Data Driven Fluid Mechanics

by

© **Amir Gholizad**

A project submitted in partial fulfillment of the requirements
for the course CMSC 6920

Instructor: Dr. Jahrul Alam

Department of Mathematics and Statistics
Memorial University

April 2023

St. John's, Newfoundland and Labrador, Canada

Abstract

The aim of our project was to investigate the behavior of a fluid by using two techniques called Proper Orthogonal Decomposition (POD) and Dynamic Mode Decomposition (DMD). We focused on examining the vorticity field of a two-dimensional flow past a cylinder at a specific Reynolds number ($Re = 200$). To do this, we collected data about the fluid flow on a 199×449 grid, with each snapshot being taken at a time interval of 0.125 seconds. We represented each snapshot as a vector of length 89351, which captured the flow's characteristics at that moment. Finally, a combination of POD and DMD methods is used to anticipate the future modes of the flow.

Table of contents

Chapter 1

Introduction

1.1 Proper Orthogonal Decomposition (POD)

The Proper Orthogonal Decomposition (POD) method is used in fluid mechanics to analyze data that depends on both space and time, denoted by $U(x,t)$. The method involves separating the variables into spatial modes $\phi_j(x)$ and their corresponding time-varying coefficients (temporal modes) $a_j(t)$, so that $U(x,t)$ can be expressed as a sum of these modes.

To perform POD, the data is assembled into a matrix where each column represents a 'snapshot' of all data measured at a given instance in time and each row consists of all measurements of a given quantity across all times. This matrix is then subjected to a reduced Singular Value Decomposition (SVD) to obtain the spatial modes and their corresponding temporal coefficients.

1.2 Dynamic Mode Decomposition (DMD)

The Dynamic Mode Decomposition (DMD) is a data-driven technique for identifying and analyzing the dominant spatiotemporal patterns and dynamics within high-dimensional, time-varying datasets. It uses the resulting modes and dynamics to reconstruct and predict future states of the system. DMD often incorporates Proper Orthogonal Decomposition (POD) to identify the dominant spatial modes.

Chapter 2

Methodology

2.1 POD [1]

Let's suppose we have an event that depends on both space and time. Such event can be denoted by the function $U(x,t)$. By separating the variables into spatial and temporal modes ($\phi_j(x)$ and $a_j(t)$), this function can be expressed as

$$U(x,t) = \sum_{j=1}^{m} \phi_j(x) a_j(t). \tag{2.1}$$

We can assemble the data related to this event into a matrix where each column represents a 'snapshot' of all data measured at a given instance in time $(t_1, t_2, \ldots, t_m)$, and each row consists of all measurements of a given quantity ($\mathbf{v} = (v_1, v_2, \ldots, v_N)$) across all times. The matrix would look like as

$$U(x,t) = \begin{bmatrix} \mathbf{v}(x_1,t_1) & \mathbf{v}(x_1,t_2) & \cdots & \mathbf{v}(x_1,t_m) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{v}(x_n,t_1) & \mathbf{v}(x_n,t_2) & \cdots & \mathbf{v}(x_n,t_m) \end{bmatrix}. \tag{2.2}$$

The SVD equation of the matrix U reduced to rank r will be

$$rU_{n\times m} = \Phi_{n\times r} \Sigma_{r\times r} \Psi^{\dagger}_{r\times m}, \tag{2.3}$$

where $\Phi \in C^{n\times r}$, $\Psi \in C^{m\times r}$, and $\Sigma \in R^{r\times r}$ is a diagonal matrix with diagonal entries

consisting of the singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$. rU is the refurbished version of original U (for $r = m$ we get $rU = U$).

By comparing Eq.s (2.3) and (2.1) we realize that each column of the matrix U can be interpreted as $U_k = U(x, t_k)$ and therefore

$$\sum_{j=1}^{m} \phi_j(x)\sigma_j\psi_j^\dagger(t_k) = \sum_{j=1}^{m} \phi_j(x)a_j(t_k). \tag{2.4}$$

POD modes are given by the columns $\phi_j(x)$ of Φ, which evolve in time via the corresponding coefficients $a_j(t) = \sigma_j\psi_j(t)^\dagger$. The resulting POD modes are given by the columns of the matrix Φ, which evolve in time via their corresponding temporal coefficients.

