

# **Proper Orthogonal Decomposition and Dynamic Mode Decomposition of Ventilated Jet Flow**

*A Project Report*

*by*

**SIDDHARTH DEY**

**ME18B075**

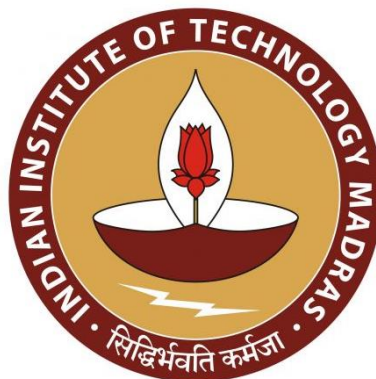
*Submitted in partial fulfillment of the requirements for the award of*

*the degree of*

**BACHELOR OF TECHNOLOGY**

*in*

**MECHANICAL ENGINEERING**



**Department of Mechanical Engineering  
INDIAN INSTITUTE OF TECHNOLOGY MADRAS  
JUNE 2022**



## PROJECT CERTIFICATE

This is to certify that the thesis entitled “**Proper Orthogonal Decomposition and Dynamic Mode Decomposition of Ventilated Jet Flow**” submitted by **Siddharth Dey (ME18B075)** to the Indian Institute of Technology, Madras for the award of the degree of **Bachelor of Technology in Mechanical Engineering**, is a bona fide record of research work carried out by him under my supervision. The contents of this report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

**Dr. Kameswararao Anupindi**  
Associate Professor  
Department of Mechanical Engineering  
Indian Institute of Technology Madras  
Chennai – 600 036.

**Prof. A. Seshadri Sekhar**  
Professor and Head of Department  
Department of Mechanical Engineering  
Indian Institute of Technology Madras  
Chennai – 600 036.

Place: Chennai

Date: 10<sup>th</sup> June 2022

## ACKNOWLEDGEMENTS

I would like to take this opportunity to thank **Prof. Kameswararao Anupindi** for his assistance and advice throughout the study. He was a constant source of motivation for me. I was able to complete this task because of his skilled assistance and keen eye for detail.

I'd also want to thank **Mr. Aritra Roy Choudhary** for his assistance during the project. I greatly benefited from the knowledge I got from him in performing the simulations. This task would not have been accomplished without his persistent assistance, patience, and enthusiasm to assist during difficult moments.

Finally, I'd want to thank my family and friends for their unwavering support during this effort.

## ABSTRACT

*Keywords:* Proper Orthogonal Decomposition (POD); Dynamic Mode Decomposition (DMD); Jet Flow; Inlet Turbulence Modelling

Model order reduction (MOR) is a class of algorithms used in numerical simulations to reduce the computational complexity of mathematical models. It is a common strategy to decompose the physical field under study (such as pressure, temperature, velocity, and so on) into orthogonal modes to study the coherent structures in the flow. The two methods used in this study for modal decomposition are the Proper Orthogonal Decomposition (POD) and the Dynamic Mode Decomposition (DMD). The different modes obtained using these methods show various characteristics of the flow field. POD decomposes the modes based on their energy contribution to the flow whereas the DMD method focuses on the frequency and the growth/decay stability of the modes. These modes can be effective in identifying the dominant coherent structures and vortices which are otherwise not visible in the original flow field. The algorithm is verified by testing it on synthetically generated data and comparing the spatial modes and temporal information obtained from the decomposed modes.

In this study, we apply POD and DMD on a ventilated jet flow. Ventilated offset jet is the combination of an offset jet issued parallel to a wall and a ventilated jet, which due to suction pressure arising from the offset jet, comes out of the slot between the offset jet inlet and the wall. To understand the interaction between the offset and the ventilated jet, the POD and DMD modes of three cases with velocity ratios (VR) of 0.0, 0.10, and 0.18 are compared.

Finally, the application of POD for the generation of turbulent inflow boundary conditions is explored. First, an LES-based channel flow simulation is performed as the auxiliary simulation. The POD modes are extracted and a low-order dynamical model is derived to provide the temporal evolution of the most energetic structures. The incoherent motion is modeled by using time series based on a random number generator. This study helps us understand the use of the reduced ordered model for inlet turbulence modeling.

## **TABLE OF CONTENTS**

ABSTRACT	ii
TABLE OF CONTENTS	iii
LIST OF FIGURES	v
NOMENCLATURE	vii
<b>CHAPTER 1</b>	<b>1</b>
<b>INTRODUCTION</b>	<b>1</b>
1.1 PROPER ORTHOGONAL DECOMPOSTION (POD)	1
1.1.1 USING COVARIANCE MATRIX (DIRECT METHOD)	1
1.1.2 USING SINGULAR VALUE DECOMPOSITION (SVD)	2
1.2 DYNAMIC MODE DECOMPOSTION (DMD)	3
1.2.1 DMD ALGORITHM	4
1.3 OBJECTIVES OF THE WORK	7
1.4 SCOPE OF THE THESIS	7
<b>CHAPTER 2</b>	<b>8</b>
<b>VALIDATION CASES</b>	<b>8</b>
2.1 POD VALIDATION CASE: POISEUILLE FLOW	8
2.1.1 POD ANALYSIS	8
2.2 DMD VALIDATION CASE	10
<b>CHAPTER 3</b>	<b>12</b>
<b>POD AND DMD ANALYSIS OF JET FLOW</b>	<b>12</b>
3.1 VENTILATED JET FLOW	12
3.2 SENSITIVITY ANALYSIS	12
3.3 POD ANALYSIS	15
3.4 DMD ANALYSIS	16
3.4.1 DMD MODES	17
<b>CHAPTER 4</b>	<b>21</b>
<b>CONCLUSION AND FUTURE WORK</b>	<b>21</b>

<b>APPENDIX</b>	<b>22</b>
A.1 TURBULENT INLET BOUNDARY CONDITION GENERATION	22
A.1.1. INTRODUCTION	22
A.1.2. LOW ORDER MODEL IDENTIFICATION	23
A.1.3. LODS TEST CASE: LORENTZ SYSTEM	24
A.1.4. RANDOM UNCOHERENT MOTION	26
A.1.5. AUXILIARY CHANNEL FLOW SIMULATION	27
A.1.6. CHANNEL FLOW POD	28
A.1.7 PROBLEMS AND FUTURE WORK	29
A.2 LODS – LORENTZ SYSTEM TEST CASE – CODE	30
<b>REFERENCES</b>	<b>35</b>

## LIST OF FIGURES

**Fig. 2.1** Schematic of Poiseuille Flow

**Fig. 2.2** Normalized singular values of the Poiseuille Flow

**Fig. 2.3** The first three POD modes of Poiseuille Flow

**Fig. 2.4** The time dynamics of the first three POD modes of Poiseuille Flow

**Fig. 2.5** Reconstruction at  $t = 3$  sec using POD considering 1 mode (U1), 2 modes (U2) and 3 modes (U3). (c): Reconstruction at  $t = 7$  sec using POD considering 1 mode (U1), 2 modes (U2) and 3 modes (U3). UR is the analytical solution

**Fig. 2.6** Original Spatiotemporal generated data, (a)  $h_1(x, t)$  (b)  $h_2(x, t)$  (c)  $h(x, t)$

**Fig. 2.7** (a) Singular Values, (b) Ritz values on the complex plane

**Fig. 2.8** (a) First DMD mode and true spatial value of first mode, (b) Second DMD mode and true spatial value of second mode

**Fig. 2.9** (a) Reconstruction on mode 1 (b) Reconstruction on mode 2 (c) Reconstruction on the first two modes

**Fig. 3.1** (a) Normalized eigenvalues of the 9 test cases (b) The count of the eigenvalues with corresponding percentage difference with the first test case

**Fig. 3.2** (a) Temporal coefficient of the first mode for the 9 test cases, (b) Temporal coefficient of the first mode for the 9 test cases

**Fig. 3.3** Ritz Values plotted on the complex plane along a unit circle for the 9 test cases

**Fig. 3.4** POD normalized eigenvalues for (a)  $vr = 0$  (b)  $vr = 10$  (c)  $vr = 18$

**Fig. 3.5** PSD plot of (a)  $vr = 0$  (b)  $vr = 10$  (c)  $vr = 18$

**Fig. 3.6** The logarithmic plot of the Ritz values defined as  $\mu = \log(\lambda) / \Delta_t$  for (a)  $vr = 0$  (b)  $vr = 10$  (c)  $vr = 18$  (The blue circles correspond to all the modes and the ones shown as orange triangles are the selected modes based on the selection criteria)

**Fig. 3.7** The amplitude of the DMD modes for (a)  $vr = 0$  (b)  $vr = 10$  (c)  $vr = 18$

**Fig. 3.7** The first 5 DMD modes ( $vr = 0\%$  (top),  $vr = 10\%$  (left) and  $vr = 18\%$  (right))

**Fig. 5.1** Phase plot of the  $x$ ,  $y$  and  $z$  given by Eqn (42)



**Fig. 5.2** The time dynamics of  $x$ ,  $y$  and  $z$  along with the 100 random sample points (shown as orange circles)

**Fig. 5.3** (a) Reconstructed LODS phase plot; (b) Reconstructed LODS time dynamics of  $x$ ,  $y$  and  $z$

**Fig. 5.4** (a) Geometry of Channel Flow simulation (b) Meshing of the computational domain

**Fig. 5.5** (a) The sectional view of the channel flow (b) Contour plot

**Fig. 5.6** POD normalized eigenvalues of the Channel Flow

**Fig. 5.7** The first 4 POD modes of the Channel Flow

## **NOMENCLATURE**

POD – Proper Orthogonal Decomposition

DMD – Dynamic Mode Decomposition

SVD – Singular Value Decomposition

SpDMD - Sparsity Promoting Dynamic Mode Decomposition

PSD – Power Spectral Density

FFT – Fast Fourier Transform

LODS – Low Order Dynamical System

ODE – Ordinary Differential Equation

DNS – Direct Numerical Solution

LES – Larger Eddy Simulation

# CHAPTER 1

## INTRODUCTION

### 1.1. PROPER ORTHOGONAL DECOMPOSITION (POD)

The POD algorithm divides the input vector field into a series of deterministic spatial functions (POD modes) with corresponding time coefficients. The physical field of interest in this work is the fluctuating component of velocity. It is decomposed as shown below:

$$\mathbf{u}'(\mathbf{x}, t) = \sum_{k=1}^{\infty} a_k(t) \Phi_k(\mathbf{x}) \quad (1)$$

where,  $\Phi_k(\mathbf{x})$  are the spatial modes and  $a_k(t)$  are the temporal coefficients. The spatial and temporal evolution can be obtained by either computing the covariance matrix, followed by computation of its eigenvalues and eigenvectors to derive the POD modes, or by directly performing the Singular Value Decomposition (SVD) on the Snapshot matrix.

#### 1.1.1 USING COVARIANCE MATRIX (DIRECT METHOD)

- The velocity data is arranged as a 2D matrix, where each row contains the value of different components of the velocity at varying spatial coordinates, for a given timestamp. Different rows contain velocity data of the same set of spatial nodes at different timestamps. It results in an  $m \times n$  matrix  $\mathbf{V}$ ,  $m$  being the number of timestamps and  $n = n_x \times n_y$ , where  $n_x$  and  $n_y$  are the number of spatial nodes in the x and y axes respectively. The time-averaged velocity field is subtracted from the individual vector fields  $\mathbf{v}(x_i, y_i, t_j)$  to obtain the fluctuating component  $\mathbf{v}'(x_i, y_i, t_j)$ .

$$\mathbf{V} = \begin{pmatrix} v_{11} & \dots & v_{1n} \\ v_{21} & \dots & v_{2n} \\ \vdots & & \vdots \\ \vdots & & \vdots \\ v_{m1} & \dots & v_{mn} \end{pmatrix} = \begin{pmatrix} v'(x_1, y_1, t_1) & \dots & v'(x_{N_x}, y_{N_y}, t_1) \\ v'(x_1, y_1, t_2) & \dots & v'(x_{N_x}, y_{N_y}, t_2) \\ \vdots & & \vdots \\ \vdots & & \vdots \\ v'(x_1, y_1, t_m) & \dots & v'(x_{N_x}, y_{N_y}, t_m) \end{pmatrix} \quad (2)$$

- The covariance matrix  $\mathbf{M}$  is calculated as follows [1]:

$$\mathbf{M} = \left( \frac{1}{m-1} \right) \mathbf{V}^T \mathbf{V} \quad (3)$$

which results in an  $n \times n$  matrix. (In the ‘Snapshot method’, the covariance matrix is computed as  $\mathbf{M} = \left( \frac{1}{m-1} \right) \mathbf{V} \mathbf{V}^T$  which results in an  $m \times m$  matrix; depending on the

values of  $m$  and  $n$ , the method which results in a smaller covariance matrix is used to reduce computational costs)

- The eigenvalues  $\lambda_1 \dots \lambda_n$  and the orthogonal eigenvectors  $\Phi$  of the covariance matrix  $\mathbf{M}$  is calculated. The  $n$  eigenvectors are the proper orthogonal spatial modes of the dataset. The modes are ordered as per the variance in their direction. The normalized eigenvalues are representative of the contribution of the mode to the Total Kinetic Energy (TKE). Hence, the modes are ranked on the basis of their energy contribution to the flow field.
- The temporal coefficients corresponding to the POD modes are obtained by projecting the original dataset onto the orthogonal modes as follows:

$$\mathbf{A} = \mathbf{V}\Phi \quad (4)$$

The columns of  $\mathbf{A}$  are the time coefficients associated with the modes.

- The data can be reconstructed on a specific mode by multiplying the orthogonal mode and its time dynamics as follows:

$$\mathbf{V}^k = \begin{pmatrix} a_{1k} \\ a_{2k} \\ \vdots \\ a_{mk} \end{pmatrix} (\Phi_{1k} \dots \Phi_{nk}) \quad (5)$$

$$\mathbf{V} = \sum_{k=1}^n \mathbf{V}^k \quad (6)$$

The original velocity fluctuations have been decomposed into a sum of  $n$  contributions from  $n$  proper orthogonal modes.

### 1.1.1 USING SINGULAR VALUE DECOMPOSITION (SVD)

Instead of manually performing the eigenvalue analysis of the covariance matrix, the modes can directly be obtained by SVD.

- The SVD decomposition is performed on the matrix  $\mathbf{V}$

$$\mathbf{V} = \mathbf{U}\Sigma\mathbf{W}^T \quad (7)$$

where  $\mathbf{U}$  is an  $m \times m$  orthogonal matrix,  $\Sigma$  is an  $m \times n$  rectangular diagonal matrix and  $\mathbf{W}$  is an  $n \times n$  orthogonal matrix.

