1 Proper Orthogonal Decomposition[1]

The Proper Orthogonal Decomposition (POD) is a powerful and elegant method of data analysis aimed at obtaining low-dimensional approximate descriptions of high-dimensional processes. The advantage of POD is reducing the degrees of freedom of numerical computational models for time-dependent PDEs, so it alleviates the calculation load and saves CPU computing time and resources for large-scale scientific computing. In this project, we consider a vorticity field of a two-dimensional flow past a cylinder at Reynolds number Re = 200, so our PDE is the Navier-Stocks equation.

1.1 Introduction

We consider generically a system of nonlinear PDEs of a single variable that can be modeled as

$$\mathbf{u}_t = \mathbf{N}(\mathbf{u}, \mathbf{u}_x, \mathbf{u}_{xx}, ..., x, t, \mu) \tag{1.1}$$

where the subscripts denote partial differentiation and $\mathbf{N}(.)$ prescribes the generically nonlinear evolution (Navier-Stokes equation in this project). The computational algorithm for solving PDEs is using standard finite-difference formulas for spatial derivatives, so the PDE transforms into a set of n ODEs,

$$\frac{d\mathbf{u}_t}{dt} = \mathbf{N}(\mathbf{u}(x_{k+1}, t), \mathbf{u}(x_k, t), \mathbf{u}(x_{k-1}, t), ..., x_k, t, \mu) \quad k = 1, 2, ..., n \tag{1.2}$$

Then we can use for example the Runge-Kutta method to solve these ODEs. But for many complex PDEs with several spatial dimensions, the discretization technique can yield millions or billions of differential equations. So we consider a second computational scheme for solving (1.1). In particular, we consider the most common technique for analytically solving linear PDEs: separation of variables.

Taking the idea of separation of variables and considering a set of modes for the spatial part of the solution which underlies the critical ideas of the method of eigenfunction expansions, we have

$$\mathbf{u}(x, t) = \sum_{k=1}^{n} a_k(t)\psi_j(x). \tag{1.3}$$

where $\psi_k(x)$ form a set of $n \gg 1$ basis modes. Notice that increasing the number of modes n is equivalent to increasing the spatial discretization in a finite difference scheme. Inserting this solution into the (1.1) and using the orthonormal property of our basis functions results in the temporal governing equations, or Galerkin projected dynamics, for each mode

$$\frac{da_k}{dt} = \left\langle \mathbf{N}\left(\sum a_j\psi_j, \sum a_j(\psi_j)_x, \sum a_j(\psi_j)_{xx}, ..., x, t, \mu\right), \psi_k \right\rangle \tag{1.4}$$

Convergence to the true solution can be accomplished by both judicious choices of the modal basis elements ψ_k as well as the total number of modes n. A good choice of modal basis elements allows for a smaller set of n modes to be chosen to achieve the desired accuracy. The POD method is designed to specifically address the data-driven selection of a set of basis modes that are tailored to the particular dynamics, geometry, and parameters.

1.2 Optimal Basis Elements: The POD Expansion

As illustrated in the introduction, the selection of a good modal basis for solving (1.1) using the Galerkin expansion in (1.4) is critical for efficient scientific computing strategies. In what follows, we present a data-driven strategy, whereby optimal modes, also known as POD modes, are selected from numerical and/or experimental observations, thus allowing for a minimal number of modes $r \ll n$ to characterize the dynamics of (1.1).

To proceed with the construction of the optimal POD modes, the dynamics of (1.1) are sampled at some prescribed time interval. In particular, a snapshot $\mathbf{u}_k$ consists of samples of the complex system, with subscript k indicating sampling at the time t_k. Now, the continuous functions and modes will be evaluated at n discrete spatial locations, resulting in a high-dimensional vector representation; these will be denoted by bold symbols. We are generally interested in analyzing the computationally or experimentally generated large data set $\mathbf{X}$:

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{u}_1 & \mathbf{u}_2 & ... & \mathbf{u}_m \\ | & | & & | \end{bmatrix} \tag{1.5}$$

Typically $n \gg m$, resulting in a tall-skinny matrix. Many of the properties of POD are inherited from the SVD. So we will discuss it in the following section.

