MEL20240 - Data Driven Analysis of Fluid Flows

Rutam S. Rajhansa[1], Ojasvi Pandey[1]
[1]*Mechanical Engineering, IITJ*

Abstract

This study aims to analyze the effectiveness of orthogonal decomposition (POD) suitable for manual identification, especially in the presence of noise. This study investigated the effects of different types of noise on POD performance to determine the best type of noise for accurate identification. The main goal is to determine the type of noise that best captures the following information while showing the true impact. Through extensive testing and analysis, measurements such as power correlation and dispersion are used to evaluate the quality of POD representation in different noise environments. In addition, the study also investigates the robustness of POD for different noise levels and its impact on data analysis and interpretation. This shows that by understanding the concepts correctly and using machine learning, we can expand our research and discover new things.

1 Review Of Machine Learning in Fluids

Review Recent progress, in fluid dynamics (CFD) has highlighted the merging of machine learning (ML) methods to improve the efficiency and accuracy of simulations. This review delves into uses of ML in fluid mechanics showcasing key approaches and outcomes detailed in recent studies. A notable advancement in mechanics involves the application of physics guided networks (PGNNs) which embed fundamental physical laws (such as the Navier Stokes equations) into the training phase of neural models. This strategy ensures that ML models not learn from data but adhere to underlying physical principles potentially reducing the volume of data needed for training and enhancing adaptability to new scenarios. Researchers like Raissi et al. (2019) have illustrated the prowess of PGNNs in solving fluid dynamics challenges demonstrating their ability to accurately predict flow patterns and pressure distributions. Machine learning techniques, methods like Orthogonal Decomposition (POD) have played a vital role in refining ROMs for fluid dynamics. ROMs are essential, for simplifying representations of flow thereby cutting down on computational expenses while preserving crucial dynamic features. Guo and Li (2020) used learning to improve the power of POD based ROMs leading to better forecasts of turbulent flows in less time. There is also the use of ML in DNS and turbulence modeling. DNS, although highly accurate, is computationally intensive, due to this

fine resolution because it is required. Studies leading to this breakthrough, e.g. Jeon, Kim (2021), have utilized deep neural networks to simulate reactive flows with orders of magnitude less effort than before. The strategy in the early stages was well-matched against the high-resolution reference data, but it started showing an error increase with time. Different of these studies show the possibility of ML in reducing computational needs but with high fidelity in the simulation. In addition to academic environments, ML has also its place in CFD implementations in industries, mainly through advanced turbulence models and faster solver convergence. For example, Stevens and Colonius (2022) took advantage of the convolutional LSTM network to enhance the precision of the finite-difference/finite-volume method in CFD. This scheme demonstrated its effectiveness in terms of speeding up numerical estimations without affecting precision, which is quite necessary for process modeling when time or resources are an issue. One of the literature findings depicts a rising trend of using ML in many areas in the fluid mechanics. The possible pioneer research could be the extension of the machine learning (ML)-enriched simulation methods to more complicated and extensive geometries and flow configuration. To overcome this challenge, it is imperative to engender more robust models with the ability to handle parameter uncertainties and unsteady situations. Such models are seen as the way forward in the advancement of ML applications in this sector.

2 New Ideas

Based on our study and findings we saw that ML has a huge potential to be incorporated in CFD and other processes

1. Automatic Way of Increasing the Efficiency of Fluid Network Design

Artificial intelligence (AI) could be employed to create more efficient process piping networks self-optimizing themselves, like water supply systems, heating systems, and complex piping in industrial facilities. Through simple learning from pipelines architectures from best cases and system efficiencies, control algorithms could recommend optimum configurations and sizes of pipes, pumps, and valves which have minimal energy consumption and cost while having high flow efficiency.

2. Combined with Machine Learning, for Leak Detection in Fluid Systems

We may develop a machine learning solution that will identify and localize leaks better than existing methods within homogeneous fluid systems (for example, oil pipelines). The system trains models using acoustic, vibration, and pressure sensors data that may pinpoint the subtle anomalies that likely mean a leak and hence it can resource and environmentally damage avert.

3. Increasing Precipitation Accuracy

Machine learning might just offer major improvement in the prediction of sediment transport in rivers and coastal areas by a considerable sum. By making use of actual time data obtained from satellite pictures, water flow gauges,

and weather reports, ML models are now able to forecast changes in sediment tendencies, which is decisive for navigation safety, flood prevention, and conservation of the environment.

