Data-Driven Analysis of Fluid Flows

Pushkin Dugam$^{\mathrm{B22ME052}}$ **and Gopala Ram Jyani**$^{\mathrm{B22ME075}}$

[*]Mechanical Department, IIT Jodhpur
[*]pushkindugam@gmail.com

ABSTRACT

This project delves into the application of machine learning (ML) techniques in the domain of fluid mechanics, focusing on the utilization of proper orthogonal decomposition (POD) for analysis and simulation.Firstly, a comprehensive literature review highlights recent advancements in ML and showcases how ML methods are revolutionizing traditional approaches by harnessing vast amounts of data to capture complex flow dynamics accurately.Following this, four novel ideas are proposed for applying ML in fluid mechanics.

Subsequently, the mathematical formulation and execution of POD on video frames depicting flow over three cylinders are described. The energy distribution among the modes and the significance of the top 10 modes are analyzed based on research articles utilizing POD.The project further explores the impact of noise on modal studies by artificially adding noise to the original frames and analyzing its effects on the POD modes. Statistical evidence and insights into the response of different noise types and magnitudes on modal energy are provided.Finally, data-driven methods, including ML and deep learning algorithms, are employed for super-resolving noisy images to compare with true images. The abstract summarizes the project's findings and contributions to understanding and enhancing fluid mechanics analysis through ML-driven approaches.

1. Literature Review: Machine Learning Advancements in Computational Fluid Dynamics

Machine learning (ML) techniques have demonstrated significant potential for enhancing computational fluid dynamics (CFD) simulations, particularly in the realms of turbulence modeling, large-eddy simulation (LES), and wall-model development. This literature review provides an overview of recent advancements and key findings in these areas:

Turbulence Modeling with ML:

ML methods offer avenues for improving turbulence modeling in CFD simulations. Researchers like Duraisamy et al. have explored the potential of ML to enhance Reynolds-averaged Navier-Stokes (RANS) models. Traditional RANS models rely on eddy-viscosity approaches, which may not adequately capture complex flow phenomena. ML-based approaches, such as those proposed by Ling et al., embed Galilean invariance in deep learning architectures to predict important components like the anisotropy tensor, leading to more accurate predictions compared to conventional methods. Additionally, Wu et al. have introduced frameworks that combine physical knowledge with deep-learning architectures to predict linear and nonlinear parts of the Reynolds-stress tensor separately, offering improved interpretability and performance in various flow configurations.

Advancements in LES Using ML:

ML techniques have also shown promise in improving large-eddy simulation (LES) by developing subgrid-scale (SGS) models. Beck et al. and Lapeyre et al. have proposed CNN-based approaches to enhance SGS models, leveraging spatial correlations present in turbulent flows. While these methods typically rely on high-fidelity data for training, Novati et al. introduced a reinforcement learning (RL) approach that does not require reference data. This unsupervised method can develop SGS models consistent with the coarse simulation being performed, potentially improving generalizability across a wider range of cases.

Wall-Model Development and Boundary Conditions:

ML techniques offer innovative solutions for wall-model development, particularly in high-Reynolds-number applications where traditional methods may face challenges. Bae and Koumoutsakos proposed a reinforcement learning-based approach to determine boundary information close to the wall, addressing uncertainties in setting boundary conditions. Deep learning methods also facilitate the mapping of flow characteristics near the wall, either through direct prediction or by determining virtual velocities. Despite challenges related to boundary conditions over rough surfaces, progress has been made using data-driven methods that incorporate physical insights.

By leveraging ML techniques, researchers aim to improve accuracy, efficiency, and generalizability in simulating complex fluid flow phenomena, paving the way for advancements in various engineering and scientific applications.

2. Innovative Insights

A] Smart Drilling Hydraulics Optimization System

This innovative system aims to enhance drilling efficiency, minimize risks, and maximize productivity by optimizing drilling hydraulics parameters.(Pump Pressure, Flow Rate, Nozzle Size and Configuration, Fluid Density, Fluid Rheology)

Key Features:

1. Real-time Data Analysis: The system continuously monitors drilling parameters, fluid properties, and wellbore conditions using advanced sensors and data acquisition systems.

2. Predictive Analytics: Utilizing machine learning algorithms, the system predicts key parameters such as pump pressure, standpipe pressure, and pressure loss, enabling proactive adjustments to optimize drilling performance.

3. Adaptive Control: Based on predictive analytics and real-time feedback, the system autonomously adjusts drilling hydraulics parameters such as flow rates, nozzle sizes, and pump speeds to maintain safe operating conditions and maximize the rate of penetration (ROP).