1.3 Singular Value Decomposition

Suppose we have a matrix X that has n rows and m columns. If X is a data matrix (here fluid data) then its entries are real, so the singular value decomposition of X gives

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{1.6}$$

where U is an $n \times n$ orthogonal matrix, V is an $m \times m$ orthogonal matrix, the superscript T indicates matrix transpose and Σ is an $n \times m$ diagonal matrix. The diagonal elements consist of $l = min(n, m)$ nonnegative numbers which are arranged in decreasing order, $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_l \geq 0$. The σ's are called the singular values of X (and also of X^T) and are unique. The rank of X equals the number of nonzero singular values it has. It is important to know that SVD exists for all matrices.

We are interested in a rank-r approximation to the true dynamics where typically $r \ll m$. The quantity of interest is then the low-rank decomposition of the SVD given by

$$\tilde{\mathbf{X}} = \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^T, \tag{1.7}$$

where $\| \mathbf{X} - \tilde{\mathbf{X}} \| < \epsilon$ for a given small value of epsilon. This low-rank truncation allows us to construct the modes of interest ψ_k from the columns of the truncated matrix $\tilde{U}$. In particular, the optimal basis modes are given by

$$\tilde{\mathbf{U}} = \mathbf{\Psi} = \begin{bmatrix} | & | & & | \\ \psi_1 & \psi_2 & ... & \psi_r \\ | & | & & | \end{bmatrix} \tag{1.8}$$

where the truncation preserves the r most dominant modes used in (1.4). The truncated r modes $\{\psi_1, \psi_2, ..., \psi_r\}$ are then used as the low-rank, orthogonal basis to represent the dynamics of (1.1).

The above snapshot-based method for extracting the low-rank, r-dimensional subspace of dynamic evolution associated with (1.1) is a data-driven computational architecture.

1.4 Numerical Results in POD Method

In this project the fluid flow is sampled on a grid 768×192, the domain is $[0, 32] \times [0, 8]$, at a time step of 0.125 seconds, 601 points in the time direction. Each snapshot of the data is stored as a vector of length 147456, which represents a measurement of the 2D flow. By doing singular value decomposition, we can plot the singular values of our data matrix (Figure1). This plot shows

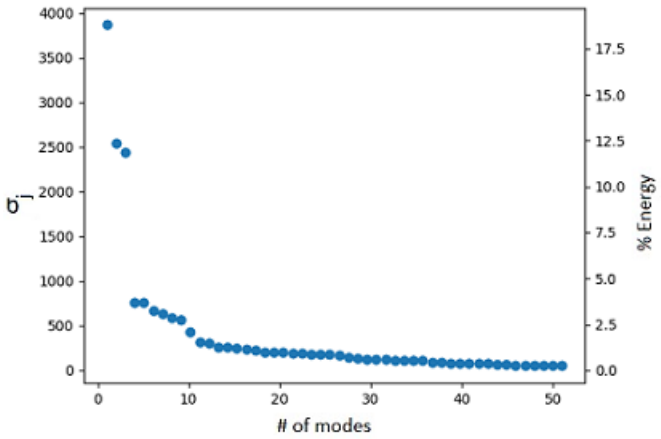

Figure 1: Singular values of the SVD.

that the singular values are arranged in decreasing order and the first mode has 18.8 percent of the total energy. The following table shows the percentage of the cumulative POD mode's energy to the total modes' energy:

# of modes	5	10	20	30	40	50	60
% Energy	50.4	64.4	76.3	84.2	89.1	92.0	94.1

To see how the POD expansion is convergence we can calculate $\| \mathbf{X} - \tilde{\mathbf{X}} \| / \| \mathbf{X} \|$ where $\| . \|$ is the Frobenius norm, $\mathbf{X}$ and $\tilde{\mathbf{X}}$ were defined in equations (1.6) and (1.7), and plot it (Figure 2). In this plot, we can see that by increasing the number of POD modes, the error decreases.