- $\mathbf{W}$  is the same as  $\Phi$ , whose columns are the orthogonal POD modes

- The non-zero diagonal elements of  $\Sigma$  are known as the singular values, whose squared values correspond to the eigenvalues of the correlation matrix in the direct method
- The time dynamics can be obtained in the same way as the previous method

Though SVD is more economical in terms of lines of code, the Snapshot POD is much faster as the SVD algorithm still needs to compute the high dimensional spatial matrix  $\mathbf{V}^T \mathbf{V}$  to calculate its eigenvectors  $\mathbf{W}$ .

## 1.2. DYNAMIC MODE DECOMPOSITION (DMD)

DMD, like POD, is a spatial dimensionality reduction approach. It was first established in the fluid mechanics community to investigate experiments and simulations aimed at extracting a small number of coupled spatial-temporal modes from very high-dimensional dynamic data. It's also related to Koopman spectrum analysis, which has applications in nonlinear system dynamics characterization.

DMD involves the eigenvalue decomposition of a linear operator. Let  $\mathbf{X}$  be a snapshot matrix containing the temporal data of a single spatial coordinate along each row.  $\mathbf{X}$  is divided into two sets of data,  $\mathbf{X}_1$  and  $\mathbf{X}_2$ , with  $\mathbf{X}_1$  containing all of  $\mathbf{X}$ 's columns from the first to the penultimate column, and  $\mathbf{X}_2$  containing all of  $\mathbf{X}$ 's columns from the second to the last column.

$$\begin{aligned}\mathbf{X}_1 &= [\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_{n-1}] \\ \mathbf{X}_2 &= [\mathbf{x}_2 \ \mathbf{x}_3 \ \dots \ \mathbf{x}_n]\end{aligned}\tag{8}$$

where  $\mathbf{x}_1, \mathbf{x}_2 \dots \mathbf{x}_n$  are the columns of  $\mathbf{X}$ . A best fit linear operator is assumed which can compute  $\mathbf{X}_2$  from  $\mathbf{X}_1$  as follows:

$$\mathbf{X}_2 = \mathbf{A} \mathbf{X}_1\tag{9}$$

Based on the values of the previous timestamp, the linear operator  $\mathbf{A}$  determines the velocity of all the spatial nodes at the following timestamp. In a way, it carries information of the derivative of the snapshot data. As a result, DMD performs an eigenvalue decomposition analysis of  $\mathbf{A}$  to gain insight into its dynamics. The Moore-Penrose pseudo-inverse can be used to calculate the matrix  $\mathbf{A}$  as follows:

$$\mathbf{A} = \mathbf{X}_2 \mathbf{X}_1^{-1}\tag{10}$$

It is not recommended to directly take the pseudo inverse as it would cause  $\mathbf{A}$  to become a high dimensional  $n \times n$  matrix, where  $n$  is the number of spatial nodes, making it computationally expensive to directly perform the eigenvalue analysis on it.

### 1.2.1 DMD ALGORITHM

The steps [2] involved in the DMD analysis are:

- SVD decomposition is applied on the first snapshot matrix  $\mathbf{X}_1$  as shown below:

$$\mathbf{X}_1 = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^* \quad (11)$$

where ‘\*’ stands for conjugate transpose of the matrix.

- **Low Rank DMD:** It could be possible to capture the essential dynamics of the data with very few DMD modes when the high-dimensional dynamics of the data have some underlying low-dimensional structure. To obtain a truncated DMD with  $r$  modes, a truncated SVD basis with only the  $r$  largest singular values can be used instead. Let  $\mathbf{U}_r$  be the first  $r$  columns of  $\mathbf{U}$ ,  $\mathbf{\Sigma}_r$  be  $\mathbf{\Sigma}$ 's upper left  $r \times r$  matrix, and  $\mathbf{V}_r$  be  $\mathbf{V}$ 's first  $r$  columns.

$$\mathbf{U}_r = \mathbf{U}(:, 1:r); \mathbf{\Sigma}_r = \mathbf{\Sigma}(1:r, 1:r); \mathbf{V}_r = \mathbf{V}(:, 1:r); \quad (12)$$

The indexing in Eqn (12) has been presented in the MATLAB notation. This reduced SVD base can be used in the upcoming steps. For the sake of simplicity, the subscript  $r$  is omitted for the truncated matrices.

- Using a basis transformation, compute the  $r \times r$  projection of the full matrix  $\mathbf{A}$  onto the POD modes (which is the truncated  $\mathbf{U}$  matrix):

$$\tilde{\mathbf{A}} = \mathbf{U}^* \mathbf{A} \mathbf{U} = \mathbf{U}^* \mathbf{X}_2 \mathbf{X}_1^{-1} \mathbf{U} = \mathbf{U}^* \mathbf{X}_2 \mathbf{V} \mathbf{\Sigma}^{-1} \quad (13)$$

As mentioned earlier, it might become computationally expensive to calculate the entire  $\mathbf{A}$  matrix. Hence,  $\tilde{\mathbf{A}}$  provides a more robust calculation of the low dimensional representation of  $\mathbf{A}$ .

- The eigenvectors  $\mathbf{W}$  and the eigenvalues  $\lambda_k$  of the low-dimensional  $\tilde{\mathbf{A}}$  is calculated

$$\tilde{\mathbf{A}} \mathbf{W} = \mathbf{W} \mathbf{\Lambda} \quad (14)$$

where  $\mathbf{\Lambda}$  is a diagonal matrix containing the corresponding eigenvalues  $\lambda_k$ . These eigenvalues are also known as the 'Ritz values' and their relative location on a unit circle drawn on the complex plane determines the stability of the DMD modes. The modes with Ritz values lying inside the circle are decaying while the ones outside the circle are growing. The mode lying on the circle are the stable modes. It is usually the stable modes which are of interest in the DMD analysis

- The eigenvalues of  $\mathbf{A}$  are the same as  $\tilde{\mathbf{A}}$ , which are the elements of the diagonal matrix  $\mathbf{\Lambda}$  and the eigenvectors of  $\mathbf{A}$ , which are given by the columns of  $\Phi$ , is related to  $\tilde{\mathbf{A}}$ 's eigenvectors  $\mathbf{W}$  as shown below:

$$\Phi = \mathbf{X}'\mathbf{V}\Sigma^{-1}\mathbf{W} \quad (15)$$

The columns of  $\Phi$  are known as the exact DMD modes as these are the exact eigenvectors of the matrix  $\mathbf{A}$ . In projected DMD modes, the eigenvectors  $\mathbf{W}$  are projected onto the subspace  $\mathbf{U}$  as:

$$\Phi = \mathbf{U}\mathbf{W} \quad (16)$$

- The data can be reconstructed as follows:

$$\omega_k = \ln(\lambda_k) / \Delta t \quad (17)$$

$$\mathbf{x}(t) = \sum_{k=1}^r \Phi_k \exp(\omega_k t) \mathbf{b}_k = \Phi \exp(\mathbf{\Omega}t) \mathbf{b} \quad (18)$$

where  $\Delta t$  is the time step used in the simulation and  $\mathbf{\Omega}$  is a diagonal matrix containing the entries  $\omega_k$  along the diagonal.

- $\mathbf{b}$  is the initial coefficient vector. At time  $t=0$ , Eqn (18) becomes:

$$\mathbf{x}(0) = \mathbf{x}_1 = \Phi \mathbf{b} \quad (19)$$

$$\mathbf{b} = \Phi^{-1} \mathbf{x}_1 \quad (20)$$

As  $\Phi$  is not always a square matrix, the pseudo inverse of  $\Phi$  is used to calculate  $\mathbf{b}$ . The entries of  $\mathbf{b}$  are the initial amplitudes of the time dynamics of the corresponding DMD modes and its magnitude is often used to rank the modes.

- The time dynamics of the DMD modes is obtained by:

$$\text{Time Dynamics} = \begin{bmatrix} 1 & \lambda_1 & \lambda_1^2 & \dots & \lambda_1^{m-2} \\ 1 & \lambda_2 & \lambda_2^2 & \dots & \lambda_2^{m-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \end{bmatrix} \quad (21)$$

- The final reconstructed matrix can be viewed as:

$$\mathbf{x}(t) = [\Phi_1 \Phi_2 \dots] \begin{bmatrix} \mathbf{b}_1 & \mathbf{0} & \dots \\ \mathbf{0} & \mathbf{b}_2 & \dots \\ \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} 1 & \lambda_1 & \lambda_1^2 & \dots & \lambda_1^{m-2} \\ 1 & \lambda_2 & \lambda_2^2 & \dots & \lambda_2^{m-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \end{bmatrix} \quad (22)$$

where  $\Phi_1, \Phi_2 \dots$  are the DMD modes arranged as column vectors. The first matrix contains the spatial information, the second matrix is a diagonal matrix with the initial amplitude of each mode as its entries, and the third matrix contains the time dynamics of each mode arranged along the rows.

- Another approach to reconstruct the data [3] at a particular time stamp is:

$$\hat{\mathbf{X}} = \Phi \Delta^t \mathbf{z}_0 \quad (21)$$

where  $t$  is the time index at which the data needs to be reconstructed

- The frequency of oscillation is determined by the phase of each eigenvalue, which is given by the expression:

$$f_i = \frac{\text{imag}\left(\ln\left(\frac{\lambda_i}{\Delta_t}\right)\right)}{2\pi} \quad (22)$$

This gives the frequency of the DMD modes' oscillations in cycles per second. It is often customary to plot the logarithmic mapping of the Ritz values given by:

$$\mu_i = \frac{\ln(\lambda_i)}{\Delta_t} \quad (23)$$

The decaying DMD modes are on the left half of this plot, the growing modes are on the right side, and the stable modes lie on the imaginary axis.

- For selection/ranking of modes based on their energy, the magnitude of the DMD modes is appropriately scaled. The DMD modes (columns of  $\Phi$ ) have unit norm. Hence, the projected  $\tilde{\mathbf{A}}$  matrix is scaled by the singular values matrix  $\Sigma$  to obtain  $\hat{\mathbf{A}}$ :

$$\hat{\mathbf{A}} = \Sigma^{-1/2} \tilde{\mathbf{A}} \Sigma^{1/2} \quad (24)$$

The eigenvectors and the diagonal matrix containing the eigenvalues of  $\hat{\mathbf{A}}$  is calculated:

$$\hat{\mathbf{A}} \hat{\mathbf{W}} = \hat{\mathbf{W}} \mathbf{\Lambda} \quad (25)$$

The scaled eigenvectors  $\mathbf{W}$  to be used for calculating the modes  $\Phi$  is obtained by:

$$\mathbf{W} = \Sigma^{1/2} \hat{\mathbf{W}} \quad (26)$$

The DMD modes can be calculated by substituting  $\mathbf{W}$  obtained from Eqn (26) in Eqn (15). The power of each DMD mode can be calculated by taking its L2 norm:

$$P_i = ||\phi_i|| \quad (27)$$

The power of the DMD modes is plotted against their frequency obtained by using Eqn (22). It is similar to calculating the power spectrum of a signal and can be used to select the modes.

- Another method to rank the DMD modes is by defining a cost function with a sparsity promoting term along with the accuracy of reconstruction, known as the Sparsity promoting DMD (SpDMD). However, this technique will not be used in this study.



### **1.3. OBJECTIVES OF THE WORK**

The objectives of this study are:

- Understand the working of the two decomposition algorithms, the Proper Orthogonal and Dynamic Mode Decomposition
- Code the two reduction methods and test them on synthetically generated mixed spatiotemporal signal as validation cases
- Deploy the method to study the coherent modes in a Jet Flow

### **1.4. SCOPE OF THE THESIS**

The report contains five chapters:

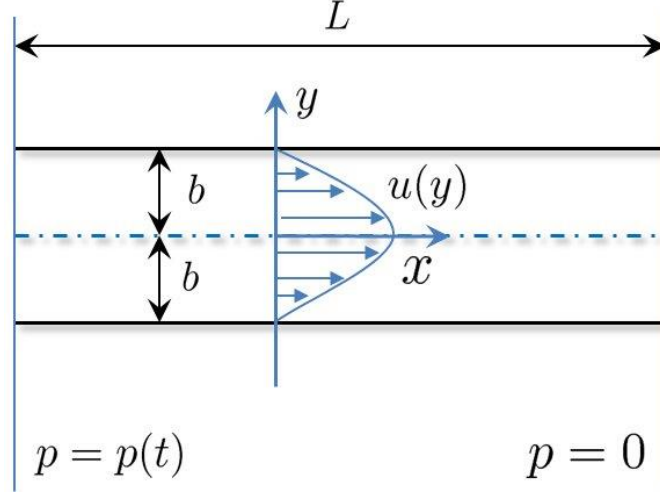
- Chapter 1: This chapter introduces POD and DMD and gives detailed methodology to be followed to obtain the orthogonal modes along with other related useful parameters
- Chapter 2: In this chapter, the developed POD and DMD algorithms are tested on two validation cases to verify their effectiveness in extracting coherent structures
- Chapter 3: In this chapter, the POD and DMD modes of a ventilated jet flow with three different velocity ratios is analyzed along with sensitivity analysis for different time durations and smallest time step of the simulation
- Chapter 4: This chapter presents the conclusion and the future works for this study
- Chapter 5: As part of future work, this chapter presents how the POD modes from an auxiliary Channel Flow simulation can be used to model the turbulent inlet boundary condition for a jet flow by developing a Low Order Dynamical System (LODS)

## CHAPTER 2

### VALIDATION CASES

#### 2.1. POD VALIDATION CASE: POISEUILLE FLOW

As seen below, Poiseuille Flow is a flow between two parallel plates with zero slip at the boundary plates:



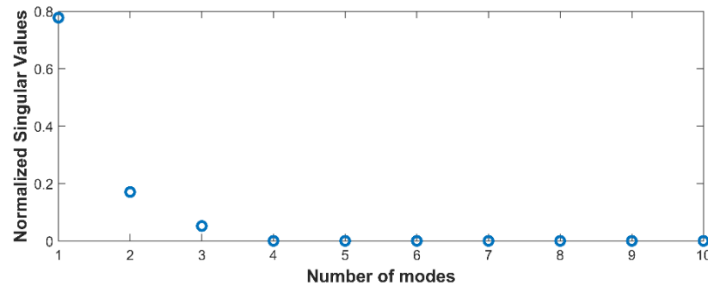
**Fig. 2.1** Schematic of Poiseuille Flow (Courtesy: Zdybal, K. 2016)

The pressure gradient and the dynamics of the flow are given by:

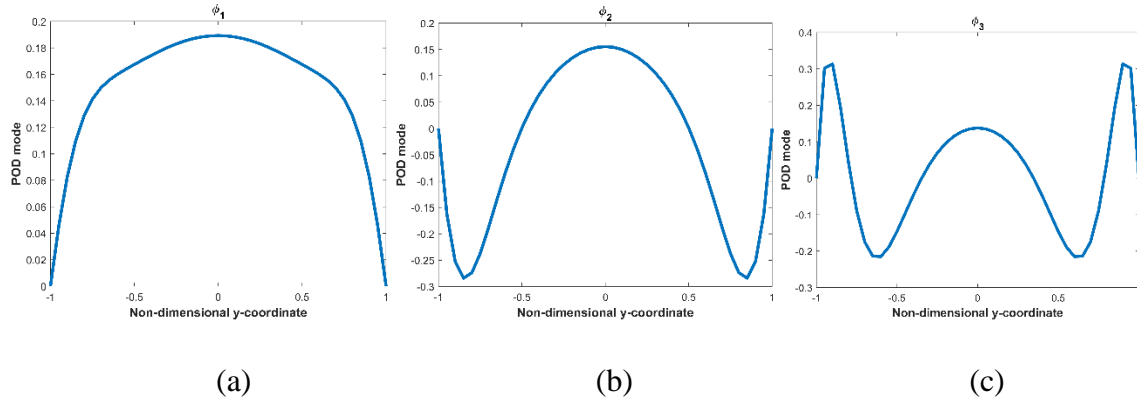
$$-\frac{\partial p}{\partial x} = -\left(\frac{p_m}{L} + \frac{p_a}{L} \cos(\omega t)\right); \quad -\frac{\partial u}{\partial t} = \left(\frac{1}{\rho}\right) \frac{\partial p}{\partial x} + \nu \frac{\partial^2 u}{\partial y^2} \quad (28)$$

The analytical solution for Eqn (28) is obtained using the Eigenfunction Expansion method and the asymptotic Complex Solution method. This has been used as the input data for the POD validation. Since the variation in velocity in the direction normal to the wall is of interest, a 1D analysis along the y-axis is conducted [4].