4. Flow Modelling With Machine-Learning

Machine learning algorithms can enhance flow information and automatically perform tasks that involve active flow control and optimization. Many Machine Learning techniques, such as Reinforcement Learning can be used to obtain the flow field parameters.

3 Proper Orthogonal Decomposition

3.1 Image Generation

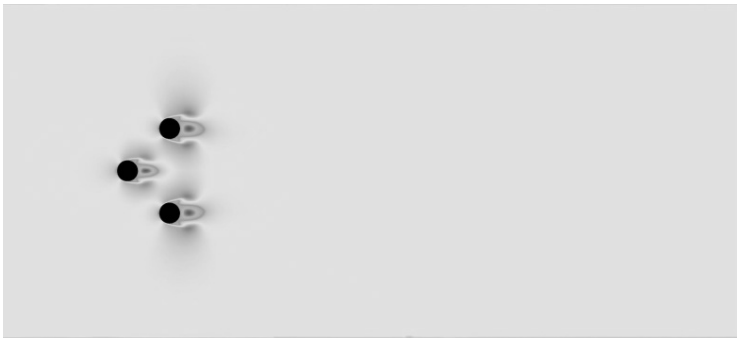


Figure 1: Generated Image

Steps:

1. We first mounted the drive with our Google Colab.
2. Further to perform POD we extracted the frames from the video provided and saved it to a folder in the drive named "output_folder".

Total frames extracted from the video: 751

Problems faced: Due to the database being large, our system crashed several times.

Solution from our side:

- We downsampled the images with the scale factor of 0.5. Although we found that it was still creating problems when we were working on further tasks like applying POD on after adding noises. So we further grayscale the images.

Example of some of the images stored as frames: `https://drive.google.com/file/d/1bNBK9uJilZNVE0G9NwC1jd8Y8wLC3U_1/view?usp=sharing`

3.2 Execute POD

For this, we followed the following steps:

1. Converting the images obtained to stack.
2. Pre-processed the image stack obtained (making it ready for POD).
3. Applying POD using SVD:

The mathematical formulation used in the `compute_pod` function is based on the Singular Value Decomposition (SVD) method. SVD is a matrix factorization technique that decomposes a matrix into three other matrices, namely U, Σ (S), and V^T, where U and V are orthogonal matrices, and Σ is a diagonal matrix containing the singular values.

Reference used to study: `https://www.youtube.com/playlist?list=PLnO2sW0-XPrd_D6lV5YBCaUTa-NsMUibJ`

Here's the mathematical formulation: Given a set of images represented as a stack X where each column vector x_i represents a flattened image, the SVD decomposition is computed as follows:

$$X = U\Sigma V^T$$

Where:

- U is the left singular matrix, containing the left singular vectors. It represents the principal components or the modes of variability in the dataset.
- Σ is the diagonal matrix of singular values, representing the amount of variance captured by each mode.
- V^T is the transpose of the right singular matrix, containing the right singular vectors.

So the function defined `compute_pod` takes a stack of images as input and returns the left singular matrix U, the singular values Σ, and the mean image.

Like this we got all 751 images. The frames were stored in the output folder

3.3 Analyze POD Modes

- For this, we followed the following steps: This step involves analyzing the modes extracted via POD, where each mode represents a significant pattern or feature in the data. The visualization above shows one such mode extracted from our dataset.

After plotting the graph for top 10 modes corresponding to the highest energy states we got the following trend