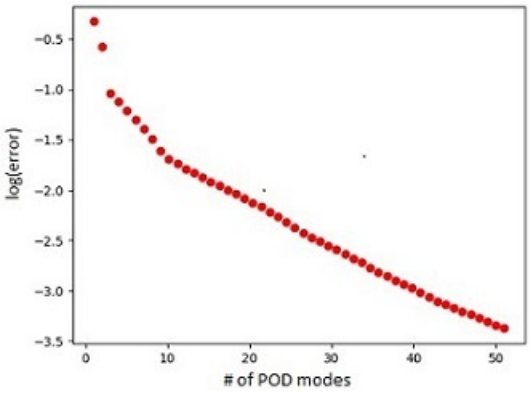

Figure 2: Error in logarithmic scale

Now we visualize some snapshots and their POD expansions. First, we consider the 10th snapshot (Fig.3).

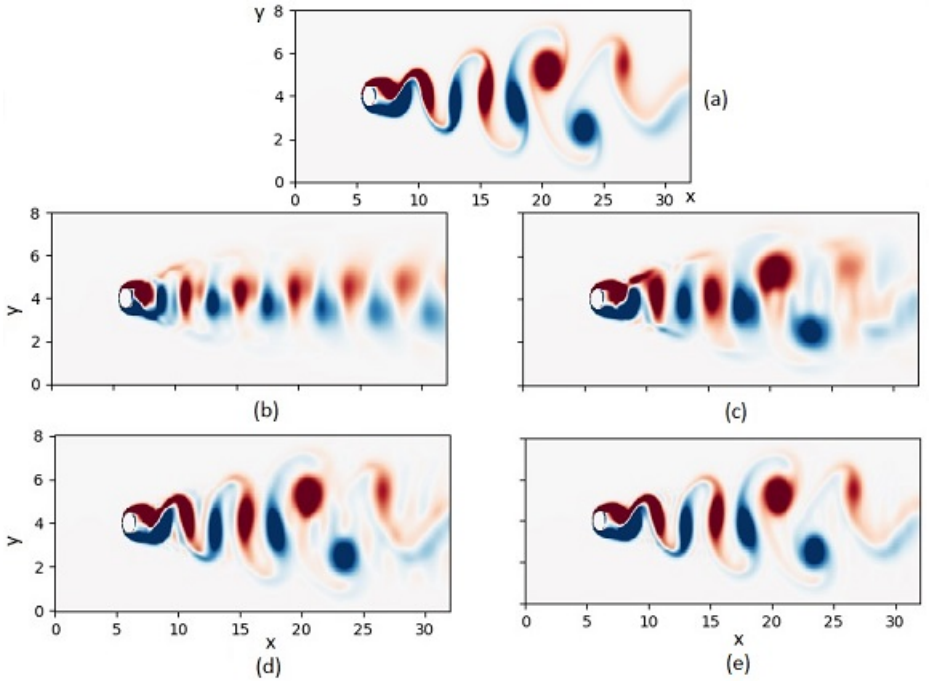

Figure 3: Visualization of the a)10th snapshot and expansions in b)5, c)15, d)30, e)50 POD modes.

We can see how the relative error changes by increasing the number of the POD modes in the expansion for this snapshot (Fig.4). This plot shows that by increasing the number of modes

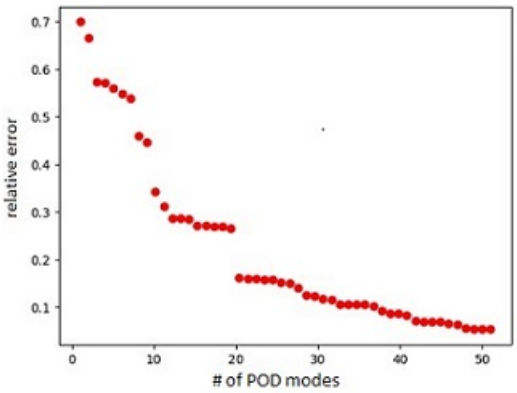

Figure 4: Relative error of POD modes expansion of the 10th snapshot.

in the expansion, the error decreases and for having a relative error smaller than %10, we have to use more than 36 modes in the expansion for this snapshot. We can repeat this for another snapshot, for example for the 200th (Figures 5 and 6). Figure(5) shows that convergency is better for this snapshot, and for having a relative error smaller than %10, we need fewer modes than the previous example (the 10th snapshot). Comparing Figures (4) and (5) shows that the POD modes converge faster for the 200th snapshot than the 10th snapshot. We can see this by comparing the visualization of the expansions of these snapshots as well (Figures 3 and 6).