4. Cuttings Transport Optimization: By analyzing cuttings concentration, particle properties, and wellbore geometry, the system optimizes cuttings transport efficiency, reducing the risk of borehole instability and improving overall drilling performance.

5. Decision Support System: Integrated with a user-friendly interface, the system provides actionable insights and recommendations to drilling engineers and operators, facilitating informed decision-making and operational planning.

Benefits:

- Enhanced Safety: By maintaining safe drilling pressures and minimizing risks of borehole collapse or fracturing, the system enhances safety for personnel and equipment.

- Improved Efficiency: Optimizing drilling parameters and cuttings transport reduces non-productive time (NPT) and enhances drilling efficiency, resulting in cost savings and increased well productivity.

Conclusion:

The Smart Drilling Hydraulics Optimization System represents a paradigm shift in drilling operations, combining advanced data analytics, machine learning, and real-time control to optimize performance, enhance safety, and maximize productivity. Embracing this innovative solution promises to revolutionize the oil and gas industry, paving the way for smarter, more efficient drilling practices.

B] Integrated Wave and Tidal Energy Forecasting Platform

The Integrated Wave and Tidal Energy Forecasting Platform revolutionizes renewable energy generation, maritime operations, and coastal infrastructure design by leveraging advanced machine learning and optimization algorithms. By accurately predicting wave energy fluxes, power generation from wave energy converters, and tidal currents in real-time, it facilitates informed decision-making and enhances operational efficiency.

Key Features:

1. Multi-Model Fusion: Integrates various machine learning techniques, including artificial neural networks (ANN), long short-term memory (LSTM) networks, support vector machines (SVM), and Gaussian processes, for accurate forecasting of wave and tidal energy fluxes, and tidal currents.

2. Optimized Model Training: Utilizes advanced optimization algorithms such as moth-flame optimization and Bayesian machine learning to optimize neural network models, ensuring efficient data utilization and enhancing prediction accuracy.

3. Utilization of Experimental and Numerical Data: Incorporates experimental data from wave energy converter tests and numerical data from computational fluid dynamics simulations for robust model development and validation across diverse oceanic conditions.

4. Time-Efficient Calibration: Implements novel approaches like time-efficient calibration using artificial neural networks to reduce computational time in simulating wave interaction with porous structures, enhancing the efficiency of wave parameter predictions.
5. Real-Time Tidal Current Prediction: Utilizes Bayesian machine learning techniques, including Gaussian processes, for real-time prediction of tidal currents, incorporating data from NOAA observation sites to accurately predict currents' speed and direction, enabling safer navigation and efficient utilization of tidal energy resources.

Benefits:

- Enhanced Prediction Accuracy: Integration of multiple machine learning techniques and optimization algorithms improves the accuracy and reliability of wave and tidal energy forecasts, facilitating informed decision-making for energy generation and maritime operations.
- Efficient Resource Utilization: Utilization of experimental and numerical data, along with time-efficient calibration methods, optimizes resource utilization and reduces computational time, leading to cost savings and increased productivity.
- Real-Time Decision Support: Real-time tidal current prediction enables timely decision-making for maritime navigation, offshore operations, and renewable energy generation, enhancing safety and operational efficiency.

Conclusion:

The Integrated Wave and Tidal Energy Forecasting Platform provides a comprehensive solution for predicting wave and tidal energy resources and oceanic conditions. Leveraging advanced machine learning techniques and optimization algorithms, the system offers accurate forecasts, optimizes resource utilization, and supports real-time decision-making for energy generation, maritime operations, and sustainable coastal infrastructure development.

C] Ionic Liquids: Numerical and Analytical Machine Learning Approaches

Ionic liquids (ILs) are garnering attention as promising candidates for energy storage systems due to their advantageous characteristics, such as low volatility, high thermal stability, and adjustable conductivity. However, optimizing IL formulations to meet specific performance criteria remains a complex and time-consuming endeavor.

To address this challenge, a novel approach leveraging machine learning (ML) techniques can expedite the optimization process. By training ML models on a comprehensive dataset encompassing IL properties and molecular structures, the system can predict the performance of novel IL formulations based on varying compositions and environmental conditions.

ML algorithms, including regression and neural networks, excel in unraveling intricate relationships between IL attributes and energy storage properties. Through meticulous analysis, these models facilitate the identification of optimal IL formulations tailored to meet precise application requirements. Moreover, by integrating domain knowledge and physical constraints, ML models can guide the exploration of promising IL candidates efficiently.