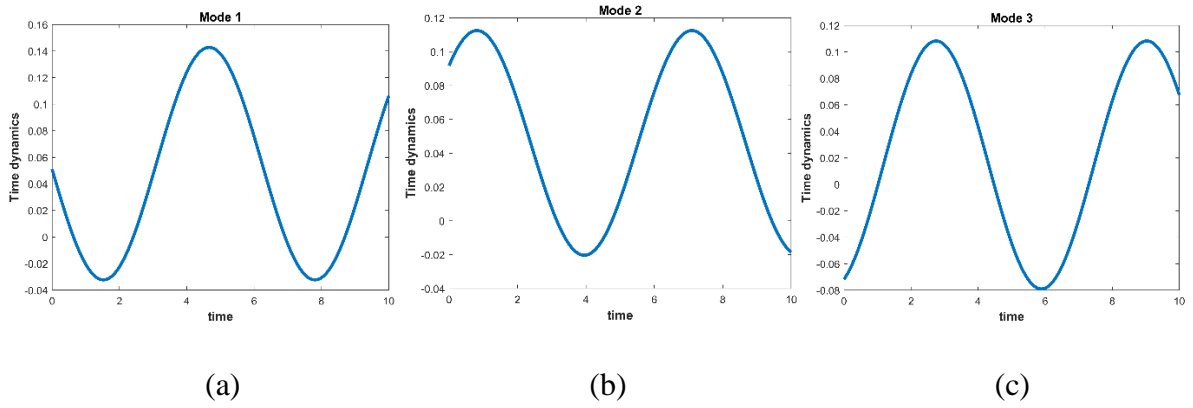
##### 2.1.1 POD ANALYSIS



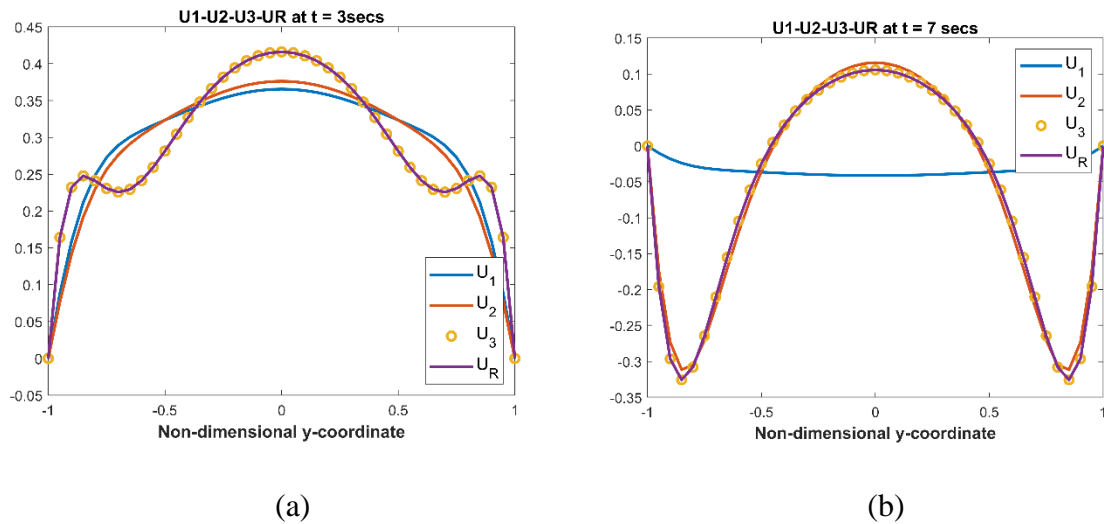
**Fig. 2.2** Normalized singular values of the Poiseuille Flow



**Fig. 2.3** The first three POD modes of Poiseuille Flow



**Fig. 2.4** The time dynamics of the first three POD modes of Poiseuille Flow



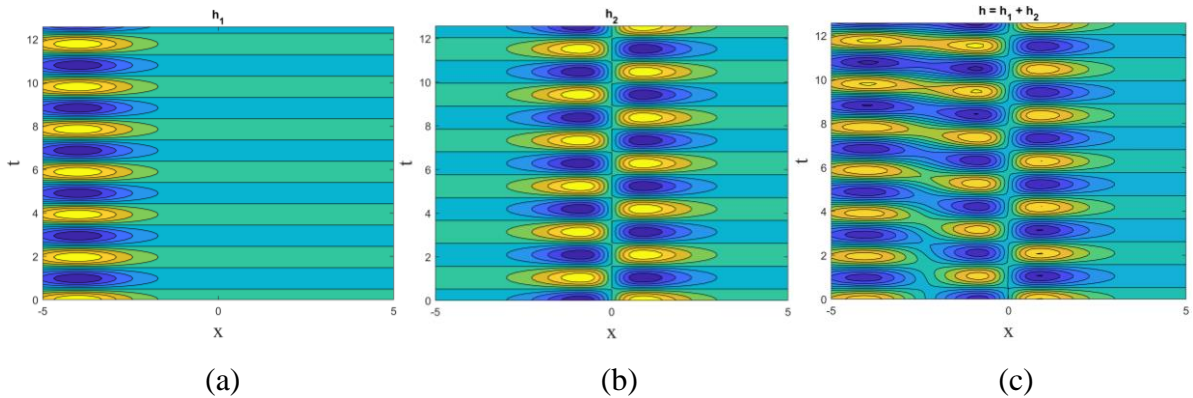
**Fig. 2.5** Reconstruction at  $t = 3$  sec using POD considering 1 mode ( $U_1$ ), 2 modes ( $U_2$ ) and 3 modes ( $U_3$ ). (c): Reconstruction at  $t = 7$  sec using POD considering 1 mode( $U_1$ ), 2 modes ( $U_2$ ) and 3 modes ( $U_3$ ).  $U_R$  is the analytical solution

The first three modes as seen in Fig. 2.2 are the most dominant ones, with the first, second, and third modes constituting 77.78%, 17.03% and 5.19% of the total kinetic energy. The amplitude decay rate is fast, and the faster it decays, the easier it is to approximate the solution with a small number of modes. The first three POD modes combined account for almost 99% of the energy, hence an accurate reconstruction can be performed using only these 3 modes as seen in Fig. 2.5 which shows the reconstructed data at two selected timestamps. The POD modes and their corresponding temporal coefficients are shown in Fig. 2.3 and Fig. 2.4 respectively. Also, worth noting is that the POD approximation technique meets the boundary criterion of no-slip, despite the fact that this information was not included in the procedures.

## 2.2. DMD VALIDATION CASE: MIXING OF TWO SPATIOTEMPORAL SIGNALS

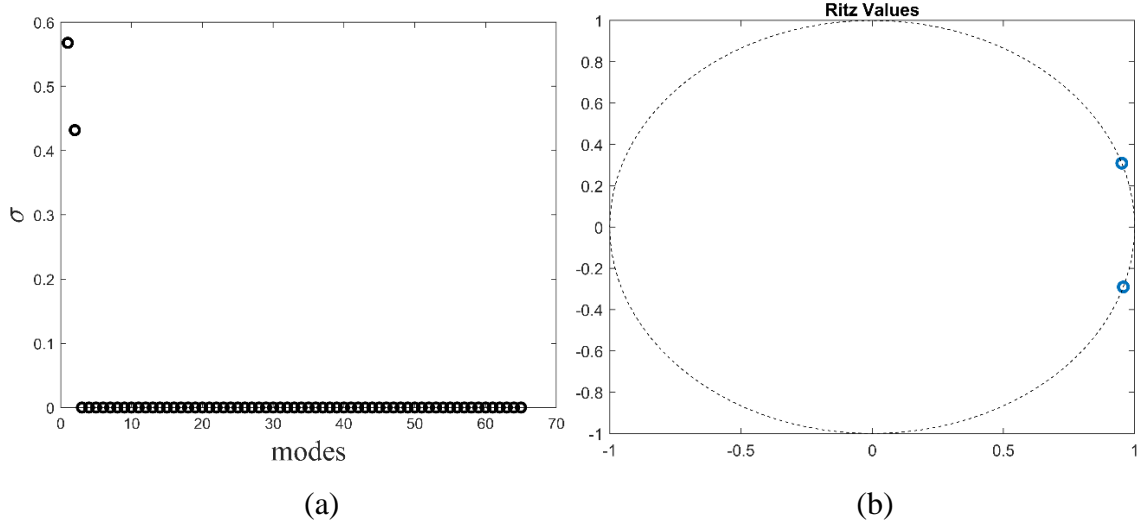
Two spatiotemporal signals are mixed as shown in Fig. 2.6 to obtain the resultant signal on which the DMD algorithm is implemented [13]. The two signals are:

$$\begin{aligned} h_1 &= \exp(i3.2t) \operatorname{sech}(x + 4) \\ h_2 &= 2 \exp(i3.0t) \tanh(x) \operatorname{sech}(x) \\ h &= h_1 + h_2 \end{aligned} \quad (29)$$

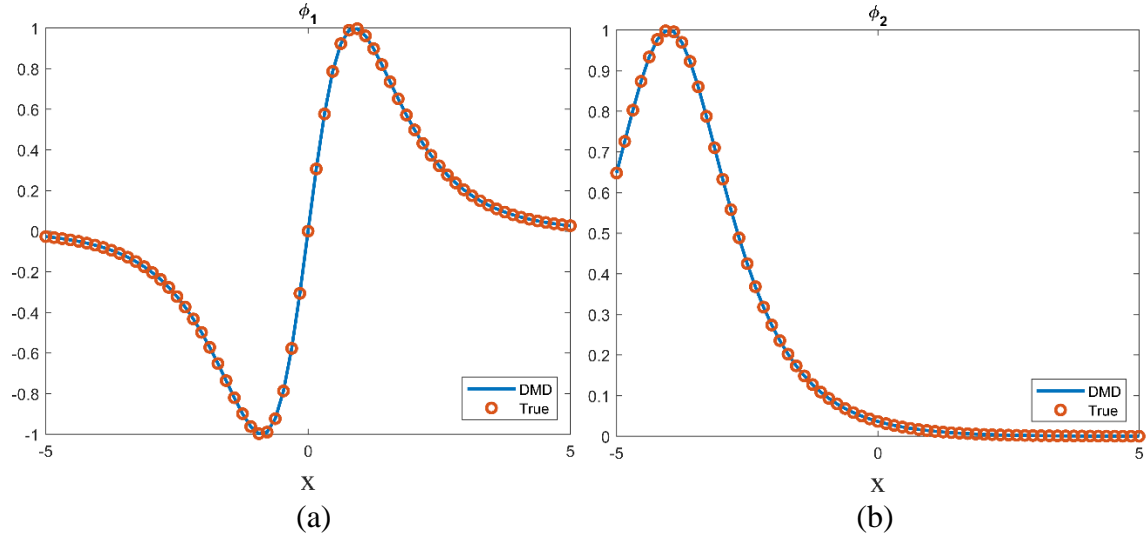


**Fig. 2.6** Original Spatiotemporal generated data, (a)  $h_1(x, t)$  (b)  $h_2(x, t)$  (c)  $h(x, t)$

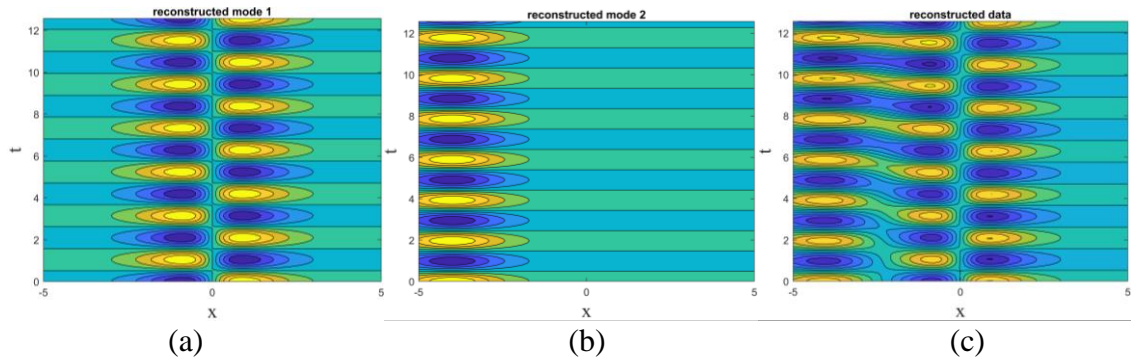
The DMD algorithm was implemented on  $h$  with rank  $r = 2$  truncation as the first two singular values were the dominant ones (Fig. 2.7 (a)). As shown in Fig. 2.9 (c), the reconstructed data is almost identical to the original synthetic data  $h(x, t)$  (Fig. 2.6 (c)). The reconstruction on the individual modes (Fig. 2.9 (a) and Fig. 2.9 (b)) matches  $h_1$  and  $h_2$  (Fig. 2.6 (a) and Fig. 2.6 (b)). The spatial components of the two signals are same as the DMD modes (Fig. 2.8).