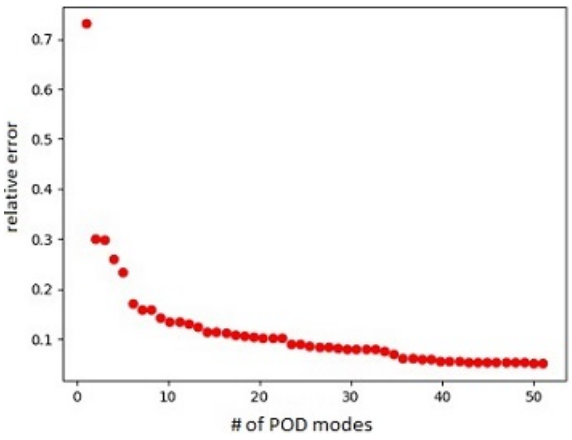

Figure 5: Relative error of POD modes expansion of the 200th snapshot.

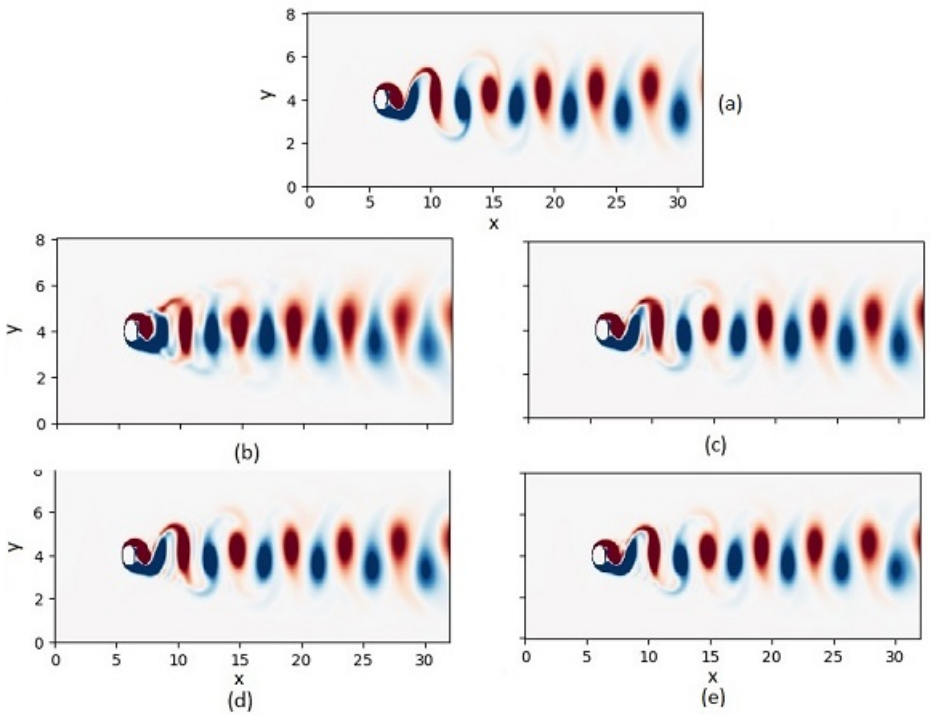

Figure 6: Visualization of a)the 200th snapshot of the vorticity field and c)the 200th snapshot in b)5, c)10, d)15, e)30 POD modes.