The significance of ILs in energy storage applications cannot be overstated, given their versatility across diverse technologies like lithium-ion batteries, supercapacitors, and fuel cells. By employing ML-assisted optimization, researchers can surmount existing challenges and unlock the full potential of IL-based energy storage systems, ushering in a new era of efficiency, sustainability, and cost-effectiveness.

References

- https://www.sciencedirect.com/science/article/pii/S1270963823002511#se0130
- https://www.sciencedirect.com/science/article/pii/S1875510022003936#sec4
- https://arxiv.org/pdf/2109.05574.pdf
- https://www.mdpi.com/2311-5521/7/10/321

3. Proper Orthogonal Decomposition

0.1 Image Generation

- Given image is RGB image.
- Image size is (708,1558).
- Image has three chanel

0.2 Execute Proper Orthogonal Decomposition

- **Step 1.** Compute the velocity field from the image using videos of the flow.

  1. firstly convert image in to numpy array
  2. Flattern the numpy array.
  3. Make the velocity field(U) using all image of the vidios.

- **Step 2.** compute the covariance matrix of the velocity field data set.

  1. covarian matrix = (1/m-1)dot(U.T,U)
  2. m is number of image in the velocity field U.
  3. U.T transpose of the matrix U.

- **Step 3.**. Calculate the eigenvalues and eigenvector form the covariance matrix.

  1. E:- eigenvector matrix of the covarance matrix
  2. dimension of the eigenvector matrix is m

- **Step 4.** Re-arrenge the eigenvector decending order of the eigenvalues.
- **Step 5.** Project all the velocity field on the eigenvector.

  1. A = U*E
  2. A : Projected velocity on the eigenvector

- **Step 6.** Calculate the top 10 energy mode.

  1. mode = U*(A.T)
  2. mode[i] = first column of the mode
  3. i is number of the mode (i=1,2,3,4,5)

- **Step 7.** Compute the mean of the energy mode.

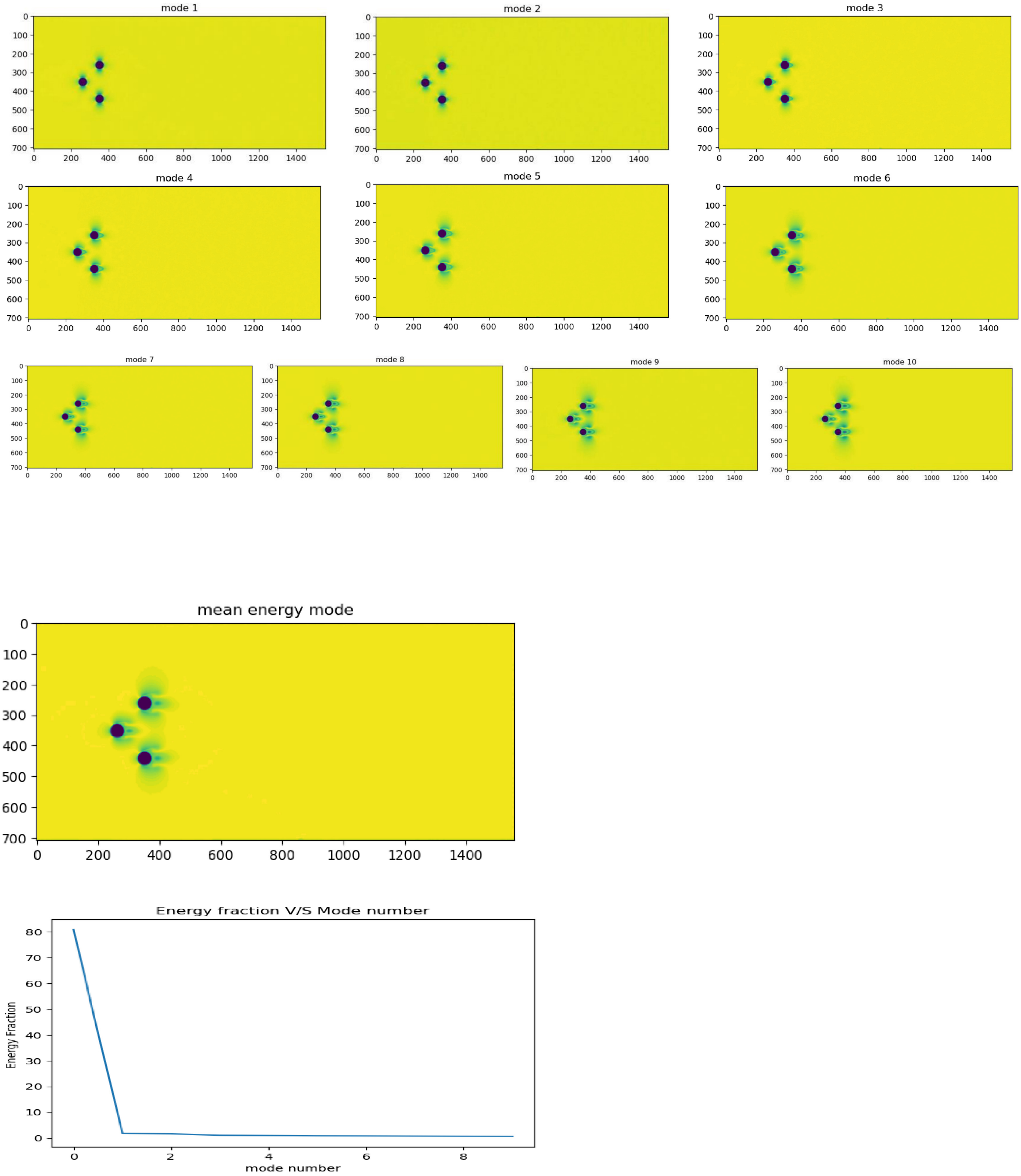
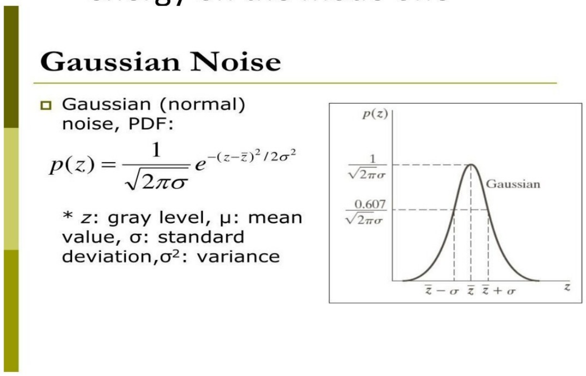


