic mode decomposition of numerical and experimental data, Peter Schmid , July 2010

7. Dynamic mode decomposition of controlled H- and K-type transitions, Sayadi et.al 2013

`

8. Trend analysis of Indian Stock Market using Dynamic Mode Decomposition, Kuttichira et.al, July 2017

9. Dynamic Mode Decomposition of Fluid Flow Data, Anupindi, December 2010

10. Data-Driven Modeling & Scientific Computation: Methods for Complex Systems & Big Data, J. Nathan Kutz

`

APPENDIX

POD modes 5 through 10 for flow past a cylinder:

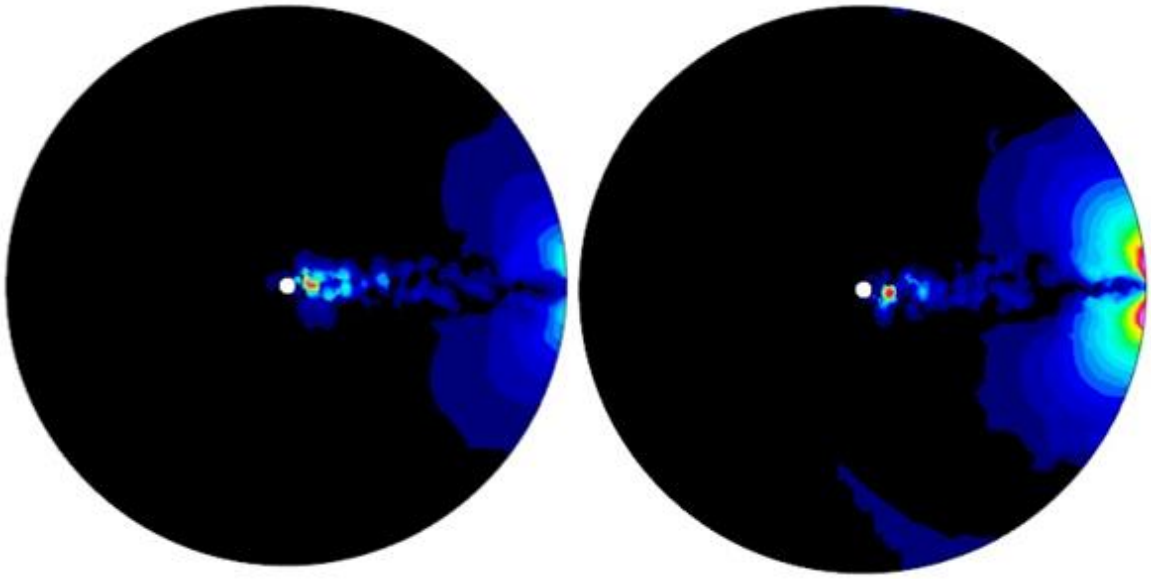


Figure A.1: Spatial POD modes 5(left) and 6 (right), showing the relative contributions of spatial coordinates.