**Fig. 2.7** (a) Singular Values, (b) Ritz values on the complex plane



**Fig. 2.8** (a) First DMD mode and true spatial value of first mode, (b) Second DMD mode and true spatial value of second mode



**Fig. 2.9** (a) Reconstruction on mode 1 (b) Reconstruction on mode 2 (c) Reconstruction on the first two modes

With these two synthetic test cases, the POD and the DMD algorithm is validated. It is then applied on the ventilated jet flow for analyzing its coherent structures.

## CHAPTER 3

### POD AND DMD ANALYSIS OF JET FLOW

#### 3.1. VENTILATED JET FLOW

In a ventilated offset planar jet, a jet flow is discharged from a nozzle parallel to a horizontal wall from an elevation while leaving area between the nozzle and the wall open. Negative pressure forms between the offset jet inlet and the wall as a result of the presence of the wall, entrapping a secondary ventilated jet that emerges from the gap between the nozzle and the bottom wall. This is a noteworthy problem with numerous applications, including thrust augmentation in aircraft design, drying processes, surface heating and cooling, fuel injection systems, and so on.

In this study, POD and DMD analysis of three cases with velocity ratios of 0.0, 0.10, and 0.18 are explored to better understand the relationship between the offset and the ventilated jet, where the velocity ratio is defined as:

$$\text{Velocity Ratio (vr)} = \frac{\text{Ventilated Jet Velocity}}{\text{Offset Jet Velocity}} \quad (30)$$

The ventilated jet case with  $vr = 0$  is the same as an offset jet flow with a vertical wall with no additional flow between the main nozzle flow at a height and the bottom horizontal wall. The simulation domain contains a planar offset jet that is defined by its offset ratio. It is the ratio of the height between the primary jet and the bottom wall ( $h$ ) to the jet's width ( $w$ ). In this study, the offset ratio is  $h/w = 3.5$ . The inlet for the ventilated jet is located below the offset jet. The primary and ventilated jet inlets are provided with a uniform velocity profile produced using RANS simulation of channel flow with the same Reynolds number for velocity boundary conditions. Random fluctuations are superimposed on the velocity profile. To compensate for the lack of correlated fluctuations, extremely high turbulence intensity is given. To model the entrainment caused in the outer region, a downward velocity component is supplied at the top boundary. Back flow is restricted by an advective boundary condition at the domain's exit. A zero-pressure gradient boundary condition is provided at the inlet and the walls. The boundaries in the spanwise direction are defined to be periodic.

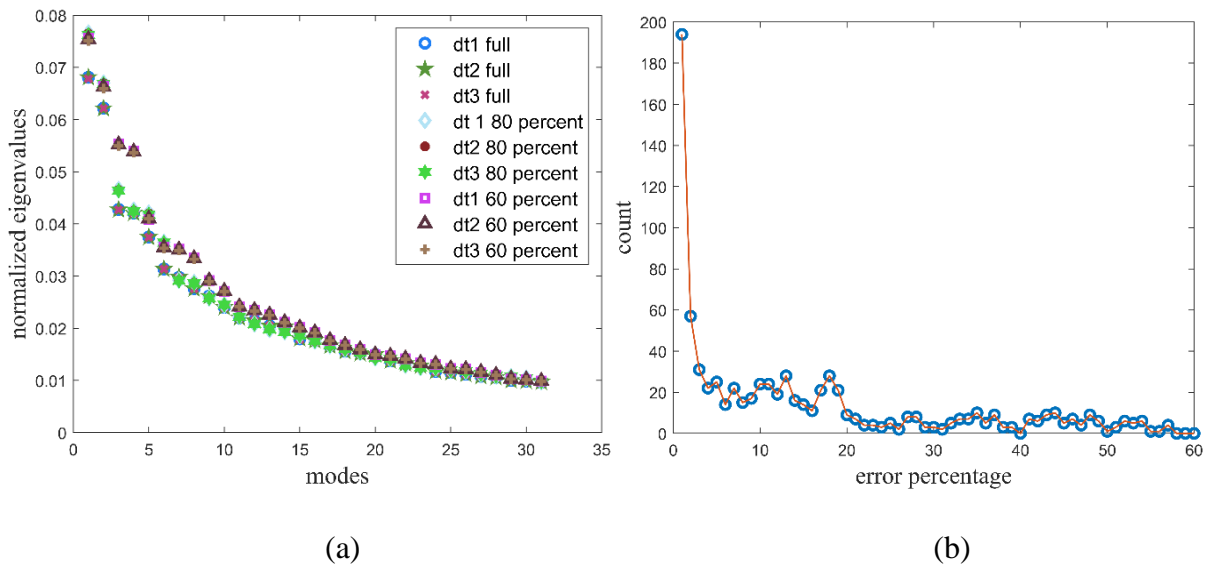
#### 3.2 SENSITIVITY ANALYSIS

The time step of the snapshots and the number of snapshots taken in the POD/DMD analysis have a significant impact on its results. Hence, a sensitivity study is performed for  $vr = 10$ . The simulation timespan  $T_s$  was taken as the time taken by the mean flow to cross the domain length as defined below:

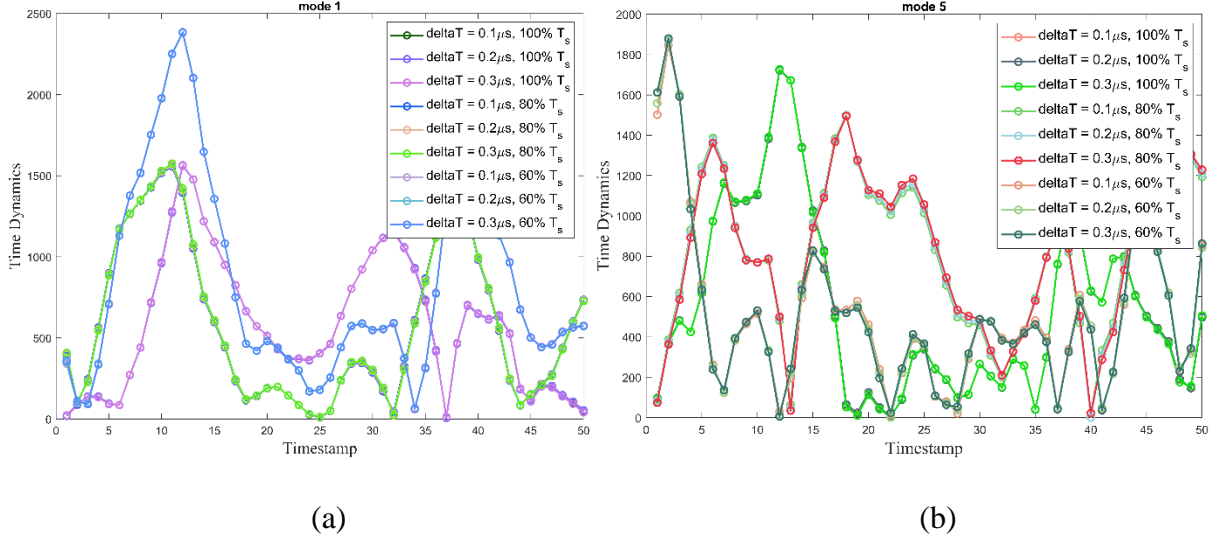
$$T_s = \frac{L}{U_{\text{mean}}} \quad (31)$$

where  $L$  is the domain length and  $U_{\text{mean}}$  is the mean velocity. For mean velocity taken at the outlet, a timespan of 13.4 milliseconds is obtained and 2.65 milliseconds with inlet mean velocity. Hence, the timespan  $T_s$  was taken to be 7 milliseconds, which is approximately the average of the two. Three timespans – 60% of  $T_s$ , 80% of  $T_s$  and 100% of  $T_s$  and three-time step  $\text{delta}T = 0.1 \mu\text{s}$ ,  $0.2 \mu\text{s}$ , and  $0.3 \mu\text{s}$  were considered for the sensitivity analysis, resulting in 9 different test cases.

For POD, the eigenvalues and the temporal coefficients are analyzed for the 9 cases to test sensitivity [10]. The eigenvalues are compared in Fig 3.1. The 30<sup>th</sup> mode contains approximately 1% energy content. Hence, only the first 30 modes are compared. The eigenvalues of all the test cases are mostly overlapping and have similar decay rate. With the test case having the highest timespan and the smallest time step as the reference, the percentage difference between the same eigenvalues of the different cases is shown in Fig 3.1 (b). The y-axis in Fig 3.1 (b) is the count of the number of eigenvalues with the corresponding percentage error in the x-axis. It can be seen that most of them have less than 20% difference.

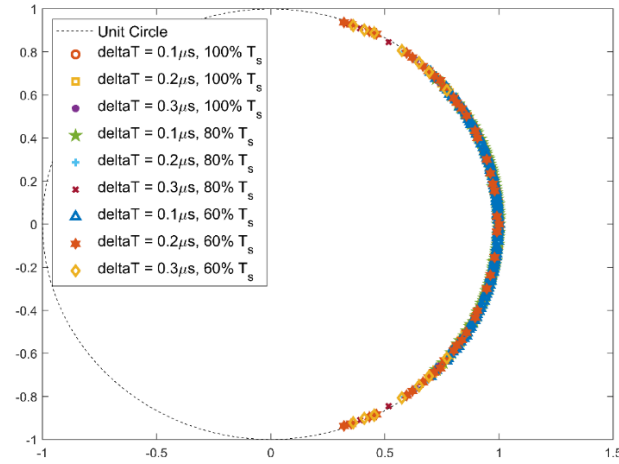


**Fig. 3.1** (a) Normalized eigenvalues of the 9 test cases (b) The count of the eigenvalues with corresponding percentage difference with the first test case



**Fig. 3.2** (a) Temporal coefficient of the first mode for the 9 test cases, (b) Temporal coefficient of the fifth mode for the 9 test cases

The temporal coefficients of the first and the fifth modes for all the test cases are compared in Fig 3.2. The temporal coefficients of the same timespan but different  $\delta T$  are overlapping. However, they change significantly with variation in timespan, which implies that the POD analysis is highly sensitive to timespan of the simulation considered. Therefore, the final timespan considered for the analysis is  $100\% T_s$  as it will capture more information about the flow.



**Fig. 3.3** Ritz Values plotted on the complex plane along a unit circle for the 9 test cases

Sensitivity analysis for DMD is more complicated as compared to POD. The Ritz values for all the test cases are compared in Fig 3.3. Only the stable modes have been considered for checking sensitivity. As mentioned earlier, stability of the DMD modes is directly related to separation of the Ritz value from a unit circle. If they are close to the unit



circle, they will have a much slower decay or growth rate depending on whether they are inside or outside the unit circle respectively. This implies that the magnitude of the complex Ritz value should be close to 1 for better stability. Hence, the DMD modes satisfying the criteria mentioned below have been used in this study:

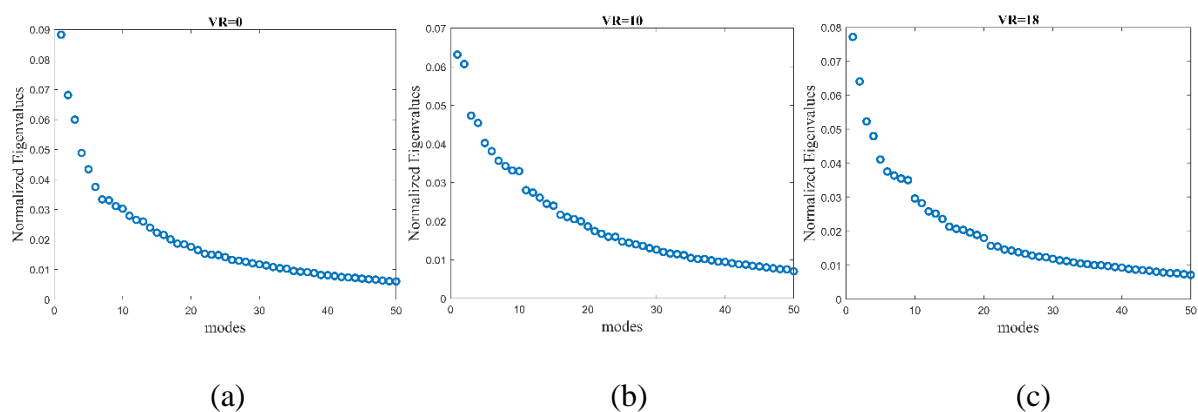
$$||\lambda| \leq \epsilon_{\max} \quad (32)$$

where  $\lambda$  is the Ritz value and  $\epsilon_{\max}$  is the maximum permissible error, which has been taken to be 0.01 in this analysis. As seen in Fig 3.3, the Ritz values for all the test cases follow similar trends and are concentrated on the right half of the unit circle.

### 3.3. POD ANALYSIS

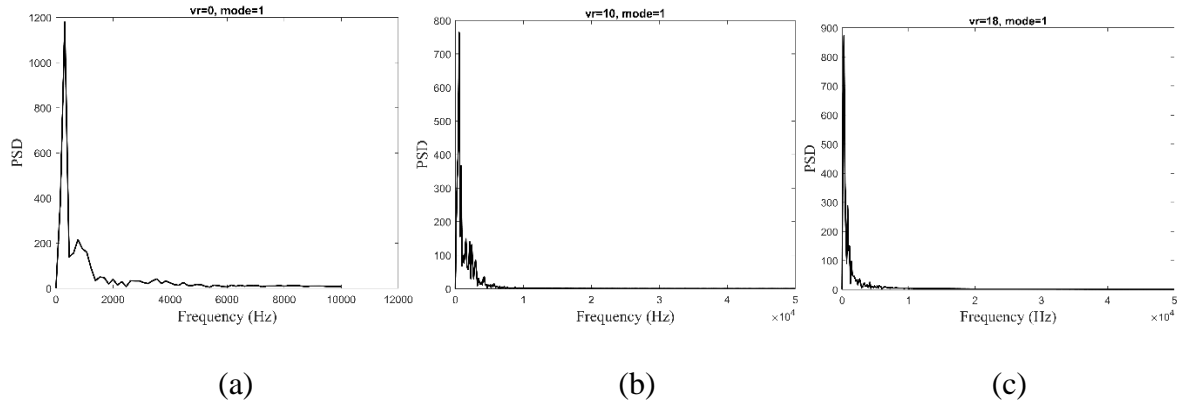
For brevity, only the eigenvalues and power spectrum of the first mode have been analyzed for POD. The POD modes have not been included in this study. A more detailed analysis for DMD has been performed in the next section.