2.2 DMD [1]

We can relate each snapshot (spatiotemporal state) of the system to the previous state and write the evolution equation of the system as

$$U_{k+1} = AU_k, \tag{2.5}$$

where A is a linear operator acting on the present state and changing it to the next state. We can create U_k and U_{k+1} by simply taking the first and last $m-1$ columns of U, or

$$U \equiv U_k = U_{n\times[1,m-1]}$$

and

$$V \equiv U_k + 1 = U_{n\times[2,m]}.$$

By solving the Eq. (2.5) for A and replacing U_k (from now on, $U_k = U$ and $U_{k+1} = V$) with its reduced SVD, we get

$$A_{n\times n} = V_{n\times m-1}\Psi_{m-1\times r}\Sigma^{-1}_{r\times r}\Phi^\dagger_{r\times n}. \tag{2.6}$$

Finally we can find the projected version of A into the POD modes of U which is

$$\tilde{A}_{r\times r} \equiv \Phi^\dagger_{r\times n}A_{n\times n}\Phi_{n\times r}, \tag{2.7}$$

thus

$$\tilde{A} = \Phi^{\dagger}_{r\times n} V_{n\times m-1} \Psi_{m-1\times r} \Sigma^{-1}_{r\times r}. \tag{2.8}$$

Solving Eq. (2.6) and identifying the eigenvalues and eigenvectors of the A matrix will be very difficult since it is an $n \times n$ matrix. We can prove that the eigenvalues of the original matrix A is same as the eigenvalues of $\tilde{A}$, or

$$\tilde{A}\omega = R\omega, \tag{2.9}$$

where ω and R are the eigenvalues and eigenvectors of $\tilde{A}$.

In 2014, *Jonathan H. Tu*[2] and his colleagues proved that eigenvectors of the A matrix can be calculated using the following equation

$$\Phi_{DMD} = U_{n\times m-1} \Psi_{m-1\times r} \Sigma^{-1}_{r\times r} R_{r\times r} \tag{2.10}$$

which basically is

$$\Phi_{DMD_{n\times r}} = A_{n\times n} \Phi_{n\times r} R_{r\times r}. \tag{2.11}$$

We can also find the projected modes which are

$$\Phi_p = \Phi_{n\times r} R_{r\times r}. \tag{2.12}$$

The Eq. (2.11) gives us the *Dynamic Modes* of the system.

Chapter 3

Prediction

The approximated differential equation for fluid flow is described as

$$\frac{\partial U(x,t)}{\partial t} = LU, \tag{3.1}$$

which is easy to solve and gives

$$U(x,t) = U_0 e^{(Lt)}, \tag{3.2}$$

or the discrete version

$$U(x, t_k + \Delta t) = e^{(L\Delta t)} U(x, t_k). \tag{3.3}$$

Comparing to Eq. (2.5) we can assume that

$$A = e^{(L\Delta t)}, \tag{3.4}$$

and therefore for the eigenvalues of L we will have

$$\lambda_j = \frac{\log \omega_j}{\Delta t}, \tag{3.5}$$

where ω_j and λ_j are the eigenvalues of A and L.

Knowing that the initial state of the system is the first snapshot, we can say

$$U_{0_{n\times 1}} = \Phi_p b_{r\times 1}, \tag{3.6}$$

where the vector b could be simply obtained by solving the equation.

Using Eq.s (3.6) and (3.5), the time evolution function of the system becomes

$$U(x,t) = \Phi_p b e^{\lambda t}. \tag{3.7}$$

Using Eq. (3.7) and the algorithm in the following chapter, we can predict each state of the system "one by one". Notice that each snapshot (state) belongs to time $t = s\Delta t$ where s is the number (position) of that snapshot.