Figure 1. POD analysis image

Adding Noise: Gaussian Noise

We are add gaussian noise with different std value/magnitude of noise w.r.t the original image.

According our analysis when we are increase the magnitude of noise then decrease the energy on the mode one



Gaussian Noise

- Gaussian (normal) noise, PDF:

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\bar{z})^2/2\sigma^2}$$

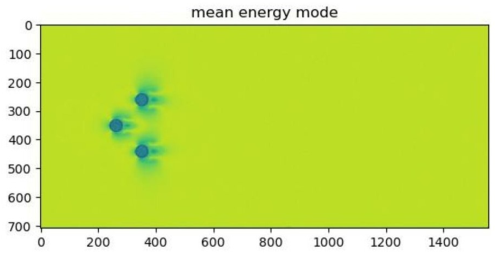
* z: gray level, μ: mean value, σ: standard deviation,σ²: variance

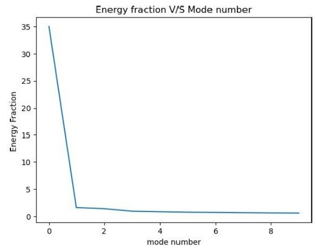
$p(z)$
$\frac{1}{\sqrt{2\pi}\sigma}$
$\frac{0.607}{\sqrt{2\pi}\sigma}$
Gaussian
$\bar{z}-\sigma$ $\bar{z}$ $\bar{z}+\sigma$
z

The magnitude of Gaussian Noise depends on the Standard Deviation(sigma). Noise Magnitude is directly proportional to the sigma value.