The normalized eigenvalues representing the energy content of the POD modes have been shown in Fig 3.4. The mean flow has been subtracted from the velocity field. The first POD mode for  $vr = 0$  has a higher energy content of 8.82% as compared to  $vr = 10$  and  $vr = 18$ , for which the first mode contains 6.31% and 7.72% energy fraction respectively. The number of modes to account for 90% of the cumulative energy content is 37 for  $vr = 0$ , which is less than the other two cases. The decay rate of the eigenvalues is also higher for the same. It is observed that the dominant eigenvalues occur in pair, which is a characteristic property of large-scale energy structures [11].



**Fig. 3.4** POD normalized eigenvalues for (a)  $vr = 0$  (b)  $vr = 10$  (c)  $vr = 18$

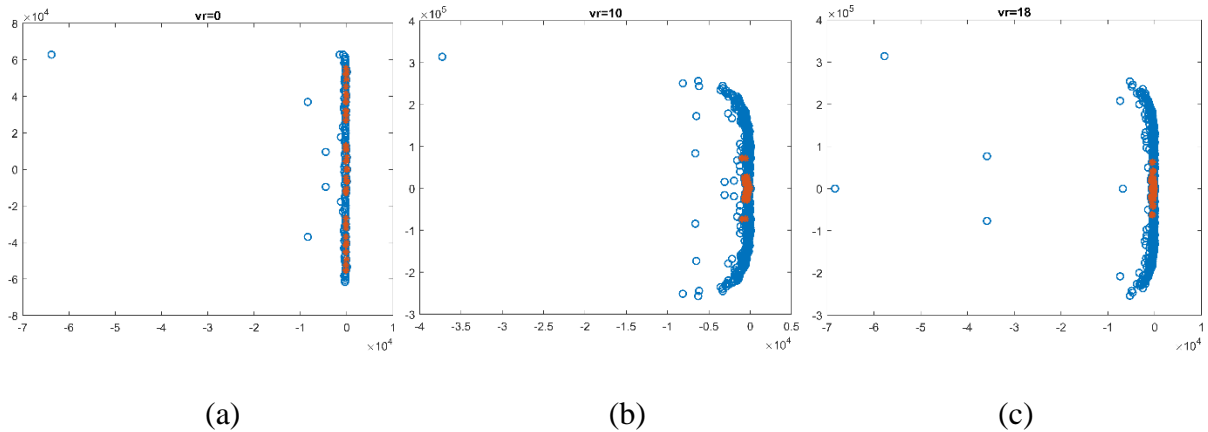
POD decomposes the modes based on their energy. Hence, the POD modes (not shown here) have larger vortex structures associated with them. Unlike DMD, which uses frequency to decompose the modes, the modes obtained using POD have multiple periodic structures associated with them, which can be seen from the Power Spectral Density (PSD) plots of the first mode for the three velocity ratios in Fig 3.5, which has multiple peaks signifying the presence of several dominant frequencies for a single mode. In this sense, DMD is more efficient whose modes have a single frequency and growth/decay rate associated with them.



**Fig. 3.5** PSD plot of (a)  $vr = 0$  (b)  $vr = 10$  (c)  $vr = 18$

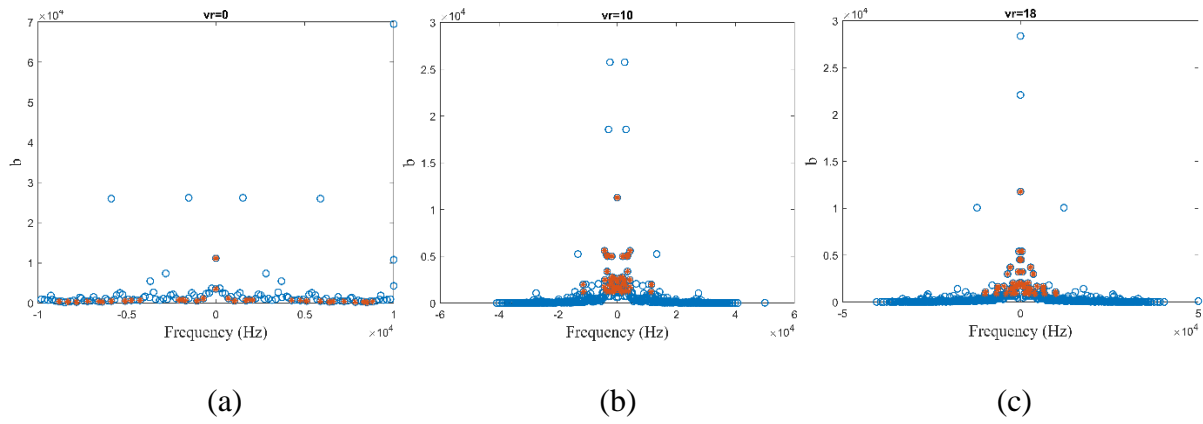
### 3.4. DMD ANALYSIS

The DMD modes have been selected based on the stability criteria as shown in Eqn (32) and are ranked based on the magnitude of their amplitude  $b$  as defined in Eqn (20). The logarithmic mapping of the Ritz values on the complex plane for the three velocity ratios is shown in Fig 3.6, which is an alternative to visualizing the eigenvalues where the imaginary component is proportional to the frequency of oscillations and the real part to its damping. The points on the left half plane of this plot are the decaying modes and the modes in the right half are growing with time. The modes close to the imaginary axis remain stable with time.



**Fig. 3.6** The logarithmic plot of the Ritz values defined as  $\mu = \log(\lambda) / \Delta_t$  for (a)  $vr = 0$  (b)  $vr = 10$  (c)  $vr = 18$  plotted on the complex plane (The blue circles correspond to all the modes and the ones shown as orange triangles are the selected modes based on the selection criteria)

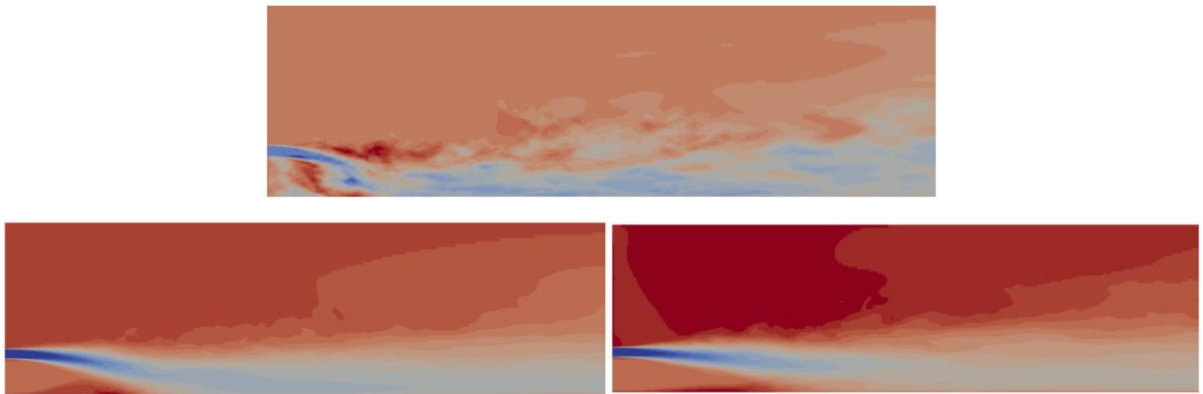
The amplitude  $b$  of the DMD modes is plotted against its frequency in Fig 3.7. These amplitude have been used for ranking the DMD modes, modes with higher amplitude and stability in terms of location of the Ritz value with respect to unit circle have been selected. Few modes of  $vr = 0$  have a higher magnitude amplitude as compared to the other two cases. It is observed that the modes with higher amplitude have a smaller frequency, implying that energetic structures have slower time dynamics.



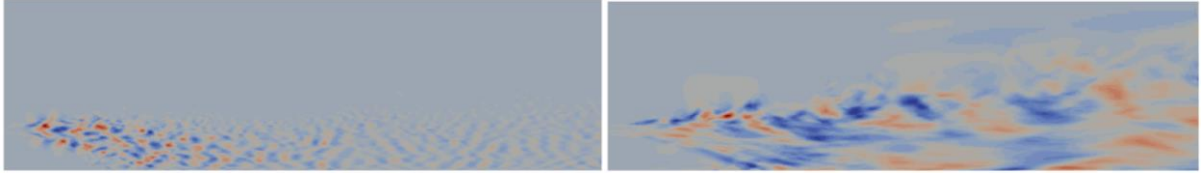
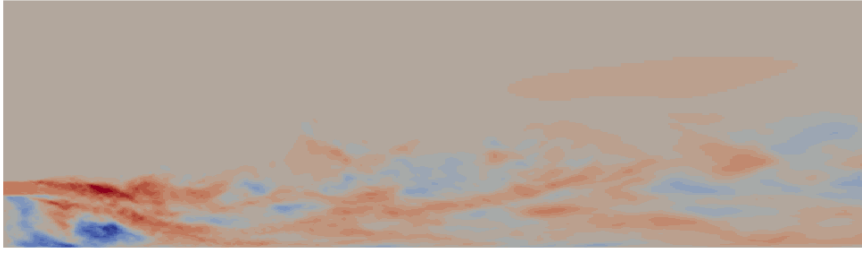
**Fig. 3.7** The amplitude of the DMD modes for (a)  $vr = 0$  (b)  $vr = 10$  (c)  $vr = 18$

### 3.4.1 DMD MODES

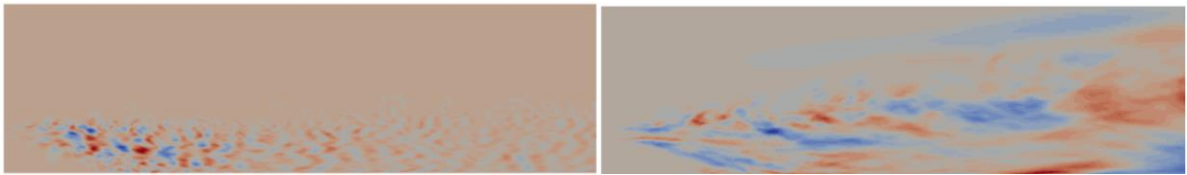
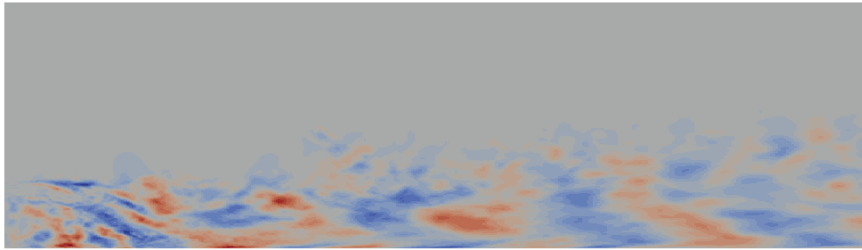
As the u-component of the velocity is more dominant as compared to v and w, the DMD modes of only the u-component is analyzed. The first five DMD modes based on the ranking algorithm mentioned earlier are shown in Fig 3.8. The DMD modes occur in complex conjugate pairs with the same amplitude. Hence, every alternate DMD mode have been omitted in this study.



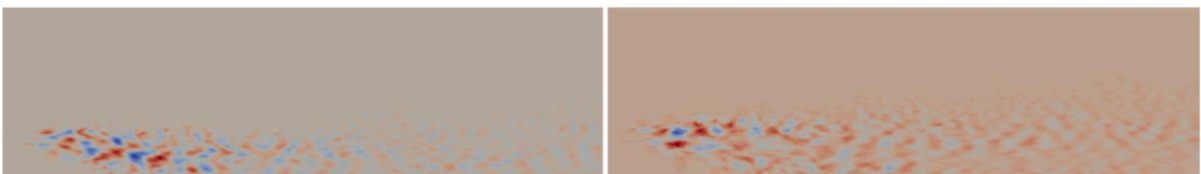
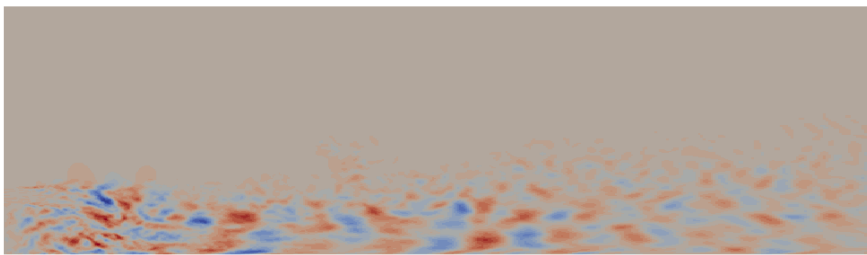
**MODE 1**



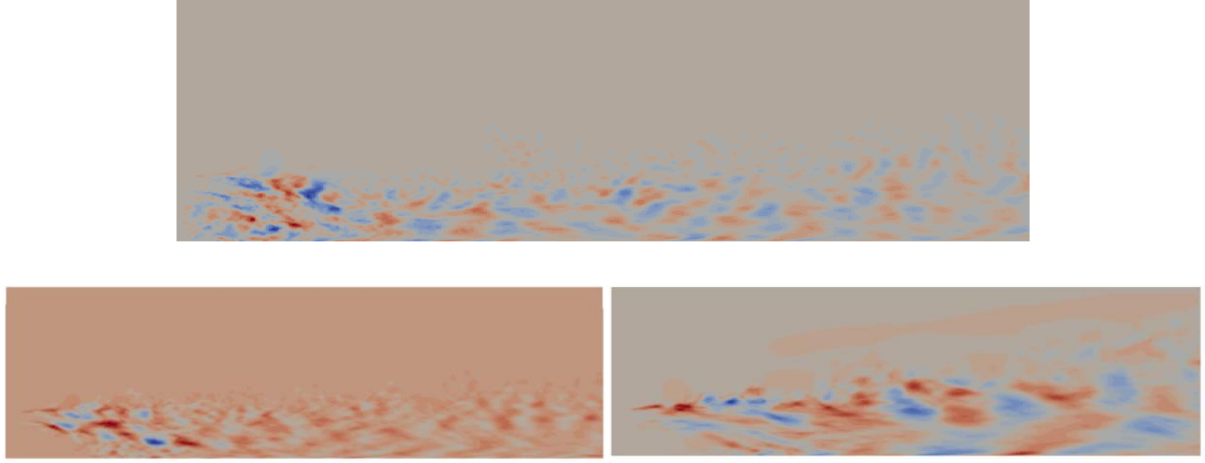
**MODE 2**



**MODE 3**



**MODE 4**



### MODE 5

**Fig. 3.7** The first 5 DMD modes ( $vr = 0\%$  (top),  $vr = 10\%$  (left) and  $vr = 18\%$  (right))

The damping factor  $\zeta$  of the DMD mode can be defined as [12]:

$$\zeta = \frac{\ln(\text{abs}(\lambda))}{\Delta_t} \quad (33)$$

The frequency  $f$  and the damping factor  $\zeta$  for the five modes are shown in Table 3.1.