Chapter 4

Simulation and Results

After loading the data set into our python script and reshaping it, we pick 4 random snapshots and visualize them as below

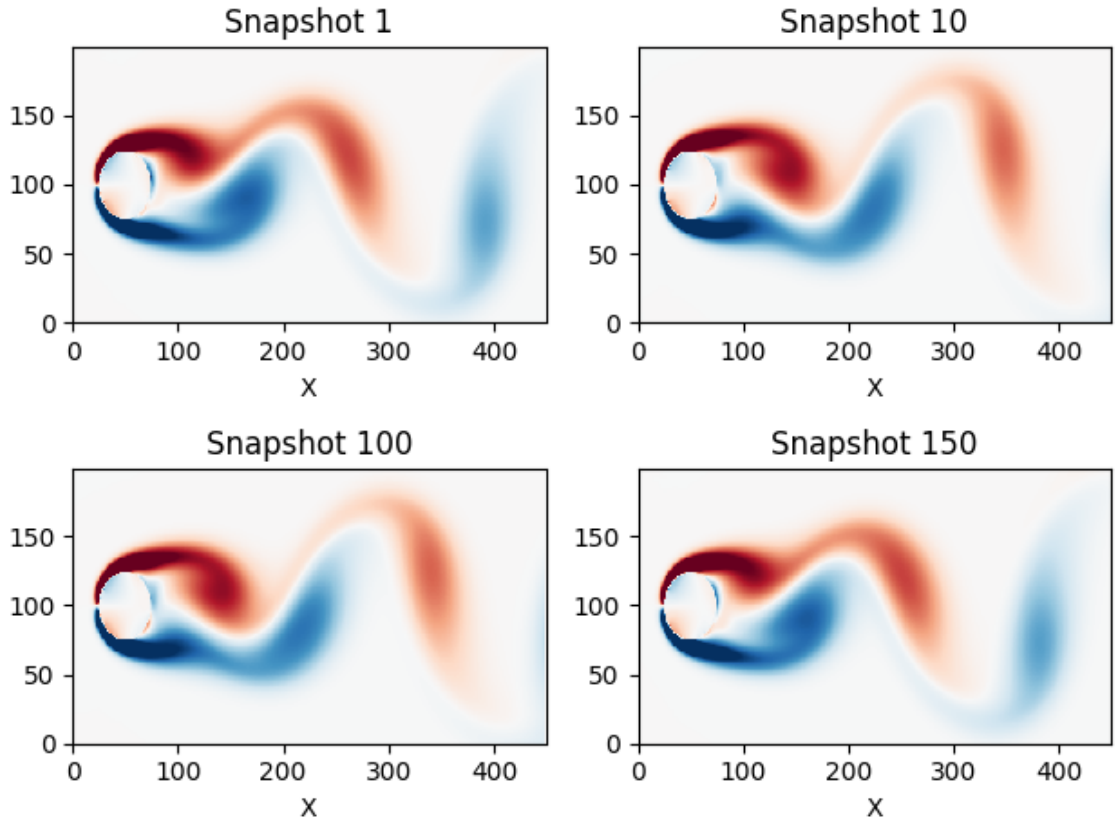
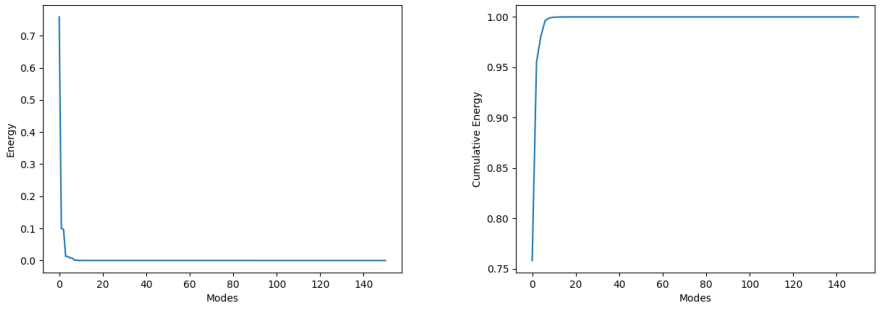