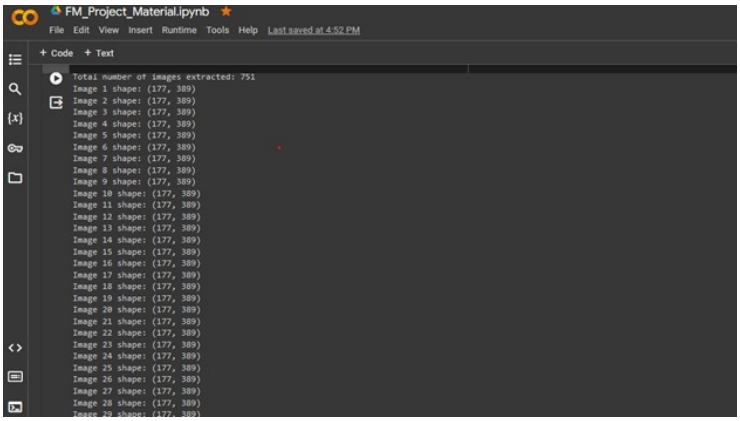


Figure 2

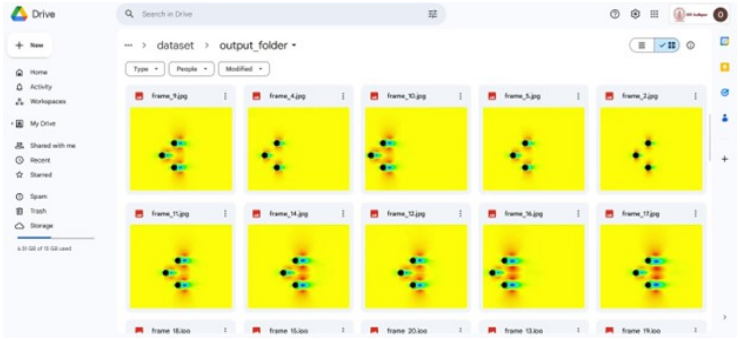


Figure 3: Images

3.4 Significance of Mode

After studying some research papers like:

1. ScienceDirect Article
2. Springer Link Article

we concluded:
These modes are of high significance as:

1. The top modes capture the dominant spatial structures or patterns in the data.
2. These modes represent the most significant modes of variation in the dynamical system.
3. They provide a compact representation of the data and can be used for various purposes such as:

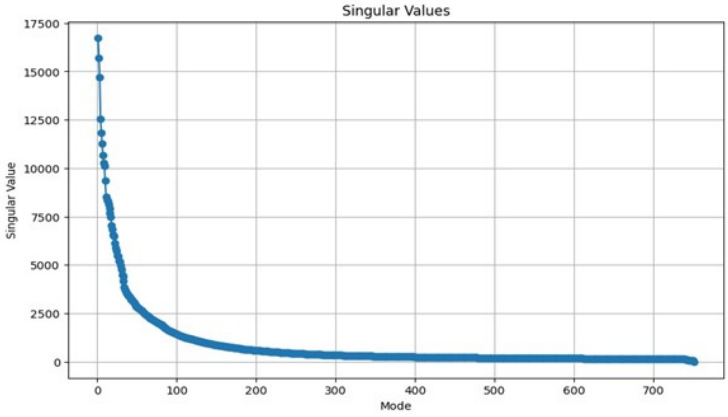


Figure 4: POD Mode Analysis

- Dimensionality reduction
- Reconstruction of original data
- Analysis of flow features, turbulence, or other physical phenomena.

We have tried to depict the significance by reconstructing the original data using these top 10 modes(Fig 6)

We have also plotted a graph showing their contribution (Fig 7)

4 Noise!

4.1 4.1 Adding Noise

Here, we added two types of noises, each with magnitudes of 20%, 40%, 60%, and 80% of the maximum magnitude of the individual frames.

The two types of noises added are:

1. Gaussian Noise
   Reference: `https://wiki.cloudfactory.com/docs/mp-wiki/augmentations/gaussian-noise`
2. Speckle Noise
   Reference: `https://cames.ippt.pan.pl/index.php/cames/article/view/290`

Difference between Gaussian Noise and Speckle Noise:
Reference: `https://medium.com/@sunil7545/speckle-vs-gaussian-noise-7f4f47230d82`