**Table 3.1** The frequency and damping factor of the selected modes

Mode	$f$ (cycle/sec), $vr=0$	$\zeta$ , $vr=0$	$f$ (cycle/sec), $vr=10$	$\zeta$ , $vr=10$	$f$ (cycle/sec), $vr=18$	$\zeta$ , $vr=18$
1	0	-29.5	0	-3.3	0	2.3
2	0	92.2	4318.0	-730.1	437	-593.3
3	702.9	-125.1	3555.6	-665.2	172	-521.9
4	1865.6	-195.8	3331.5	-688.5	2854.6	-781.4
5	2064.8	-164.3	1862.5	-625.9	544.8	-320.9

The first mode for the three cases is representative of the mean flow. The jet from the main nozzle is clearly visible and follows a streamline like structure before dispersing near the end region. For  $vr = 18$ , the primary jet travels a longer distance before impinging onto the bottom wall and is less curved. For  $vr = 0$ , the flow gets dispersed much faster. This is because as  $vr$  increases, the suction pressure in the initial region drops gradually, causing the jet to spread away from the bottom wall. As the first mode is symbolic of the mean flow, it has no periodic structures associated with it which is evident by the fact that the associated frequency is approximately zero as seen in the first row of Table 3.1.

For modes 2, 3 and 5,  $\nu r = 10$  has alternating vortex structures which decays in the streamwise direction.  $\nu r = 18$  contains larger periodic structures and is more spread out further downstream as seen in the top right half for the same modes. As can be inferred from Table 3.1, larger vortex structures correspond to smaller frequencies and vice versa. Mode 4 of  $\nu r = 18$  has smaller vortices and a higher frequency as compared to its other modes. A characteristic feature of the offset jet is the recirculation region which is less dominant in the presence of the ventilated jet. For  $\nu r = 0$ , modes 3 and 4 show clockwise vortices in the recirculation region. For the ventilated case, this region gets shifted in the streamwise direction due to the secondary jet below the nozzle which has a pushing effect on the primary flow.

For brevity, only five modes were analyzed in this study. More modes could be considered to get a better idea of the flow. For DMD, other ranking algorithms like SpDMD, multi-resolution DMD for transient phenomena, compressed DMD for computational speedup, and so on could be used to get modes based on different criteria.

## CHAPTER 4

### CONCLUSION AND FUTURE WORK

In the first chapter, the POD and the DMD algorithms were laid out in detail. The second chapter presented two validation cases – POD analysis of Poiseuille Flow and DMD analysis of a mixed spatiotemporal signal, to confirm the ability of the two algorithms to decompose the input flow field into orthogonal modes. The ventilated jet flow was introduced in the third chapter. A parametric study was done for three velocity ratios  $vr$  including  $vr = 0$ ,  $vr = 10\%$  and  $vr = 18\%$  to understand the influence of the ventilated jet on the main nozzle flow. The POD and DMD analysis of the u-component of the velocity for the three velocity ratios was performed to obtain the coherent structures and vortexes in the flow field. Before applying the two decomposition algorithms on the velocity dataset, a sensitivity analysis was carried out to check the effect of the simulation time-step  $\Delta T$  and the timespan on the POD and DMD results. Nine test cases, with three different  $\Delta T$  and three timespans for  $vr = 10$  were considered for the sensitivity analysis.

Future work of this study includes the use of POD and DMD for turbulent inlet boundary condition generation. Most papers add Gaussian noise to the mean profile of the inlet velocity to generate turbulence. POD and DMD modes of an auxiliary simulation like a channel flow can be extracted and the temporal dynamics of the dominant modes can be obtained using methods like Galerkin projection or Low Order Model Identification. The modes and their temporal evolution can be used to model the inlet velocity condition for the main simulation, which in this study is the ventilated/offset jet flow and the resultant flow field could be compared with the existing results. A more detailed discussion of this approach has been provided in the appendix.

## APPENDIX

### A.1 TURBULENT INLET BOUNDARY CONDITION GENERATION

#### A.1.1 INTRODUCTION

The database obtained from an auxiliary simulation is interfaced to a computational algorithm to generate turbulent inflow boundary conditions. This method relies on POD to interpolate and extrapolate auxiliary simulation data onto the numerical mesh and describe both the temporal dynamics and the spatial organization of the flow in the offset jet flow's inlet patch. To describe the temporal development of the most energetic structures, a low-order dynamical model is derived from the auxiliary simulation database. To simulate the temporal evolution of higher order POD modes, time series of Gaussian random numbers are used to model incoherent motion.

As proposed in [5], the velocity field can be subjected to triple decomposition. Hence, the mean flow  $\mathbf{u}_m$ , the coherent part  $\mathbf{u}_c$ , and the random part  $\mathbf{u}_r$  were considered as the three mutually independent components of the velocity field. The POD modes are used to model the coherent and the incoherent part. For the contribution of the coherent part, the first  $N_c$  highly energetic POD modes are considered, where  $N_c$  are the selected dominant POD modes having the highest contribution to the energy spectrum. As will be explained in a later section, the temporal POD coefficients reflecting the dynamic behavior of the coherent part of the flow, denoted by  $a(t)$ , are modelled by a Low Order Dynamical System (LODS) determined from an auxiliary simulation. Random number series are used to describe the temporal coefficients associated with the uncoherent component, denoted by  $b(t)$ . The modeled velocity field  $\hat{\mathbf{u}}$  can hence be represented as a superposition of the mean flow, the coherent and the uncoherent part as shown below [6]:

$$\hat{\mathbf{u}}(t) = \mathbf{u}_m + \mathbf{u}_c + \mathbf{u}_r = \mathbf{u}_m + \sum_{k=1}^{N_c} a_k(t)\phi_k + \sum_{k=N_c+1}^{N_t} b_k(t)\phi_k \quad (34)$$

where  $N_t$  is the total number of modes obtained from POD. Based on their energy contribution, the first  $N_c$  modes are selected for developing the LODS. However, model stability issue arises from the fact that just a small number of modes are kept, thereby limiting the impact and interaction of the other modes. Hence, the number of modes retained to derive the LODS is an important parameter to be finalized based on the case-specific study.



### A.1.2 LOW ORDER MODEL IDENTIFICATION

The most common approach in developing a reduced order model is to use Galerkin projection. In this method, the selected POD modes are substituted into the governing ODEs (which mainly includes the Navier Stokes and the continuity equation for a Direct Numerical Solution) and the equations are rearranged to obtain the time evolution of the temporal coefficients of the POD modes. However, the system of ODEs has to be solved for the entire domain. In the presented approach, the dynamics can be solved for a desired subdomain, which is this study, is obtaining the dynamics of the POD modes of a 2D plane from 3D simulation results. It assumes that its dynamics are driven by equations that are similar to those that may be derived from information of the complete flow field. As a result, it can be characterized by a set of polynomials and at most cubic ODEs of the form:

$$\dot{a}_m(t) = A_m + B_{mn}a_n + C_{mno}a_na_o + D_{mnop}a_na_oa_p \quad (35)$$

where  $1 \leq m \leq n \leq o \leq N_c$ ,  $N_c$  being the number of POD modes used for developing the LODS. The unknown parameters of the dynamical system that must be extracted from the data are  $A_m$ ,  $B_{mn}$ ,  $C_{mno}$  and  $D_{mnop}$ . The equation can be rearranged to get:

$$\dot{a}_m(t) = \sum_{i=1}^{N_p} z_i A_i(a_1, \dots, a_{N_c}) \quad (36)$$

where  $z_i$  are the unknown parameters and  $A_i$  are polynomials which are at most cubic in nature.  $N_p$  is the total number of unknowns to be determined from the LODS. For a LODS with up to cubic polynomials,  $N_p$  can be calculated as shown below:

$$N_p = 1 + N_c + \frac{N_c(N_c+1)}{2} + \frac{N_c(N_c+1)(N_c+2)}{6} \quad (37)$$

where the four terms on the right-hand side of Eqn (37) correspond to the constant, linear, quadratic and cubic terms of Eqn (35) respectively. Finite difference is used to calculate the time derivative  $\dot{a}_m(t)$  as shown below:

$$\dot{a}_m \left( t_k + \frac{\Delta_t}{2} \right) = \frac{a_m(t_k + \Delta_t) - a_m(t_k)}{\Delta_t} \quad (38)$$

where  $\Delta_t$  is the simulation time-step. For all successive tuples of  $(a_m(t_k), a_m(t_k + \Delta_t))$ , its derivative can be approximated as shown above. To be consistent with the timestamps, the temporal coefficients at the intermediate time-step is used which is obtained by taking the tuple's average as shown below:

$$a_m \left( t_k + \frac{\Delta t}{2} \right) = \frac{a_m(t_k) + a_m(t_k + \Delta t)}{2} \quad (39)$$

Given  $N_t$  tuples of  $(a_m(t_k), \dot{a}_m(t_k))$  at discrete time instants  $t_k$  obtained from the POD of the auxiliary simulation, Eqn (36) can be solved with the help of least mean square regression to optimize the following cost function:

$$R^2 = \sum_{k=1}^{N_t} [\dot{a}_m(t_k) - \sum_{i=1}^{N_c} z_i A_i (a_1(t_k), \dots, a_{N_c}(t_k))] \quad (40)$$

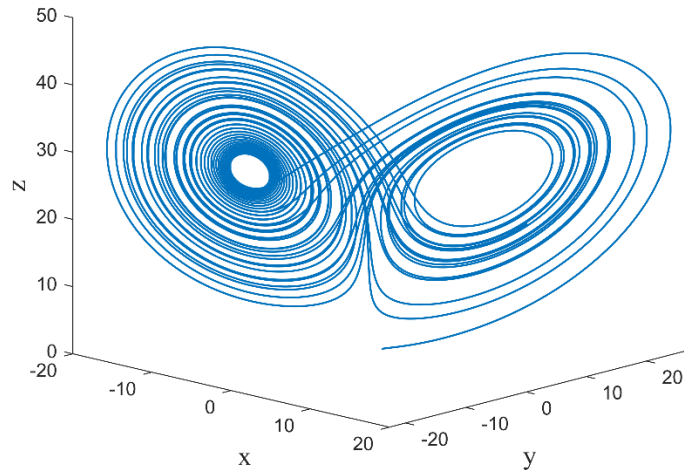
After calculating the unknown parameters by solving the regression problem, Eqn (35) can be integrated based on the given initial condition to obtain the estimated temporal coefficient  $\hat{a}$ . Hence, the reconstructed coherent part of the velocity field based on the dominant modes can be written as:

$$\hat{u}(\mathbf{x}, t) = \sum_{i=1}^{N_c} \hat{a}_i(t) \phi_i(\mathbf{x}) \quad (41)$$

### A.1.3 LODS TEST CASE: LORENTZ SYSTEM

The aforementioned method to develop the LODS is tested on the Lorentz system. The Lorentz system is a set of ODEs with chaotic solutions for specific parameter values and initial conditions. The dynamics is described by three ODEs, known as the Lorentz equations, as shown below:

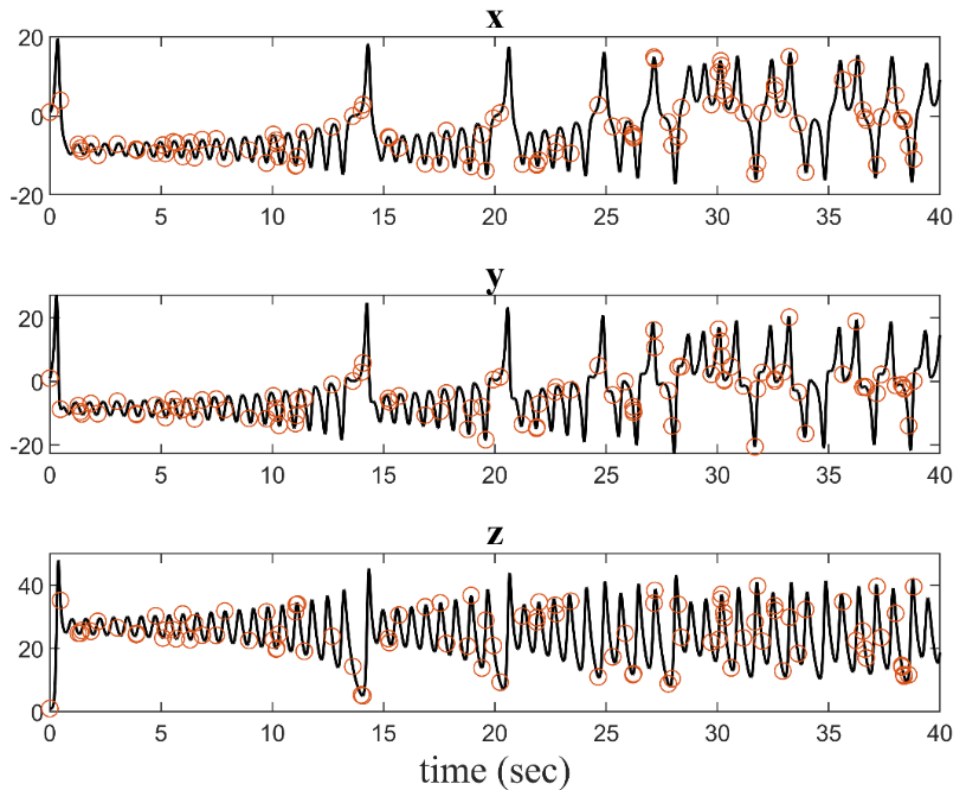
$$\dot{x} = \sigma(y - x); \quad \dot{y} = x(\rho - z) - y; \quad \dot{z} = xy - \beta z \quad (42)$$