Figure 4.1: 1st, 10th, 100th and 150th snapshots of the fluid past a cylinder

To understand the amplitude and importance of the modes, we define energy for each mode. We assign each diagonal entry of Σ matrix (σ_j) to be interpreted as the energy of a mode. Using this fact we take the squared $L2$ norm of normalized σ_j for the jth mode as below[3]

$$E_j = \| \frac{\sigma_j}{\sum_j^r \sigma_j} \|^2. \tag{4.1}$$

Fig. (4.2)(a) shows the energy of each mode. We can see that energy levels approaches to zero after the first few modes. Fig. (4.2)(b) shows the cumulative sum of energies. The number of modes that hold the 99% of the energy is 6.



(a) (b)

Figure 4.2: (a) Energies and (b) Cumulative Energies of modes

We define a more robust criteria for the convergence of modes. After calculating the energy of all modes (original data), we can find the relative difference between the energy of a mode, before and after the truncation. We call this *Error* and write it as[3]

$$ERR_j = |1 - \frac{e_j}{E_j}| \tag{4.2}$$

where E_j and e_j are the energies of the jth mode before and after the truncation. When it comes to choosing the mode, we choose the first one. Fig. (4.3) shows the Error levels for the POD modes as we increase the truncation number. We can clearly see that for $r \geq 6$ the Error levels drop very fast, meaning that the system with 6 or more number of modes almost converges to the actual system.

After setting the rank to $r = 6$ and truncating the SVD components we generated the refurbished data rU using Eq (2.3). The results for a U vs rU plot is shown in the Fig. (4.4). We achieved 99.82% correlation between these two data sets.

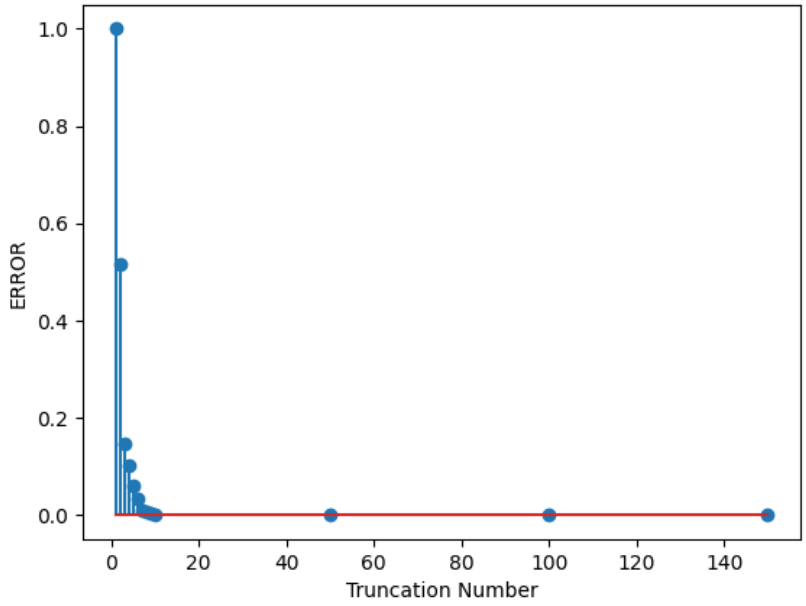


Figure 4.3: Calculated Errors for different truncation numbers (r)