20% noise

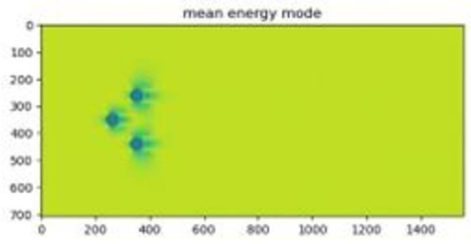
Std value after adding noise is :21

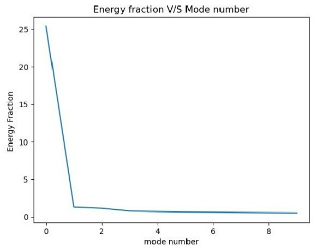




40% noise

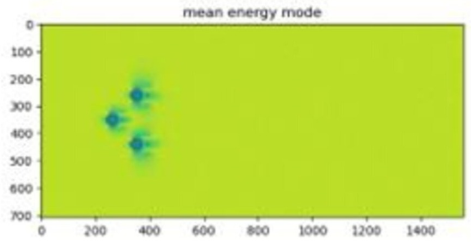
Std value after adding noise is 32

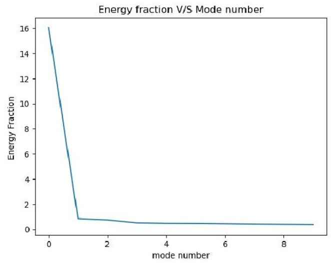




60% noise

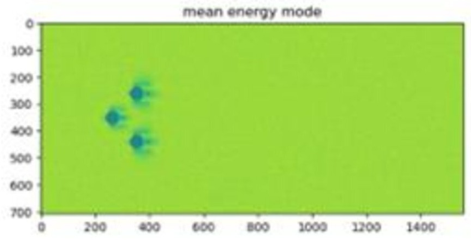
Std value after adding noise is 41

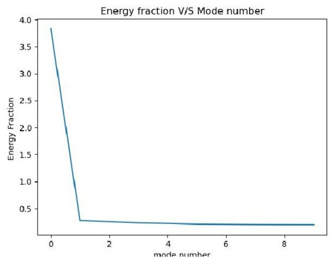




80% noise

Std value after adding noise is 49





Adding Noise: Poisson Noise

1. Second we are add poisson noise in the image and then apply proper orthogonal decoposition
2. Poisson distribution function is

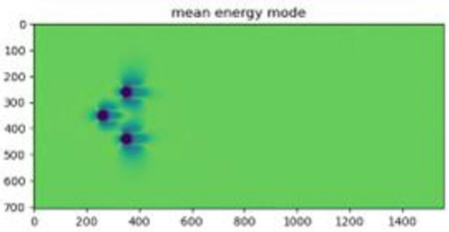
   P(k) = (lemda^k*exp(-lemda))/(k!)

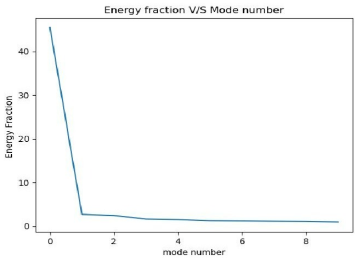
   P(k) is is the probability of observing *k* events,

   Lemda is mean of the distribution

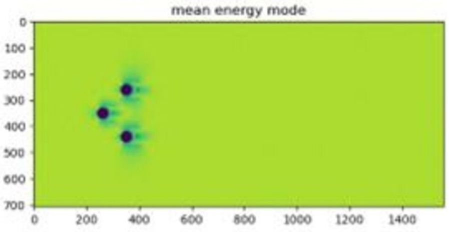
   K represents the number of events or occurrences

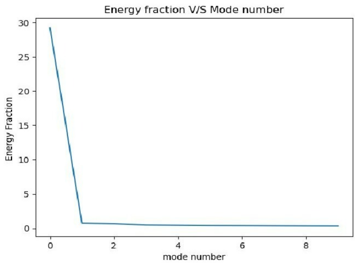
20% Noise



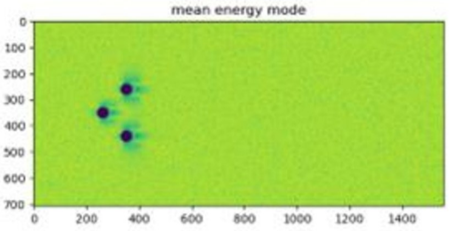


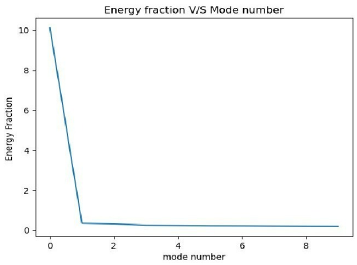
40% Noise





60% Noise





4.2 Effect on POD Modes

NOTE : *We have performed POD on these noisy images but have refrained from putting them in this report to abide by the page limitations.*

Q] How does the energy transcend through modes?
According the origin image the first mode energy is very high 80 percentage energy is distribute in the first mode and the 20 percentage energy is remaining mode .After adding 20% gaussian noise the first mode has 38% energy ,40% gaussian noise the first mode has 27% energy ,similarly for 60% noise - 15% energy and 80% noise - 5% energy.

Noise may result in a decrease in the energy fraction of the top modes. This could happen if the noise introduces disturbances that disrupt the coherent structures represented by these modes, thereby reducing their ability to capture the dominant features of the flow field. As a result, the energy fraction associated with the top modes may diminish relative to the noise-free condition.

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

The graphs for energy fraction for each modes after adding different magnitudes of sound have been showed above.