2 Dynamic Mode Decomposition[2]

The DMD method originated in the fluid dynamics community as a method to decompose complex flows into a simple representation based on spatiotemporal coherent structures. The growing success of DMD stems from the fact that it is an equation-free, data-driven method capable of providing an accurate decomposition of a complex system into spatiotemporal coherent structures that may be used for short-time future-state prediction and control.

2.1 Formulation of the DMD architecture

We consider generically a dynamic system

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}, t; \mu) \tag{2.1}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is a vector representing the state of our dynamical system at time t, μ contains parameters of the system, and $f(.)$ represents the dynamics. The state $\mathbf{x}$ is typically quite large, $n \gg 1$; this state may arise as the discretization of a partial differential equation at a number of discrete spatial locations, n; as we saw in section (1-1) in equation (1.2).

The DMD framework takes the equation-free perspective, where the dynamics $f(\mathbf{x}, t; \mu)$ may be unknown. The DMD procedure constructs the proxy, approximate locally linear dynamical system

$$\frac{d\mathbf{x}}{dt} = \mathcal{A}\mathbf{x} \tag{2.2}$$

with initial condition $\mathbf{x}(0)$ and the solution:

$$\mathbf{x}(t) = \sum_{k=1}^{n} \phi_k \exp(\omega_k t) b_k = \mathbf{\Phi} \exp(\mathbf{\Omega} t)\mathbf{b}, \tag{2.3}$$

where ϕ_k and ω_k are the eigenvectors and eigenvalues of the matrix $\mathcal{A}$, and the coefficients b_k are the coordinates of $\mathbf{x}(0)$ in the eigenvector basis.

For doing numerical calculations we discretize this equation in time, so we have the analogous discrete-time system sampled every Δt in time:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k \quad where \quad \mathbf{A} = \exp(\mathcal{A}\Delta t). \tag{2.4}$$

The goal is to find eigenvectors and eigenvalues of the linear operator of $\mathbf{A}$. The first step is to collect data and organize them into very tall and skinny matrices ($n \times m - 1$ matrices):

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & ... & \mathbf{x}_{m-1} \\ | & | & & | \end{bmatrix}, \qquad \mathbf{X}' = \begin{bmatrix} | & | & & | \\ \mathbf{x}_2 & \mathbf{x}_3 & ... & \mathbf{x}_m \\ | & | & & | \end{bmatrix} \tag{2.5}$$

The next step is to find a best-fit linear operator $\mathbf{A}$ that maps $\mathbf{X}$ to $\mathbf{X}'$. This matrix is given by

$$\mathbf{A} = \mathbf{X}'\mathbf{X}^{\dagger} \tag{2.6}$$

where † is the Moore-Penrose pseudoinverse. This solution minimizes the error $\| \mathbf{X}' - \mathbf{A}\mathbf{X} \|_F$ where $\| \, . \, \|_F$ is the Frobenius norm, given by

$$\| \mathbf{X} \|_F = \sqrt{\sum_{j=1}^{n} \sum_{k=1}^{m} X_{jk}^2}. \tag{2.7}$$

In practice, when the state dimension(n) is large, the matrix $\mathbf{A}_{n \times n}$ may be intractable to analyze directly. Instead, DMD circumvents the eigendecomposition of $\mathbf{A}$ by considering a rank-reduced representation in terms of a POD-projected matrix $\tilde{\mathbf{A}}$.

2.2 The DMD Algorithm

The DMD algorithm is as follows:

1. First, take the singular value decomposition (SVD) of $\mathbf{X}$: $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$

2. The matrix $\mathbf{A}$ from (2.6) may be obtained by using the pseudoinverse of $\mathbf{X}$ obtained via the SVD:

$$\mathbf{A} = \mathbf{X}'\mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^T \tag{2.8}$$

   In practice, it is more efficient computationally to compute $\tilde{\mathbf{A}}$, the $r \times r$ projection of the full matrix $\mathbf{A}$ onto POD modes:

$$\tilde{\mathbf{A}} = \mathbf{U}^T\mathbf{A}\mathbf{U} = \mathbf{U}^T\mathbf{X}'\mathbf{V}\mathbf{\Sigma}^{-1} \tag{2.9}$$

   The matrix $\tilde{\mathbf{A}}$ defines a low-dimensional linear model of the dynamical system on POD coordinates: $\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{A}}\tilde{\mathbf{x}}_k$. It is possible to reconstruct the high-dimensional state $\mathbf{x}_k = \mathbf{U}\tilde{\mathbf{x}}_k$.