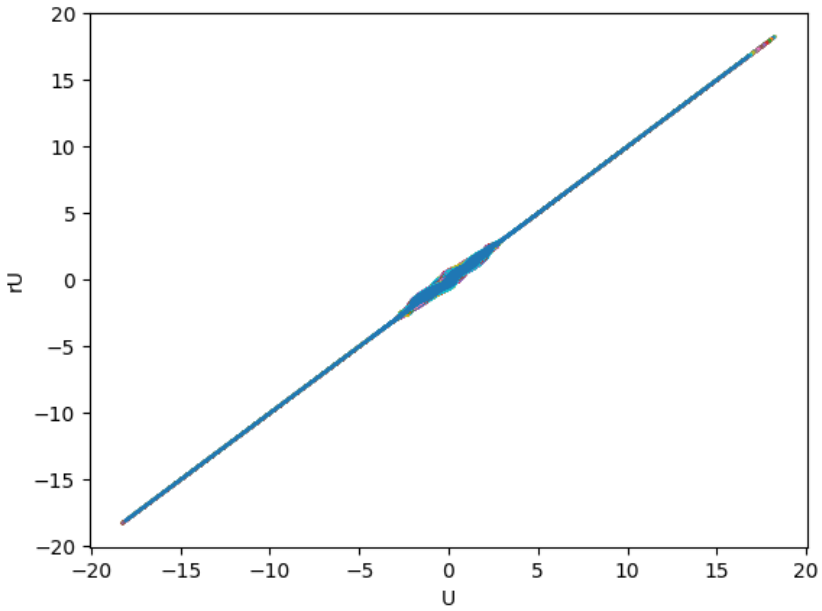


Figure 4.4: U vs rU (Original data vs the refurbished data)

After successfully determining the sampling number r, we set to find the DMD modes. After converting Eq.s (2.10) and (2.12) to python scripts, we obtain the Dynamic Modes. First 6 DMD modes are shown in Fig. (4.5).

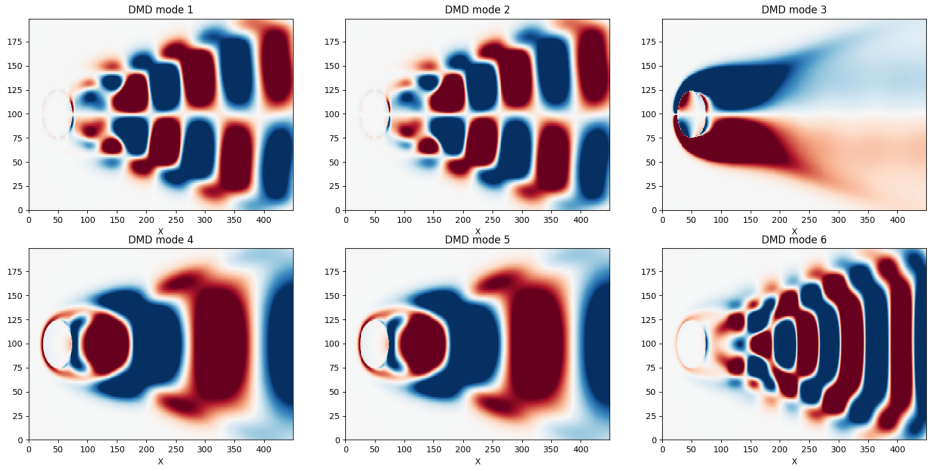


Figure 4.5: Calculated first 6 DMD modes

The second figure shows the comparison of a Dynamic and corresponding Projected mode. We can see that the difference between them is negligible. We can also see a $Re(\omega_j)$ vs $Im(\omega_j)$ plot for the eigenvalues of A matrix in Fig. (4.7). Each dot shows the stability and convergence of the selected mode. Since none of the dots are outside the unit circle, we can state that all modes are stable. Almost all of the 6 DMD modes are on, or near the unit circle, meaning that they are converged. This means:

"The convergence of POD modes, leads to converged DMD modes."