Q] Do different noise types have different type of response in terms of how the energy of the modes changes?

the response of mode energy distribution to different noise types is complex and depends on a variety of factors including std value .... But according my result more energy decerease in the gaussian noise compare possson noise . This observation suggests that the Gaussian noise, with its continuous and symmetric distribution, may have a more pronounced impact on the dominant modes captured by the system compared to the Poisson noise.

Q] How does the magnitude of noise affect the modal energy?
=> Increasing the magnitude of noise leads to more pronounced disturbances in the system, resulting in an overall increase in the energy content of the modes particularly higher-order ones......and according to our results the energy fraction in the top modes have decreased which justifies the above statement . This increase reflects the influence of noise-induced fluctuations on the flow field. Additionally, higher levels of noise cause a shift in the energy distribution among modes, with the energy fraction associated with higher-order modes increasing more rapidly compared to lower-order modes. This shift occurs because noise perturbations predominantly affect finer-scale structures captured by higher-order modes.

Q] What sort of noise is best suited to be used for POD for artificial analysis?
=> Gaussian noise provides a versatile and realistic means of introducing artificial disturbances into the flow field for POD analysis. Its statistical properties, additivity, randomness, and ease of generation make it well-suited for studying the sensitivity of POD modes to noise and evaluating the robustness of the analysis technique.

5. Super Resolving

A] OpenCV (Non-local Means denoising)

Non-local Means Denoising (NLM) is a powerful technique used for image denoising in computer vision and image processing. It's particularly effective for reducing Gaussian noise while preserving image details and edges.

- **Basic Idea of Non-local Means Denoising:**

  In mathematical terms, we aim to find an estimate $\hat{f}(x)$ for each pixel x in the image. Instead of directly averaging pixel values within a local window $W(x)$, NLM computes the denoised estimate using similar patches across the entire image. Mathematically, this can be expressed as:

  $$\hat{f}(x) = \frac{1}{Z(x)} \sum_{y \in \Omega} w(x,y) \cdot \hat{f}(y)$$

  where $Z(x)$ is the normalization factor, Ω is the entire image domain, $w(x,y)$ are the weights based on patch similarity, and $\hat{f}(y)$ represents the pixel value at position y.

- **Patch Matching:**

  Given a reference patch $P(x)$ centered at pixel x, we search for similar patches in the image. Patch similarity can be computed using various metrics such as the Euclidean distance:

  $$d(P(x),P(y)) = \sum_{i=1}^{N} (f_i(x) - f_i(y))^2$$

  where $f_i(x)$ and $f_i(y)$ are the intensities of pixel i in patches $P(x)$ and $P(y)$ respectively.

- **Weighted Averaging:**

  The denoised estimate $\hat{f}(x)$ is computed by averaging pixel values from similar patches, weighted by their similarity:

  $$\hat{f}(x) = \frac{1}{Z(x)} \sum_{y \in \Omega} w(x,y) \cdot \hat{f}(x)$$

  The weights $w(x,y)$ are determined based on the similarity between patches. Typically, a Gaussian-like function is used:

  $$w(x,y) = \frac{1}{\sigma^2} \exp\left(-\frac{d(P(x),P(y))^2}{2\sigma^2}\right)$$

  where σ is a parameter controlling the level of smoothing.

- **Edge Preservation:**

  Since similar patches tend to belong to the same underlying structures, NLM effectively preserves edges by averaging pixel values from similar patches. Edges are characterized by sharp intensity transitions. Similar patches along edges will have similar intensity gradients, leading to higher weights and thus preserving edges in the denoised image.

- **Computational Complexity:**

  NLM involves comparing patches across the entire image, making it computationally expensive. Various optimizations, such as using integral images or hierarchical search strategies, are employed to speed up the process while maintaining effectiveness.

Unlike simpler techniques, NLM preserves fine details, patterns, and small-scale features that might have been obscured by noise in the original image. It excels at preserving edges, ensuring sharp intensity transitions are maintained, thereby enhancing visual quality and clarity. Overall, NLM yields an image that closely resembles the original scene but with reduced noise levels, making it visually more appealing and suitable for various image processing tasks.