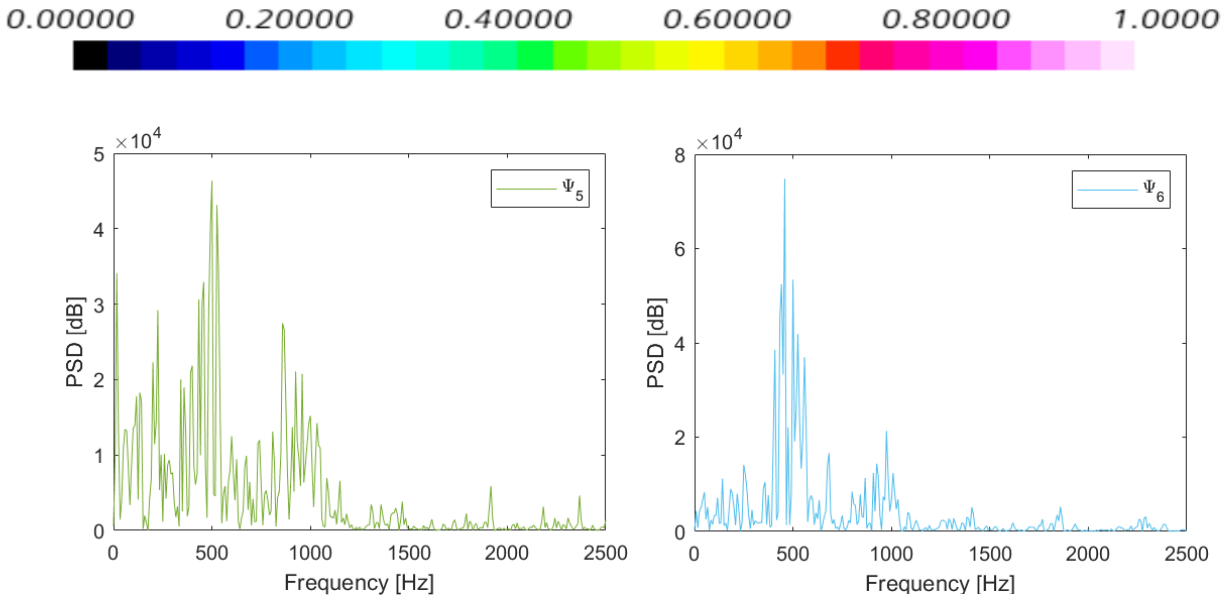


Figure A.2: Temporal POD modes 5 (left) and 6 (right). Mode 5 has peaks at 16.67 Hz, 60 Hz, 120 Hz, 200 Hz, 400 Hz, 525 Hz, 858 Hz, 960 Hz, mode 6 has peaks at 408 Hz, 460 Hz, 500 Hz, 560 Hz, 683.33 Hz, 975 Hz.

`

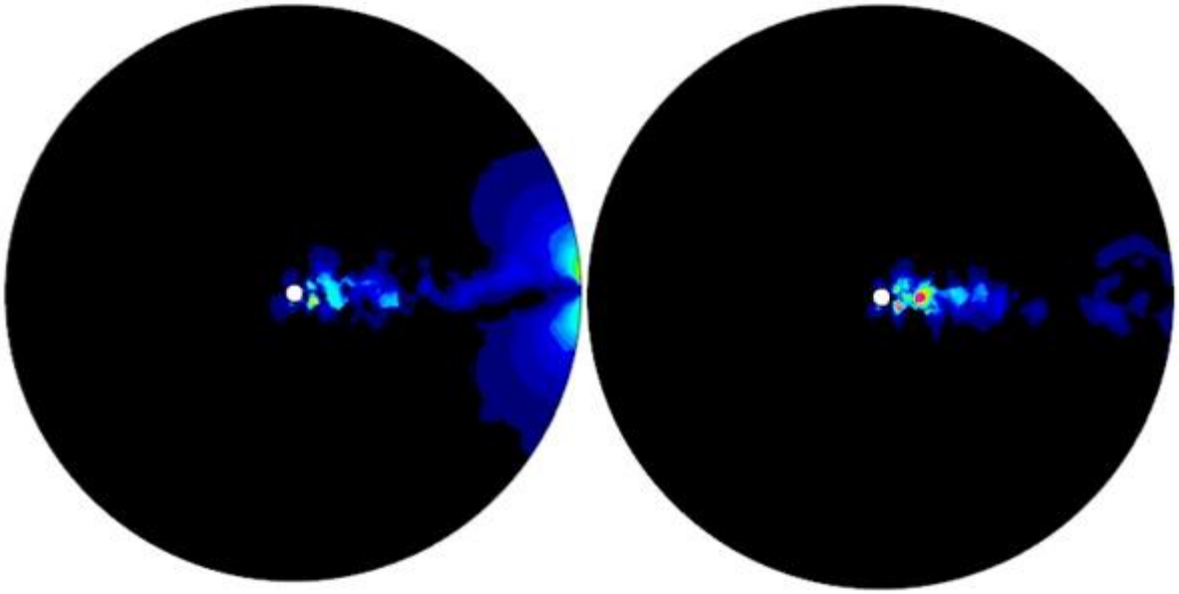


Figure A.3: Spatial POD modes 7(left) and 8 (right), showing the relative contributions of spatial coordinates.