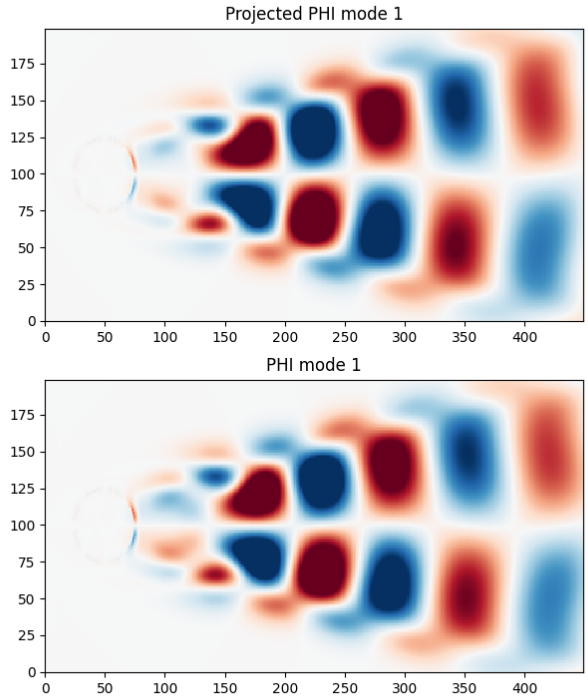


Figure 4.6: Actual vs Projected modes

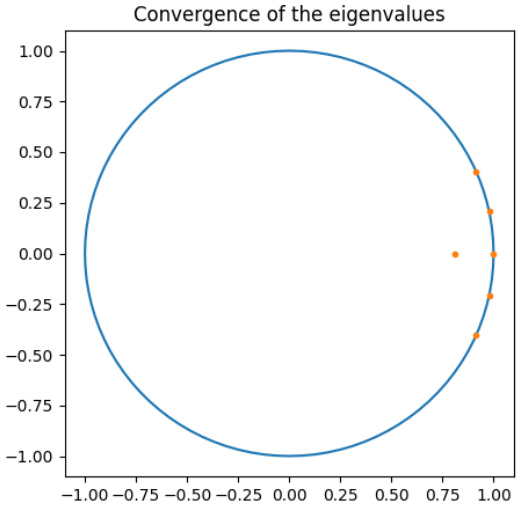
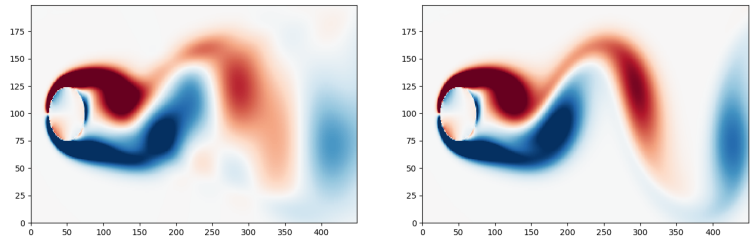
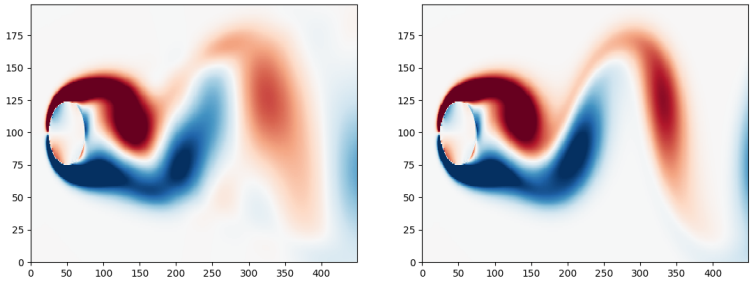


Figure 4.7: Convergence of Main DMD Modes

After solving the Eq. (3.6) for b (using python algorithms), we set the Δt to $0.125s$ and choose two modes from inside ($5 < r = 6$) and outside the training sample ($100 > r = 6$) for data reconstruction. The predictions made by Eq. (3.7) are shown in Fig. (4.8). We achieved 98.6% of correlation for the first pairs of dynamic modes (predicted and actual) and 99.0% for the second pair.



(a) Snapshots from inside the training sample



(b) Snapshots from inside the training sample

Figure 4.8: Predicted vs Actual snapshots

Bibliography

[1] Miguel A. Mendez, Andrea Ianiro, Bernd R. Noack, and Steven L. Brunton, editors. *Data-Driven Fluid Mechanics*. Cambridge University Press, January 2023.

[2] Xuan Dai, Da Xu, Mengqi Zhang, and Richard JAM Stevens. A three-dimensional dynamic mode decomposition analysis of wind farm flow aerodynamics. *Renewable energy*, 191:608–624, 2022.

[3] Jonathan H Tu. *Dynamic mode decomposition: Theory and applications*. PhD thesis, Princeton University, 2013.