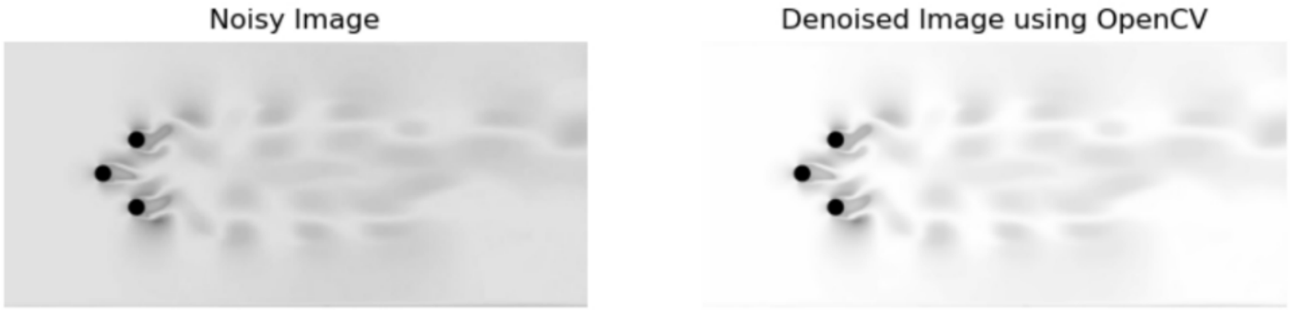


Figure 2. grayscale image of frame_100 is denoised

ANALYSIS

- Visual Quality - the denoised image has more clarity and sharpness due to the white background and looks more asthetic.

- Structural Preservation - the denoised image looks less blurry as it has a bit more well-defined boundaries and edges.

- Noise Reduction - noise is present in an image, it contributes to the variability of pixel values, leading to a higher standard deviation.The standard deviation of the pixel values has reduced therefore the noise has been reduced.

- Detail Enhancement - the fine details have been enhanced.

- Quantitative Metrics: An increase in PSNR indicates that the denoised/super-resolved images have a higher peak signal-to-noise ratio, suggesting reduced distortion and noise
  . An increase in SSIM indicates a higher structural similarity between the true and denoised/super-resolved images, reflecting better preservation of structural information and texture details.
  A decrease in MSE signifies lower mean squared error between the true and denoised/super-resolved images, indicating closer alignment of pixel values.

B] Scikit image (Non-local Means denoising)

While both scikit-image and OpenCV offer Non-Local Means (NLM) denoising algorithms, there are differences in their implementations and usage:

Implementation:

Scikit-image: The NLM denoising algorithm in scikit-image is implemented in Python and utilizes NumPy arrays for image processing operations. It is part of the skimage.restoration module.

OpenCV: OpenCV's NLM denoising algorithm is implemented in C++ with bindings for Python. It leverages OpenCV's optimized image processing routines and is part of the cv2 module.

Functionality:

Scikit-image: The NLM denoising function in scikit-image offers flexibility in parameter tuning, allowing users to adjust parameters such as patch size, patch distance, and filtering strength.

OpenCV: OpenCV's NLM denoising function provides similar functionality but may offer additional parameters or optimizations specific to the OpenCV implementation.

Performance:

Scikit-image: Scikit-image prioritizes simplicity and ease of use, making it suitable for smaller images or applications where performance is not a critical concern.

OpenCV: OpenCV is known for its performance optimization and efficient implementation of image processing algorithms. It may offer faster execution times, especially for larger images or real-time applications.

Firstly I have added random gaussian noise with variance 0.1 to the original image of frame_100.

Then I have used the scikit image library to denoise the image with parameters :

h (Smoothing Parameter):
The h parameter controls the strength of denoising in NLMeans. A higher h value results in stronger denoising. Here, h=0.15 indicates a moderate denoising strength. Increasing h would lead to more aggressive smoothing, while decreasing it would preserve more detail but may leave more noise.

sigma_est (estimation of Standard Deviation of Noise):
Accurate estimation of sigma is crucial for NLMeans to adapt its denoising strength effectively.

fast_mode (Fast Mode Flag):
The fast_mode parameter determines whether to use a faster version of the NLMeans algorithm. When set to True, speed is prioritized over accuracy. Here, fast_mode=True is chosen, indicating a preference for faster processing. This can be beneficial for real-time applications or when processing large images.

patch_size (Size of Patches):
The patch_size parameter specifies the size of patches used for similarity comparison during denoising. It defines the local neighborhood around each pixel considered for denoising. A patch_size of 5 indicates relatively small patches. Larger patch sizes capture more context but can increase computational complexity.

patch_distance (Patch Distance):
The patch_distance parameter controls the maximum distance between patches for similarity comparison. It defines the search radius within which NLMeans looks for similar patches. With patch_distance=3, NLMeans considers patches within a small distance from each pixel, limiting the search space and reducing computational cost.