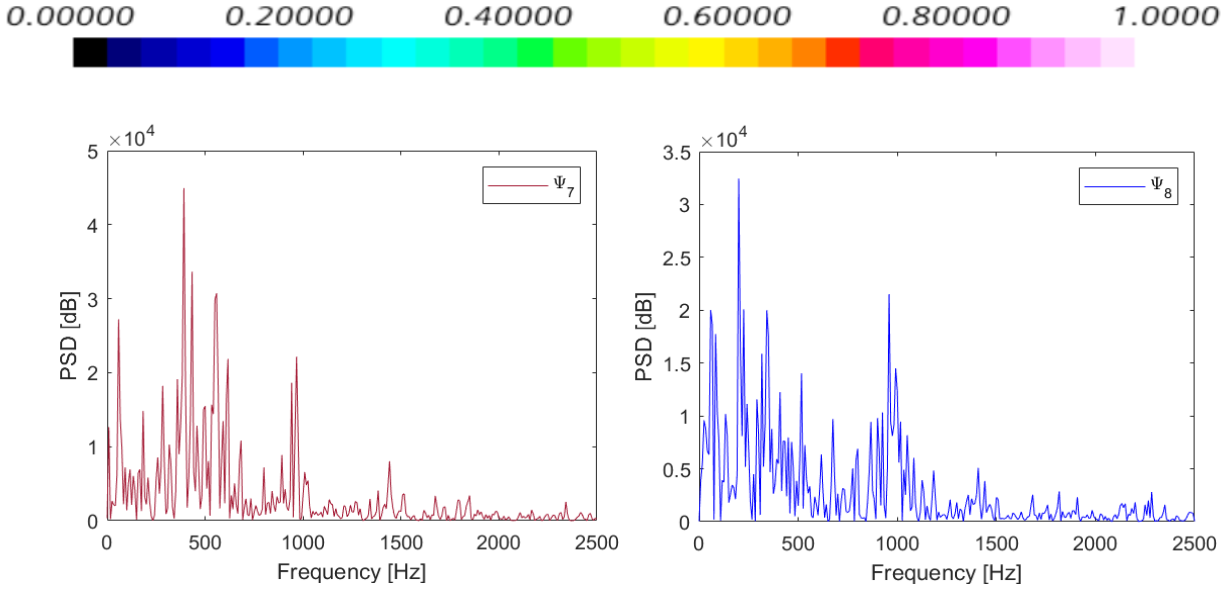


Figure A.4: Temporal POD modes 7 (left) and 8 (right). Mode 7 has peaks at 60 Hz, 400 Hz, 433 Hz, 560 Hz, 616 Hz, 977 Hz mode 8 has peaks at 60 Hz, 200 Hz, 225 Hz, 342 Hz, 520 Hz, 960 Hz.

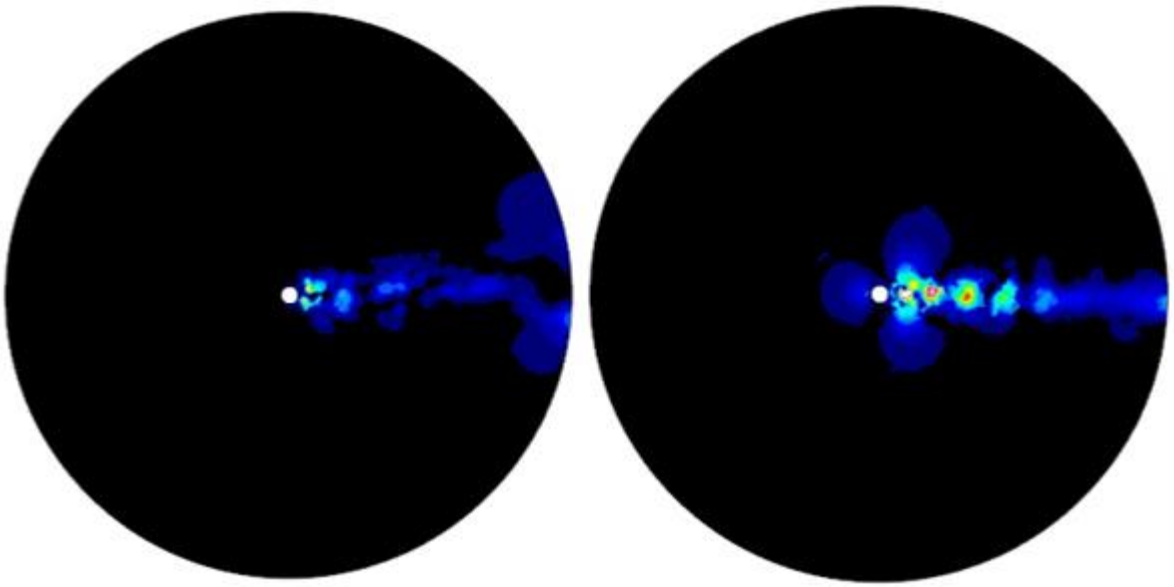


Figure A.5: Spatial POD modes 9 (left) and 10 (right), showing the relative contributions of spatial coordinates.