3. Compute the eigendecomposition of $\tilde{\mathbf{A}}$: $\tilde{\mathbf{A}}\mathbf{W} = \mathbf{W}\mathbf{\Lambda}$

   where columns of $\mathbf{W}$ are eigenvectors and $\mathbf{\Lambda}$ is a diagonal matrix containing the corresponding eigenvalues λ_k.

4. Finally, we may reconstruct the eigendecomposition of $\mathbf{A}$ from $\mathbf{W}$ and $\mathbf{\Lambda}$. In particular, the eigenvalues of $\mathbf{A}$ are given by $\mathbf{\Lambda}$ and the eigenvectors of $\mathbf{A}$(DMD modes) are given by columns of $\mathbf{\Phi}$: $\mathbf{\Phi} = \mathbf{X}'\mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{W}$

With the low-rank approximations of both the eigenvalues and the eigenvectors in hand, the projected future solution can be constructed for all time in the future. By first rewriting for convenience $\omega_k = \ln(\lambda_k)/\Delta t$, and using (2.3) the approximate solution at all future times is:

$$\mathbf{x}(t) \approx \sum_{k=1}^{r} \phi_k \exp(\omega_k t) b_k = \mathbf{\Phi} \exp(\mathbf{\Omega} t)\mathbf{b}, \tag{2.10}$$

It only remains to compute the initial coefficient values b_k. If we consider the initial snapshot $\mathbf{x}_1$ at time $t_1 = 0$, then (2.10) gives $\mathbf{x}_1 = \mathbf{\Phi}\mathbf{b}$. The matrix of eigenvectors $\mathbf{\Phi}$ is generically not a square matrix so by using Moore–Penrose pseudoinverse of $\mathbf{\Phi}$, the initial conditions can be found:

$$\mathbf{b} = \mathbf{\Phi}^\dagger \mathbf{x}_1. \tag{2.11}$$

2.3 Numerical Results in DMD

By using (2.9) and finding the $\tilde{\mathbf{A}}$ eigenfunctions, we can reconstruct $\mathbf{A}$. By choosing $r = 30$ we will plot the first 6 DMD modes (Fig.7).

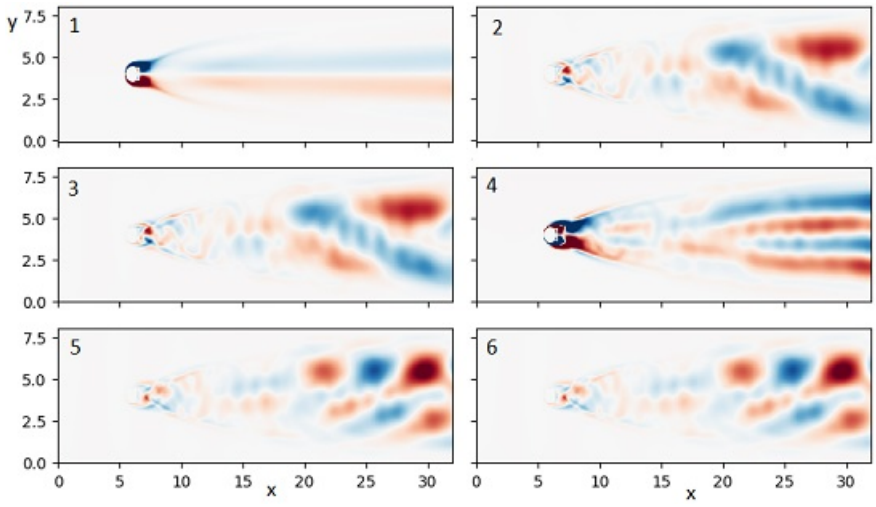

Figure 7: The first 6 DMD modes.