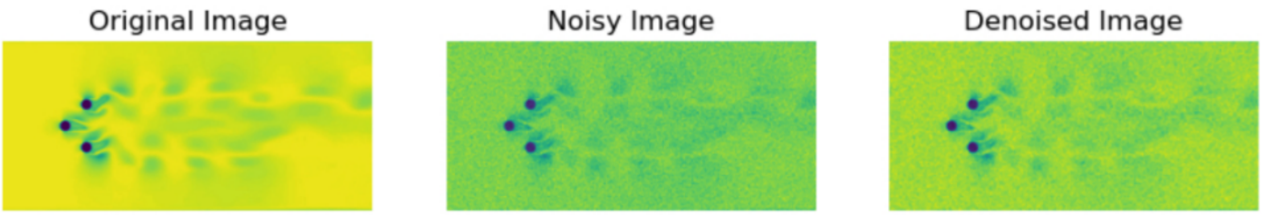
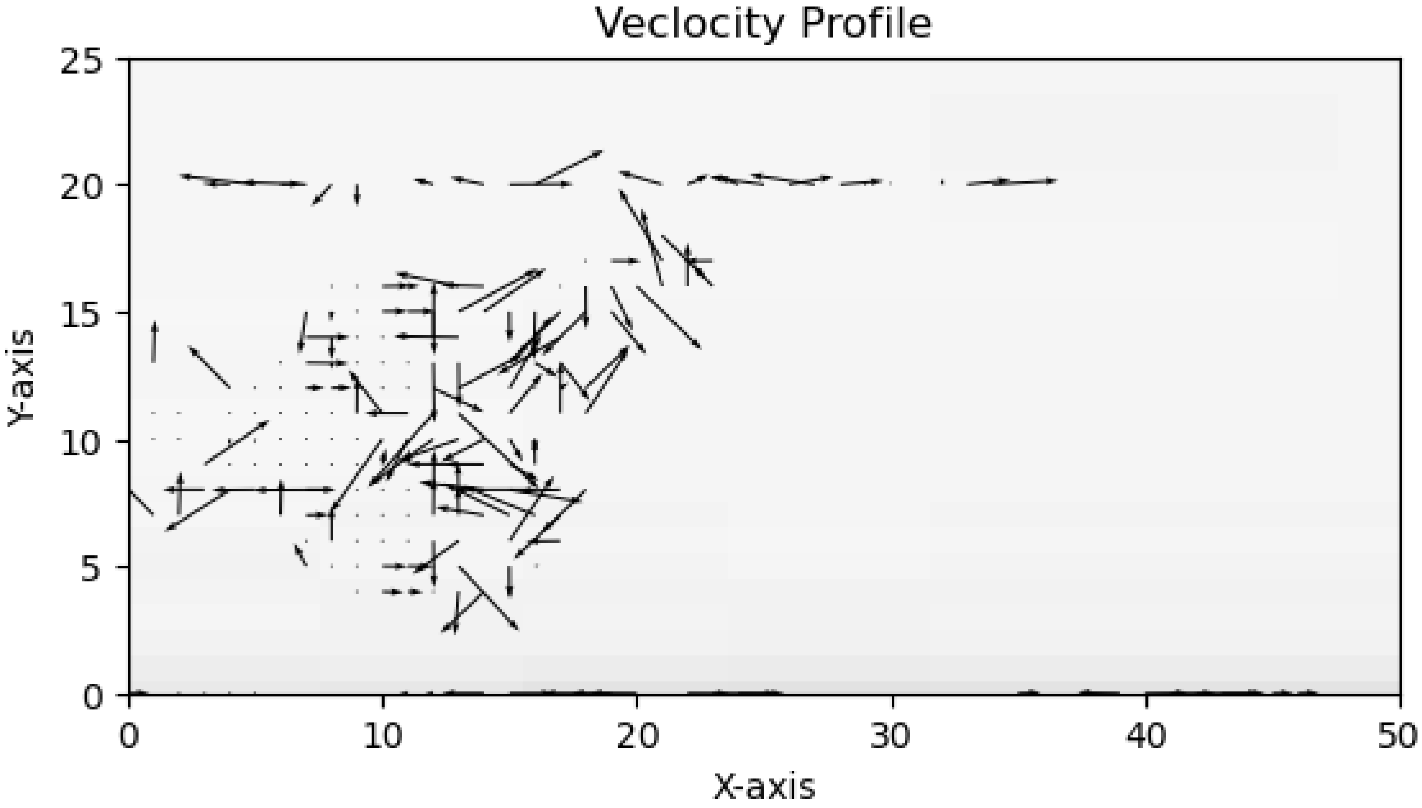


Figure 3. grayscale image of frame_100 is denoised

ANALYSIS

Though there may be differences in visual appearance and computational efficiency , the overall denoised image has very subtle variations and depending on the specific image characteristics and noise levels, one library's output may be preferred over the other

Veclocity Field Profile


Veclocity Profile
Y-axis
X-axis