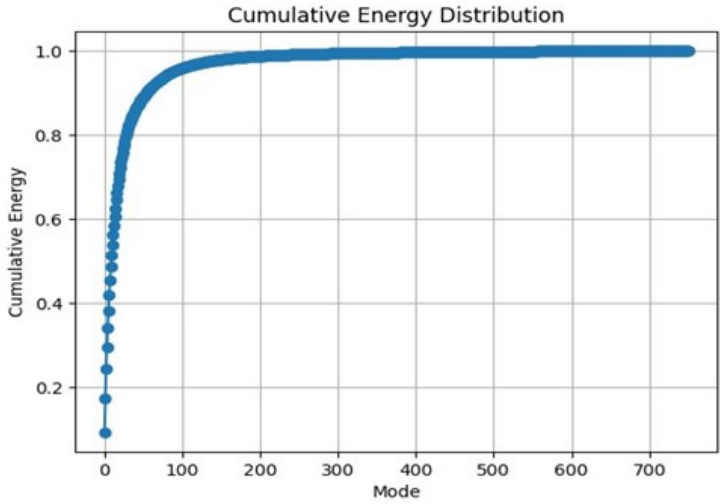


Figure 5

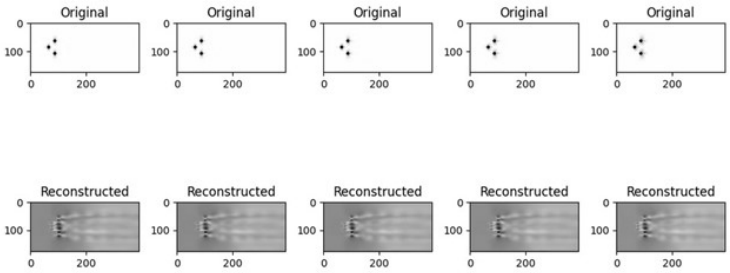


Figure 6: Top 10 Modes

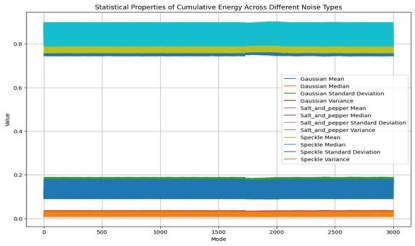
4.2 4.2 Effect on POD Modes (15 marks)

4.2.1 How does the energy transcend through modes?

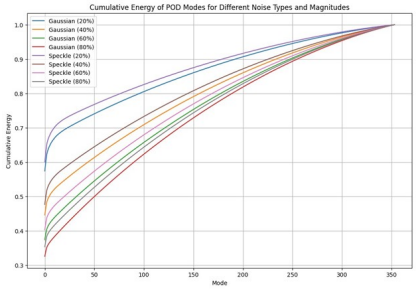
Within the framework of Proper Orthogonal Decomposition (POD), energy is commonly denoted by the singular values derived from Singular Value Decomposition (SVD). Elevated singular values signify modes that encapsulate greater energy or variance within the dataset. Typically, the energy diminishes sequentially as we transition from the primary mode (housing the highest energy) to subsequent modes.

4.2.2 Can you give any statistical evidence of the energy changes in the modes?

For this, we plotted a graph showing statistical properties of Cumulative Energy across Different Noise Types. Additionally, plotting the cumulative sum of



(a) Statistical Properties



(b) Energy of POD Modes

singular values helps visualize the cumulative energy captured by each mode.

4.2.3 Do different noise types have different types of response in terms of how the energy of the modes changes?

From the above graph, we made the following observations:

- Different types of noises have different responses. For the same magnitude, Gaussian noise showed more effect on the data compared to Speckle.

4.2.4 How does the magnitude of noise affect the modal energy?

Elevated levels of noise typically result in more pronounced disturbances in the energy distribution across modes. Stronger noise intensities have the potential to enhance the variability captured by modes with lower energy levels, thereby complicating the extraction of meaningful patterns from the dataset.

4.2.5 What sort of noise is best suited to be used for POD for artificial analysis? Provide your comprehensive analysis

The whole point of adding the noise is to check the robustness of the POD analysis and make as much real noise as possible to simulate real-world noise present in the data. From the graphs obtained, we can conclude that Gaussian noise is best suited to be used for POD for artificial analysis.

- Reasons:
  - Robustness: Gaussian noise models many natural processes and measurement errors, making it a robust choice for simulating real-world conditions.
  - Simplicity: Gaussian noise is easy to generate and add to datasets. Its simplicity makes it a convenient choice for conducting experiments and analyzing algorithms.
  - Statistical Properties: Gaussian noise is well-studied and its statistical properties are well-understood. This makes it easier to analyze and model mathematically.

5 Super-Resolving

Using the ML model provided in code, we attempted to remove noise.