We see in Fig.7 that the modes #2 and #3 and the modes #5 and #6 are the same. By looking at their corresponding eigenvalues, we realize that their eigenvalues are complex conjugates of each other, for example, the eigenvalues of the second and third modes are: $0.99139556 \pm 0.04421244j$. As we can see in these plots (Fig.7/ 2,3,5,6) these modes are showing some vortices and it can mean that they rotate in opposite directions. The modes #1 and #4 are the only modes that have real eigenvalues among the total modes, and they appear once in contrast to the other modes that appear twice.

To see how the DMD expansion converges, we will choose 2 snapshots (10th and 200th) to visualize their expansions in the DMD modes and then plot their relative errors. The plots in Fig.9 and Fig.11 show that DMD modes are convergent. We also see that the expansions in the same number of modes work better for the 10th snapshot that is in the training sample (except for the $r < 10$) than for the 200th snapshot that is outside the training samples.

3 Conclusion

In this project, we observed that the two methods converge by increasing the number of modes. We also saw that the POD method worked better for the snapshots that are later in time. It is probably because the fluid reaches its equilibrium state after a while and most of the snapshots are in this state. But the DMD method worked better for the snapshots that are in the training sample than for those outside the training sample.

The dimensionality of the fluid in DMD or POD methods is based on the balance between accuracy and computational cost. In this project, according to the error plots (Figures 4,5,9,11) for 2 specific snapshots, for having a relative error smaller than 0.1 in general in the POD method, we need to use almost the 40 first modes (this number can be much smaller for the later snapshots

in the POD modes, for example for the 600th snapshot this number is 9). And for the DMD method for having a relative error smaller than %15 in general, we need to use almost the 50 first modes (this number reaches almost 35 modes for snapshots in the training sample, for example, this number is 35 for the 10th snapshot and 36 for the 20th snapshot)

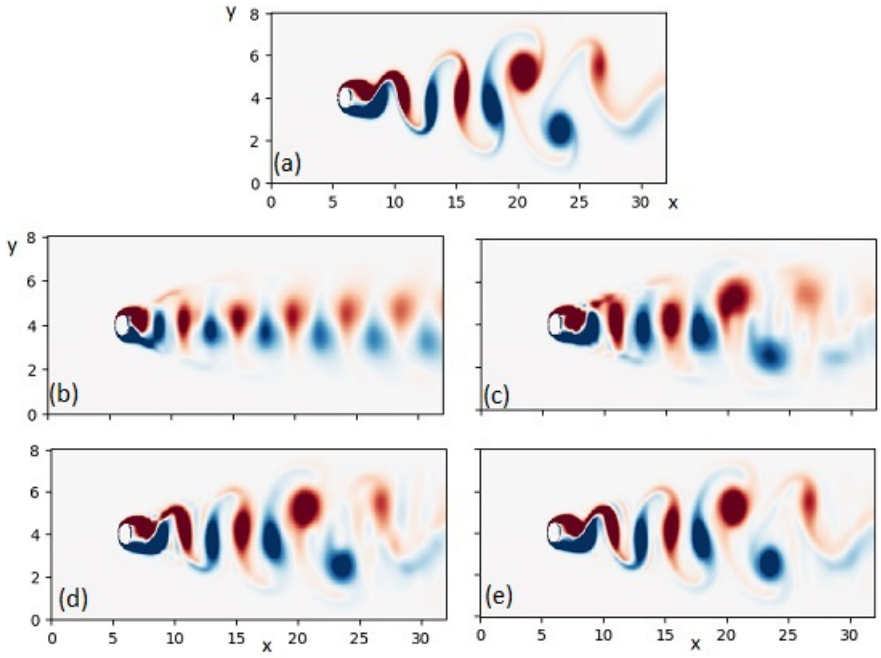

Figure 8: Visualization of the a)10th snapshot and expansions in b)5, c)15, d)30, e)50 DMD modes