**Fig. 5.1** Phase plot of the  $x$ ,  $y$  and  $z$  given by Eqn (42)

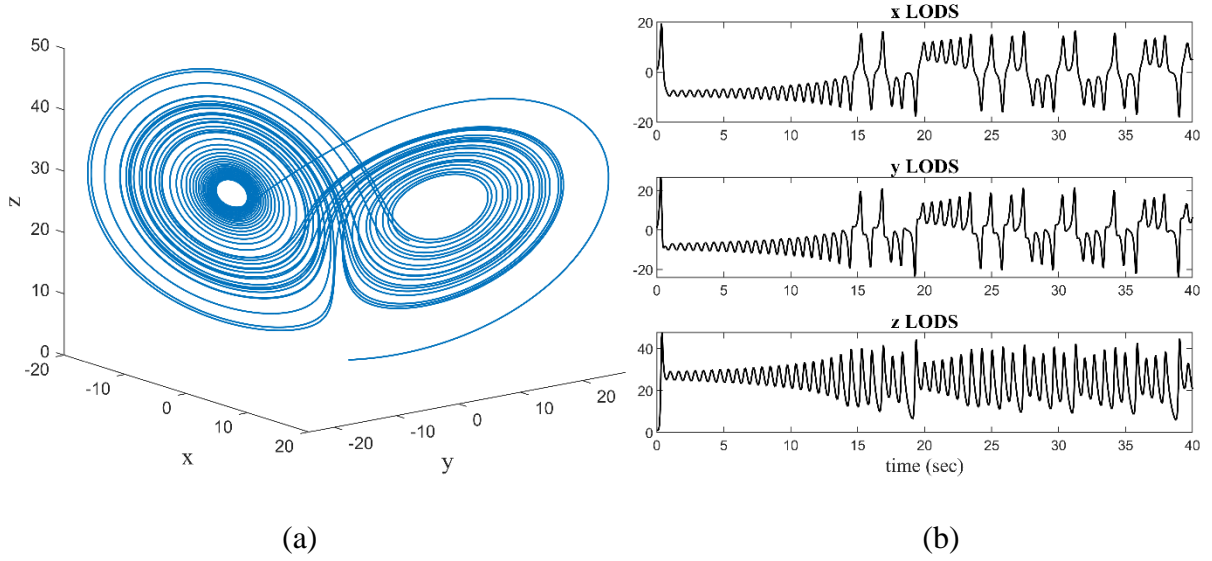
The parameters  $\sigma$ ,  $\rho$ , and  $\beta$  are constants related to the system and are proportional to the Prandtl number, Rayleigh number, and the geometric characteristics. The phase plot of the

Lorentz system for  $\sigma = 8/3$ ,  $\rho = 28$  and  $\beta = 8/3$  is shown in Fig 5.1, which has two characteristic strange attractor associated with it. The system of ODEs shown in Eqn (42) was integrated up to 40 sec to get the time series values for  $x$ ,  $y$  and  $z$  with the initial values for each variable equal to 1. The time step was taken to be 0.001 sec to generate 40001 data points for each variable. The solution obtained is periodic in nature. Hence, sample points for developing the LODS were taken at random to avoid matching with the inherent frequencies of the system. To show that the LODS can be built from only a handful of sample points, only 100 random points were used from the original dataset to determine the unknown coefficients of the LODS as shown in Fig 5.2 (a).



**Fig. 5.2** The time dynamics of  $x$ ,  $y$  and  $z$  along with the 100 random sample points (shown as orange circles)

The time derivative at the corresponding sample points was calculated as shown in Eqn (38). With these 100 random sample points containing tuples of the temporal value and their derivative, the unknown coefficients of the LODS were obtained and then integrated with the same initial conditions to get the reconstructed dynamic system as shown in Fig 5.3 and Fig 5.4 and are remarkably close to the original system.



**Fig. 5.3** (a) Reconstructed LODS phase plot; (b) Reconstructed LODS time dynamics of x, y and z

#### A.1.4 RANDOM UNCOHERENT MOTION

The POD modes with lower energy content not included in the LODS are represent by random number series which have the POD modes' eigenvalues incorporated into them. A random number generator with the following probabilistic distribution  $g(y)$  is used [7]:

$$g(y) = \frac{2}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} \quad (43)$$

where  $y$  is the random number. To enforce correlation with the past time steps, the current random number  $y_{new}$  is averaged with the previous random number  $y_{old}$  as shown below:

$$y_{new} = \frac{y_{new} + y_{old}}{2} \quad (44)$$

The temporal amplitude is then correlated with the eigenvalue of the POD modes and the random number generated as follows:

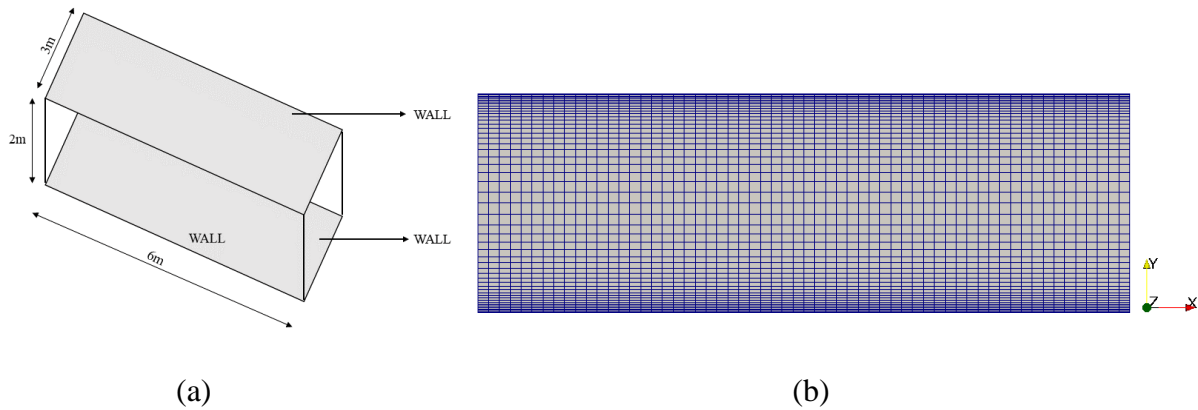
$$a_{new}(t) = x_{new} \sqrt{3\lambda} \quad (45)$$

This can be used to model the temporal coefficients of the low energy POD modes to represent the random uncoherent part of the velocity field.

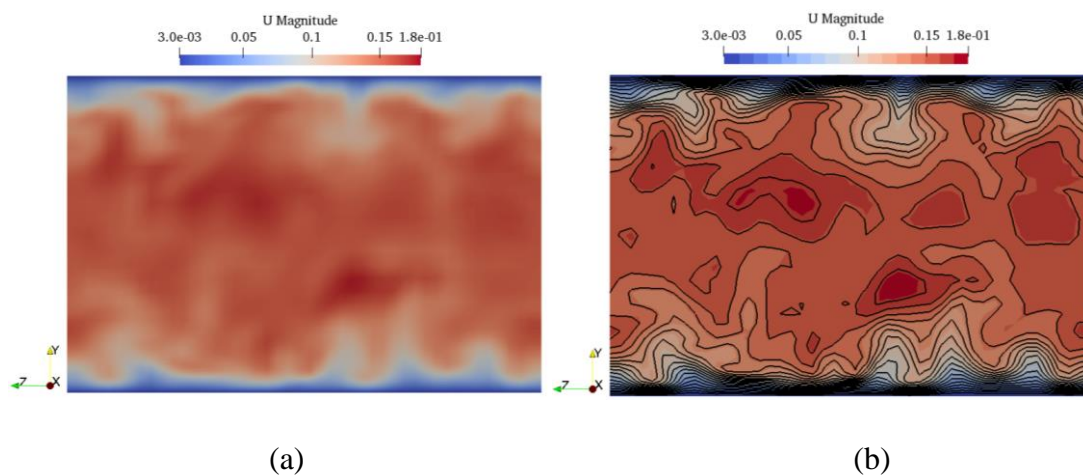
### A.1.5 AUXILIARY CHANNEL FLOW SIMULATION

A 3D LES channel flow simulation using OpenFOAM is used as the auxiliary simulation for generation of turbulent inlet boundary condition. The dimensions of the inlet in the channel flow simulation will be the same as the inlet patch used in the ventilated/offset jet flow simulation. The channel flow simulation used in this study has been taken from [8].

A Cartesian coordinate system is used to understand the flow components. The negative x-axis is oriented in the pressure gradient's direction. Because this orientation is the same as the mean flow, it is referenced to as the streamwise direction. The y-axis is assumed to be perpendicular to the walls of the channel flow, pointing from the bottom to the upper wall. This direction is called as the wall-normal. Lastly, the z-axis is selected so that the coordinates are mutually orthonormal and forms the spanwise direction. The bulk Reynolds number  $Re_b$  for this simulation is 13350 and the friction Reynolds number  $Re_\tau$  is 350. The geometry of the simulation domain is shown in Fig 5.4.



**Fig. 5.4** (a) Geometry of Channel Flow simulation (b) Meshing of the computational domain

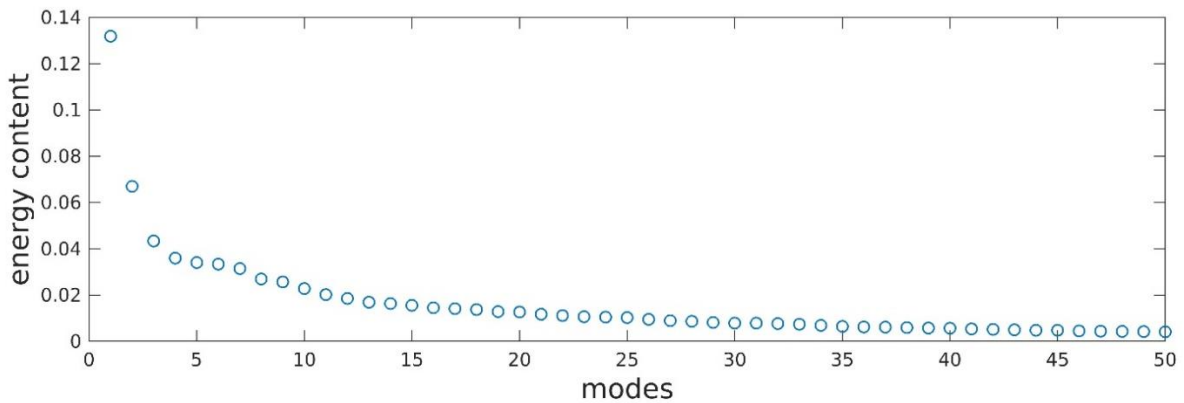


**Fig. 5.5** (a) The sectional view of the channel flow (b) Contour plot

The mesh used in the simulation is shown in Fig 5.4 (b). The resolution is  $240 \times 200 \times 180$  grid points. Periodic boundary condition is imposed at the two extreme planes along the x-axis and z-axis. The two boundary planes normal to the y-axis represents the walls and hence, have a no-slip boundary condition for velocity and von Neumann boundary condition for pressure.

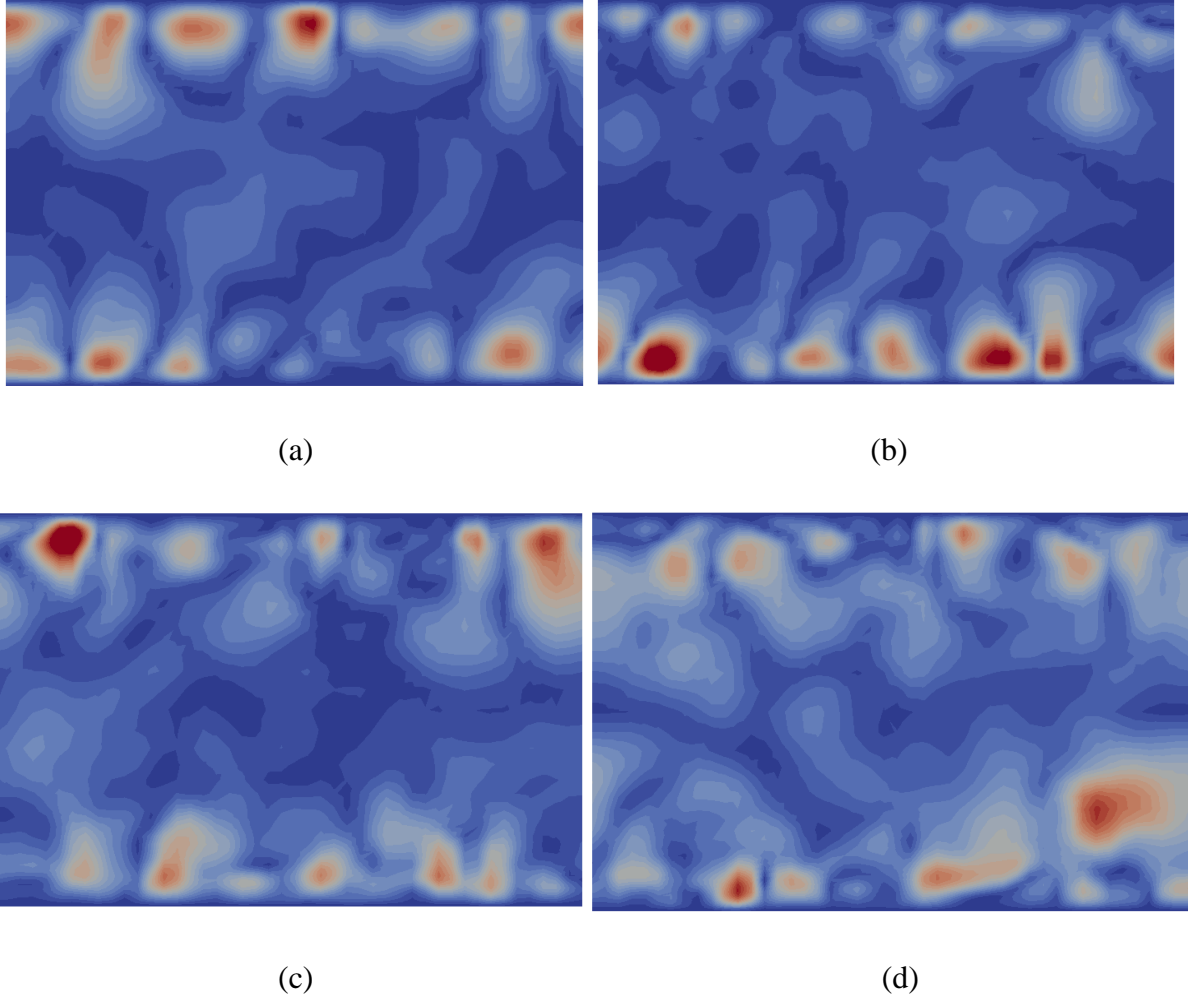
### A.1.5 CHANNEL FLOW POD

The POD is implemented on the velocity database of the mid-plane of the channel flow simulation normal to the streamwise direction. The objective is to develop the temporal dynamics of the POD modes to reconstruct the coherent and incoherent part of the velocity as mentioned in Eqn (34) using LODS and random numbers as discussed in Section A.1.4. respectively. The reconstructed velocity profile will be used as the inlet condition for the main simulation which is the offset jet flow and compared with the existing results of the same, which uses a parabolic profile for the mean component of the inlet velocity and Gaussian random numbers to model the fluctuating component.