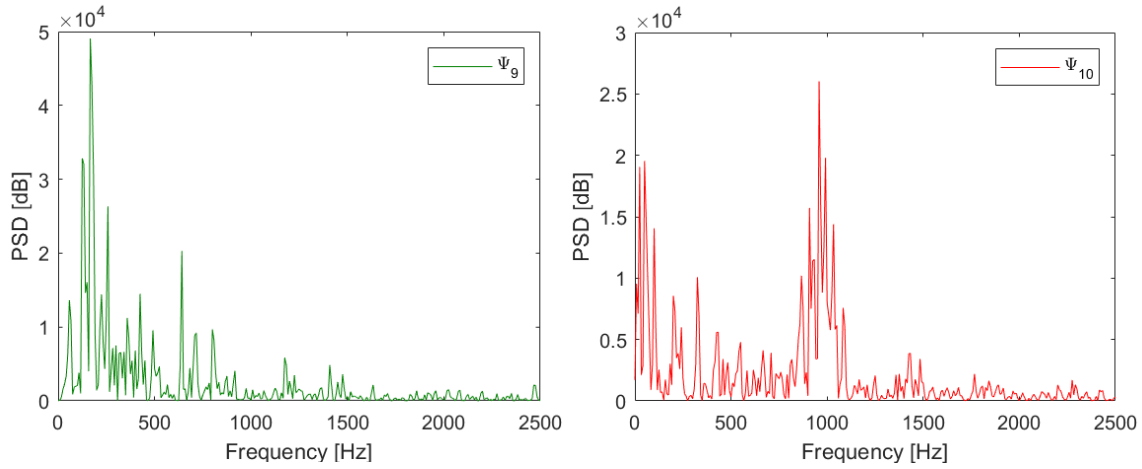


Figure A.6: Temporal POD modes 9 (left) and 10 (right). Mode 9 has a peak at 125 Hz, 166.67 Hz, 260 Hz, 642 Hz mode 10 has peaks at 25 Hz, 50 Hz, 100 Hz, 325 Hz, 867 Hz, 991 Hz, 1033 Hz.

`

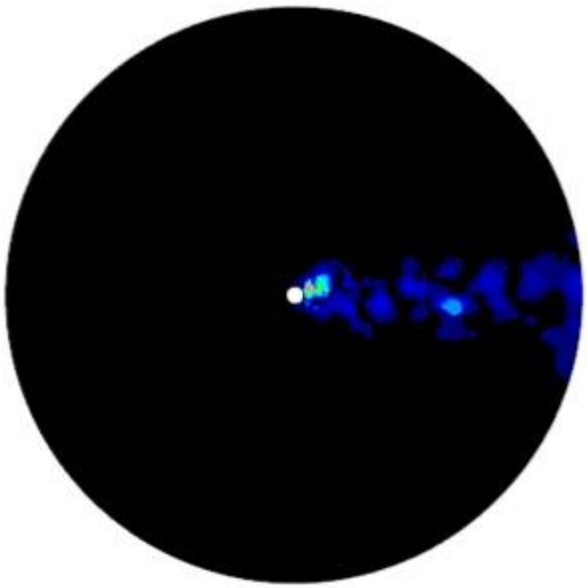


Figure A.7: Spatial POD mode 11, showing the relative contributions of spatial coordinates.

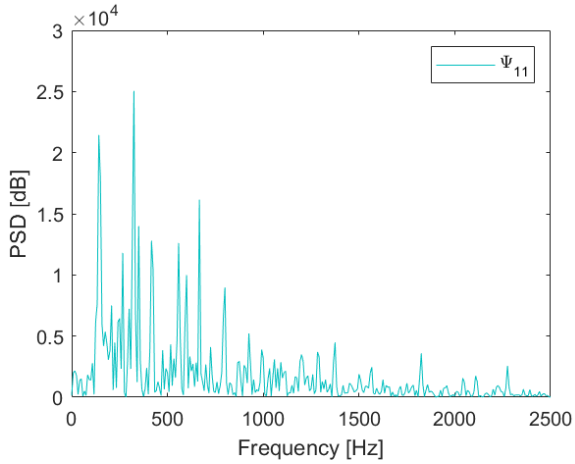


Figure A.8: Temporal POD mode 11. Mode 11 has peaks at 142 Hz, 325 Hz, 420 Hz, 560 Hz, 666 Hz.