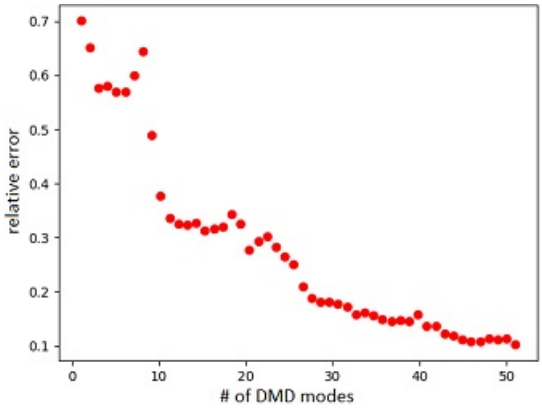

Figure 9: Relative error of the DMD mode expansion of the 10th snapshot.

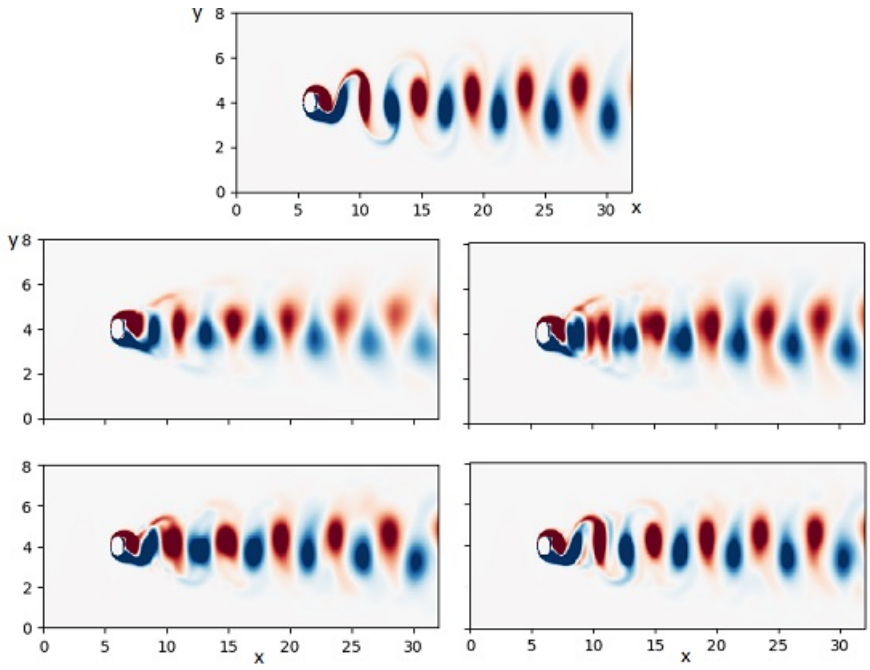

Figure 10: Visualization of the a)200th snapshot and expansions in b)5, c)15, d)30, e)50 DMD modes

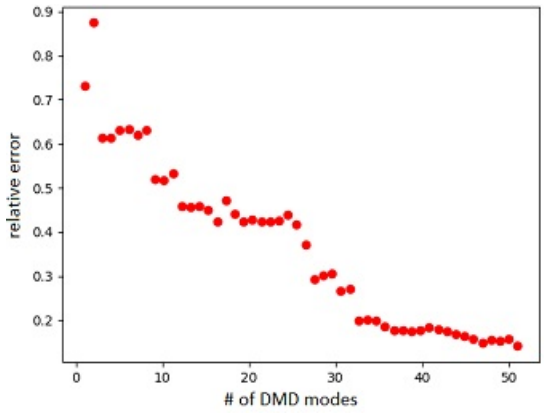

Figure 11: Relative error of the DMD mode expansion of the 200th snapshot.

References

[1] Steven L. Brunton; J. Nathan Kutz (2019) *DATA-DRIVEN SCIENCE AND ENGINEERING; Machine Learning, Dynamical Systems, and Control (chapter 11)*, Cambridge University Press.

[2] J. Nathan Kutz; Steven L. Brunton; Bingni W. Brunton; Joshua L. Proctor (2016) *DYNAMIC MODE DECOMPOSITION, Data-Driven Modeling of Complex Systims*, SIAM.