**Fig. 5.6** POD normalized eigenvalues of the Channel Flow

The energy content of the POD modes of the channel flow are shown in Fig 5.6. The first 20 modes account for 60.8% of the energy spectrum and the first 50 modes contribute up to 81.7%. Hence, the first 50 modes could be used for the developing the LODS which will represent the coherent part of the flow. The rest of the POD modes can be used for modelling the incoherent part by using random time series as mentioned earlier. Fig 5.7 shows the first 4 POD modes for visualization of the coherent structures in the channel flow.



**Fig. 5.7** The first 4 POD modes of the Channel Flow

The reconstructed velocity needs to be used at the inlet patch of the offset simulation. This can be done by using the ‘timeVaryingMappedFixedValue’ velocity boundary condition [9], which allows reading boundary data from a file. This boundary condition is quite flexible. To extract the values at the face centers from the values at the specified locations, interpolation in space will be applied if necessary. In addition, linear interpolation in time is also supported. A folder named `boundaryData/<patch_name>` has to be created inside the `constant` directory. There should be a file named `points` in the folder. The set of points where boundary data is available is provided in this file. The boundary data is stored in folders with the same name as the time-value associated with it.

#### **A.1.6 PROBLEMS AND FUTURE WORK**

The temporal coefficients of the high energy modes obtained from the POD analysis of the Channel Flow are noisy, making the ODEs developed from the LODS highly stiff in nature. MATLAB has a few in-built stiff ODE solvers including ODE 15s, ODE 23s and so on, but



none of them were able to integrate the system of ODEs of the LODS. As mentioned by a few authors [6], the fourth order Runge Kutta was also coded to solve the same but the values exploded after a few time steps. Three possible factors could be looked into for solving this problem:

- Make use of a more effective stiff ODE solver
- Increase the number of modes considered for developing the LODS as they will capture a higher energy content of the flow
- Use a smaller time step 'deltaT' in the OpenFOAM simulation of the Channel Flow to get a smoother data of the temporal coefficients

Once the LODS is developed, another problem to look into is to make use of a better way to specify the inlet boundary condition for the main offset jet flow simulation. For the 'timeVaryingMappedFixedValue' boundary condition, the velocity data at each time stamp needs to be stored in the boundaryData folder, which might take up a lot of computer memory depending on the duration of the main simulation. However, while reconstructing the inlet data, the temporal coefficients obtained from the LODS and the random numbers for the coherent and incoherent parts respectively need to be multiplied with the same POD modes. Hence the POD modes can be saved once in a file and a code can be scripted to load the POD modes from this file and multiply them with the corresponding temporal coefficient, making it efficient in terms of memory.

## A.2 LODS – LORENTZ SYSTEM TEST CASE – CODE

```

%% Lorentz_profile_generator.m - main script for LODS
%% Generate Lorentz profile
clc,clear;
time_domain = (0:0.001:40);

IC1 = 1;
IC2 = 1;
IC3 = 1;

IC = [IC1 IC2 IC3];

[~, DVsol] = ode45('DEdef', time_domain, IC);

a1_sol = DVsol(:,1);
a2_sol = DVsol(:,2);
a3_sol = DVsol(:,3);

%% random sample selection
sample_points_rand = sort(ceil(rand(99,1)*40000));
sample_points_ind = zeros(100,1);

```



```

sample_points_ind(1) = 1;
sample_points_ind(2:end) = sample_points_rand;

%% plot sample points
clc,clear;
fontsize = 14;
time_domain = (0:0.001:40);
a1_sol = load("data\a1_sol.mat").a1_sol;
a2_sol = load("data\a2_sol.mat").a2_sol;
a3_sol = load("data\a3_sol.mat").a3_sol;
sample_points_ind = load("data\sample_points_ind.mat").sample_points_ind;

figure;
subplot(3,1,1);
plot(time_domain, a1_sol, 'k', 'LineWidth',1);
hold on;
plot(time_domain(sample_points_ind),a1_sol(sample_points_ind),'o');
hold off;
title('x', 'FontName', 'Times New Roman', 'FontSize',fontsize);

subplot(3,1,2);
plot(time_domain, a2_sol, 'k', 'LineWidth',1);
hold on;
plot(time_domain(sample_points_ind),a2_sol(sample_points_ind),'o');
hold off;
title('y', 'FontName', 'Times New Roman', 'FontSize',fontsize);

subplot(3,1,3);
plot(time_domain, a3_sol, 'k', 'LineWidth',1);
hold on;
plot(time_domain(sample_points_ind),a3_sol(sample_points_ind),'o');
hold off;
title('z', 'FontName', 'Times New Roman', 'FontSize',fontsize);
xlabel('time (sec)', 'FontName', 'Times New Roman', 'FontSize',fontsize);

figure;
plot3(a1_sol,a2_sol,a3_sol, 'LineWidth',1);
xlabel('x', 'FontName', 'Times New Roman', 'FontSize',fontsize);
ylabel('y', 'FontName', 'Times New Roman', 'FontSize',fontsize);
zlabel('z', 'FontName', 'Times New Roman', 'FontSize',fontsize);
%% title('Phase Plot', 'FontName', 'Times New Roman', 'FontSize',fontsize)

%% modified a and a_dot
a1_LODS = (a1_sol(sample_points_ind) + a1_sol(sample_points_ind+1))/2;
a2_LODS = (a2_sol(sample_points_ind) + a2_sol(sample_points_ind+1))/2;
a3_LODS = (a3_sol(sample_points_ind) + a3_sol(sample_points_ind+1))/2;
a_LODS = [a1_LODS'; a2_LODS'; a3_LODS'];

dt = time_domain(2) - time_domain(1);

a1_dot_LODS = (a1_sol(sample_points_ind+1) - a1_sol(sample_points_ind))/dt;
a2_dot_LODS = (a2_sol(sample_points_ind+1) - a2_sol(sample_points_ind))/dt;
a3_dot_LODS = (a3_sol(sample_points_ind+1) - a3_sol(sample_points_ind))/dt;
a_dot_LODS = [a1_dot_LODS'; a2_dot_LODS'; a3_dot_LODS'];

%% LODS parameter estimation
clc;
N_tr = 3;
sample_points = 100;

```

```

A_cols = 1 + N_tr + N_tr*(N_tr+1)/2 + N_tr*(N_tr+1)*(N_tr+2)/6;

A_LODS = zeros(sample_points,A_cols);
A_LODS(:,1) = ones(sample_points,1);
col_ind = 2;

for j = 1:N_tr
    A_LODS(:,col_ind) = a_LODS(j,:)' ;
    col_ind = col_ind + 1;
end

for k = 1:N_tr
    for j = 1:k
        A_LODS(:,col_ind) = (a_LODS(j,:).*a_LODS(k,:))';
        col_ind = col_ind + 1;
    end
end

for l = 1:N_tr
    for k = 1:l
        for j = 1:k
            A_LODS(:,col_ind) = (a_LODS(j,:).*a_LODS(k,:).*a_LODS(l,:))';
            col_ind = col_ind + 1;
        end
    end
end

X = zeros(A_cols,N_tr);
for i = 1:3
    B_LODS = a_dot_LODS(i,:)' ;
    X(:,i) = A_LODS\B_LODS;
end
%% Coefficient check
Pr_1 = -X(1,1);
Pr_2 = X(2,1);
R_1 = X(1,2);
a2_dot_a2_coeff = X(2,2);
a2_dot_a1_a3_coeff = X(7,2);
b_1 = -X(3,3);
a3_dot_a1_a2_coeff = X(5,3);

%% Generate time series using LODS
clc;
time_domain = (0:0.001:40);

IC1 = 1;
IC2 = 1;
IC3 = 1;

IC = [IC1 IC2 IC3];

[IVsol, DVsol] = ode45('DEdef_LODS', time_domain, IC);

a1_sol_LODS = DVsol(:,1);
a2_sol_LODS = DVsol(:,2);
a3_sol_LODS = DVsol(:,3);

fontsize = 12;
figure;

```

```

subplot(3,1,1);
plot(time_domain, a1_sol_LODS,'k','LineWidth',1);
title('x LODS','FontName','Times New Roman','FontSize',fontsize);

subplot(3,1,2);
plot(time_domain, a2_sol_LODS,'k','LineWidth',1);
title('y LODS','FontName','Times New Roman','FontSize',fontsize);

subplot(3,1,3);
plot(time_domain, a3_sol_LODS,'k','LineWidth',1);
title('z LODS','FontName','Times New Roman','FontSize',fontsize);
xlabel('time (sec)','FontName','Times New Roman','FontSize',fontsize);

figure;
plot3(a1_sol_LODS,a2_sol_LODS,a3_sol_LODS,'LineWidth',1)
xlabel('x','FontName','Times New Roman','FontSize',fontsize);
ylabel('y','FontName','Times New Roman','FontSize',fontsize);
zlabel('z','FontName','Times New Roman','FontSize',fontsize);

%% DEdef.m - MATLAB function file to solve the original Lorentz system
function Ddv_Div = DEdef(I, D)

Pr = 10;
R = 28;
b = 8/3;

a1 = D(1);
a2 = D(2);
a3 = D(3);

Ddv_Div(1) = -Pr*a1 + Pr*a2;
Ddv_Div(2) = R*a1 - a2 - a1*a3;
Ddv_Div(3) = -b*a3 + a1*a2;

Ddv_Div = Ddv_Div';

end

%% DEdef_LODS.m - MATLAB function file to solve the LODS
function Ddv_Div = DEdef_LODS(I, D)

a1 = D(1);
a2 = D(2);
a3 = D(3);

X = load("data\LODS_parameters.mat").X;

Ddv_Div(1) = X(1,1) + X(2,1)*a1 + X(3,1)*a2 + X(4,1)*a3 + X(5,1)*a1*a1 +
X(6,1)*a1*a2 + X(7,1)*a2*a2 + X(8,1)*a1*a3 + X(9,1)*a2*a3 + X(10,1)*a3*a3 +
X(11,1)*a1*a1*a1 + X(12,1)*a1*a1*a2 + X(13,1)*a1*a2*a2 + X(14,1)*a2*a2*a2 +
X(15,1)*a1*a1*a3 + X(16,1)*a1*a2*a3 + X(17,1)*a2*a2*a3 + X(18,1)*a1*a3*a3 +
X(19,1)*a2*a3*a3 + X(20,1)*a3*a3*a3;

Ddv_Div(2) = X(1,2) + X(2,2)*a1 + X(3,2)*a2 + X(4,2)*a3 + X(5,2)*a1*a1 +
X(6,2)*a1*a2 + X(7,2)*a2*a2 + X(8,2)*a1*a3 + X(9,2)*a2*a3 + X(10,2)*a3*a3 +
X(11,2)*a1*a1*a1 + X(12,2)*a1*a1*a2 + X(13,2)*a1*a2*a2 + X(14,2)*a2*a2*a2 +
X(15,2)*a1*a1*a3 + X(16,2)*a1*a2*a3 + X(17,2)*a2*a2*a3 + X(18,2)*a1*a3*a3 +
X(19,2)*a2*a3*a3 + X(20,2)*a3*a3*a3;

```

```

Ddv_Div(3) = X(1,3) + X(2,3)*a1 + X(3,3)*a2 + X(4,3)*a3 + X(5,3)*a1*a1 +
X(6,3)*a1*a2 + X(7,3)*a2*a2 + X(8,3)*a1*a3 + X(9,3)*a2*a3 + X(10,3)*a3*a3 +
X(11,3)*a1*a1*a1 + X(12,3)*a1*a1*a2 + X(13,3)*a1*a2*a2 + X(14,3)*a2*a2*a2 +
X(15,3)*a1*a1*a3 + X(16,3)*a1*a2*a3 + X(17,3)*a2*a2*a3 + X(18,3)*a1*a3*a3 +
X(19,3)*a2*a3*a3 + X(20,3)*a3*a3*a3;

```

```

Ddv_Div = Ddv_Div';

```

```

end

```

## REFERENCES

- [1] **Weiss, J. (2019).** A tutorial on the proper orthogonal decomposition. In AIAA Aviation 2019 Forum (p. 3333).
- [2] **Kutz, J. N., Brunton, S. L., Brunton, B. W., & Proctor, J. L. (2016).** Dynamic mode decomposition: data-driven modeling of complex systems. Society for Industrial and Applied Mathematics.
- [3] **Brunton, B. W., Johnson, L. A., Ojemann, J. G., & Kutz, J. N. (2016).** Extracting spatial-temporal coherent patterns in large-scale neural recordings using dynamic mode decomposition. *Journal of neuroscience methods*, 258, 1-15.
- [4] **Zdybal, K. (2016).** POD AND DMD DECOMPOSITION OF NUMERICAL AND EXPERIMENTAL DATA.
- [5] **Reynolds, W. C., & Hussain, A. K. M. F. (1972).** The mechanics of an organized wave in turbulent shear flow. Part 3. Theoretical models and comparisons with experiments. *Journal of Fluid Mechanics*, 54(2), 263-288.
- [6] **Perret, L., Delville, J., Manceau, R., & Bonnet, J. P. (2008).** Turbulent inflow conditions for large-eddy simulation based on low-order empirical model. *Physics of Fluids*, 20(7), 075107.
- [7] **Johansson, P. S., & Andersson, H. I. (2003).** Generation of turbulent inlet-boundary conditions using POD-modes. In *Third Symposium on Turbulence and Shear Flow Phenomena*. Begel House Inc..
- [8] **Mukha, T., & Liefvendahl, M. (2015).** Large-eddy simulation of turbulent channel flow (No. 2015-014). Technical Report.
- [9] **Mukha, T. (2017).** eddylicious Documentation.
- [10] **Wu, Z., Laurence, D., Utyuzhnikov, S., & Afgan, I. (2019).** Proper orthogonal decomposition and dynamic mode decomposition of jet in channel crossflow. *Nuclear Engineering and Design*, 344, 54-68.

- [11] **Chiekh, M. B., Michard, M., Guellouz, M. S., & Béra, J. C.** (2013). POD analysis of momentumless trailing edge wake using synthetic jet actuation. *Experimental thermal and fluid science*, 46, 89-102.
- [12] **Lárusson, R., Hafsteinsson, H. E., Andersson, N., & Eriksson, L. E.** (2014). Investigation of supersonic jet flow using modal decomposition. In *20th AIAA/CEAS Aeroacoustics Conference* (p. 3312).
- [13] **Demo, N., Tezzele, M., & Rozza, G.** (2018). PyDMD: Python dynamic mode decomposition. *Journal of Open Source Software*, 3(22), 530.