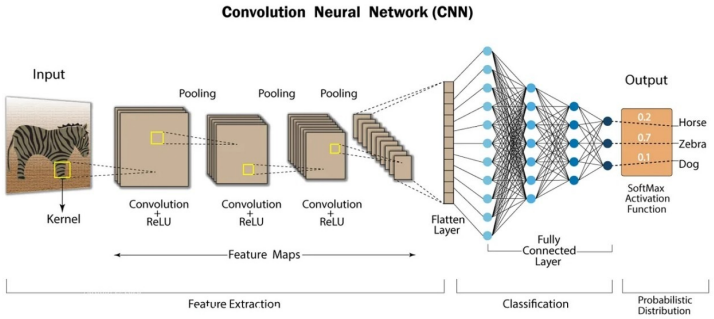


Figure 8

5.1 Problems Faced

The system crashed due to a large dataset.

Solution: We applied it on a smaller subset of the dataset. We set the epochs to 10 and then trained the model.

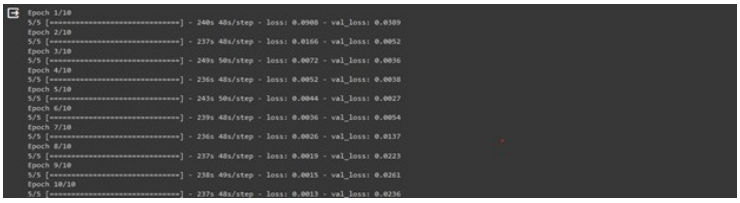


Figure 9

6 References

- *Sciencedirect Article:* Link
- *ResearchGate Paper:* Link
- *Google Colab Notebook:* Link
- *YouTube Playlist:* Link

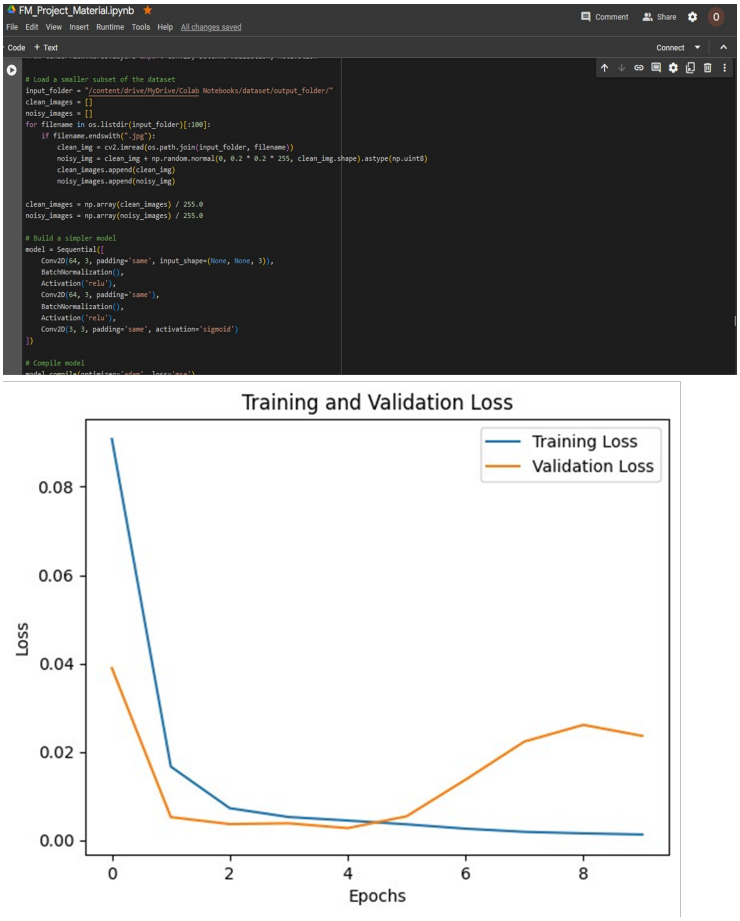


Figure 10

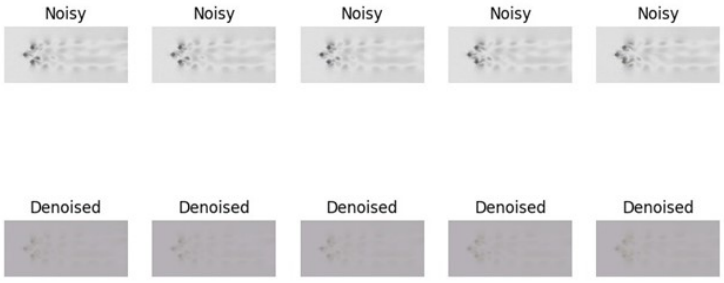


Figure 11