Environment. Technology. Resources. Rezekne, Latvia
Proceedings of the 16th International Scientific and Practical Conference, 2025. Volume IV, 337-343

Proper Orthogonal Decomposition Technique for Dimensional Reduction for Numerical Snapshot Matrix

Rajeev Ralla
Institute of Mechanical and Biomedical Engineering
Riga Technical University
Riga, Latvia
rajeev.ralla@edu.rtu.lv

Karunamoorthy Rengasamy Kannathasan
Institute of Mechanical and Biomedical Engineering
Riga Technical University
Riga, Latvia
karunamoorthy.rengasamy-kannathasan@rtu.lv

Ilgar Jafarli
Institute of Mechanical and Biomedical Engineering
Riga Technical University
Riga, Latvia
ilgar.jafarli@rtu.lv

Abstract*— Computational fluid dynamics (CFD) simulations generate huge datasets with many variables such as Velocity and Pressure at nodes, elements, and time steps. It is challenging and expensive to analysis large datasets and storage, developing applications over the large datasets required additional computational resources and storage volume. To addresses this problem, we have employed data reduction technique for the reduction of the complexity and utilize its computational resources. Reduced Order Modelling emerged technique leading dimensional reduction yet preserving dominant features, patterns and discarding less significant features. Focusing on developing the Reduced Order Models (ROM) for data compression, we implemented Proper Orthogonal Decomposition (POD) technique for dimensionality reduction. Proper Orthogonal Decomposition (POD) is widely used for building efficient Reduced Order Model (ROM) in CFD simulations, enabling effective dimensionality reduction while preserving key flow characteristics. POD extracts the high-energy structures from the flow field, representing dominant patterns through POD modes that contributes to the overall dynamics. Dimensional reduction is achieved by truncating lower-energy modes based on rank, retaining only the most significant features. In this paper we conducted numerical simulation flow over the convex shape object cantered in the flow path, with parametric velocity ranging 25 -300 m/s and density 900 - 2000 kg/m2 of the water. The solution matrix is extracted and total volume was 8.83 MB. After computing POD modes and captured five most energetic modes (Rank = 5) and truncated other modes. The resulting ROM took 4.42 MB achieving a data volume saving of 4.41 MB (8.83 - 4.42 MB). when compared with volume consumptions over the single full numerical simulation (482 MB) to ROM (4.2 MB) is 477.8 MB and percentage occupation for the ROM is only 0.87% which is less than 1% of the original size of full simulation. This study also explores effective techniques for generating and truncating POD-based ROM. It also demonstrates the use of EZyRB-based ROM combined with Radial Basis Function (RBF) interpolation is applied to the POD coefficients using EZyRB library allows fast reconstruction of flow fields without running Full Order Model (FOM) to predict system responses for new parameter values, density 1232 kg/m2 and velocity 152 m/s we successfully reconstructed the flow field using ROM without conducting FOM. These new parameter values were projecting onto the original spatial domain for the visualization and comparing results with Ansys Fluent FOM. Highlighting the efficiency and effectiveness of ROM in parametric interpolation with lower dimensions and provides an intuitive approach for developing digital models that accurately represent physical systems with significantly reduced the computational volume and resources.

Keywords*— CFD, Data Reduction, EZyRB, POD, Radial Based Function, ROM.

I. Introduction

High fidelity CFD simulations are essential for analyzing fluid flow behavior over an object in applications like aerospace and biomedical field, often requires significant computational resources and results huge datasets [1], [2]. These datasets are not only difficult to store and manage, but also practically infeasible to work with the machine learning models since data sets contains millions of nodal points in spatial and field variables. Interpretation or manipulation of these datasets, sometimes required additional libraries, which can make real time analysis difficult. Reduction of this data set dimensionality will reduces the data store capacity, facilitate faster

Online ISSN 2256-070X
https://doi.org/10.17770/etr2025vol4.8447

computation. In many industrial and research applications, parametric simulations are repeatedly conducted to evaluate system performances. This highlights the demand for Model reduction techniques and interpolation methods to handle unseen variables, aiming to lower the computational cost and data volume [3]. This paper addresses the advantages of developing Reduced Order Model (ROM) and integrating with interpolation techniques to fasten the simulations without conducting Full Order Model (FOM) [4]. A data-driven technique is required to reduce complexity while preserving the high accuracy of the original data. Proper Orthogonal Decomposition (POD), forms the basis of Reduced Order Modeling [3]-[5]. When developing ROMs, generating data is the key step, and various methods can be employed. The snapshot method, or solution matrix of the simulated system is commonly used [4]-[6]. POD technique plays a major role, by decomposing the solution datasets and finding spatial structures [7] that captures the variance in a dataset typically identify the flow features during the transient simulation of the flow. EZyRB is a Python based open library for the developing ROM and RBF for the interpolation [8], [9]. EZyRB based ROM process typically in three steps 1. Data generation from CFD simulation high fidelity dataset. 2. POD extracting the dominant features and preserve. 3. RBF interpolation method can be used to predict the solution matrix for the new parameters. The purpose of following these steps is to develop the ROM and enhancing predictive models by reducing the complexity of dataset, while retaining key flow features. Developing a bridge by accessing the ROM models to other techniques like optimization and predicting modeling [10]-[11]. A comparative study between the FOM and ROM to demonstrate the reduction in storage capacity and runtime simulation making it valuable tool for reducing computational resources.

II. Materials and Methods

A. Method of Generating Datasets

The numerical snapshot [6] is a combination of spatial points, grid numbers and various parameters of interests such as velocity, pressure, and turbulence in matrix format. Typically, data generated from a series of snapshots, known as the snapshot method, is stored in Excel format. However, using Python, this data can be efficiently stored in arrays or matrices (e.g., NumPy arrays), enabling easy manipulation and processing without relying on Excel, thus improving computational efficiency. Where high resolution photos are taken, and each pixel are considered as grid points or elements. The core ideal behind developing the ROMs to preserve the physical system in a digital format. During ROM development, the data is typically organized as a series of rows and columns, representing different parameters or time steps. This approach provides a clear picture of the transformation of row and columns helping to understand the reduction of data points during implementation. Solution data is derived from the flow over a convex shaped solid body in middle of the rectangular fluid domain of length of 135 mm and height of 60 mm.

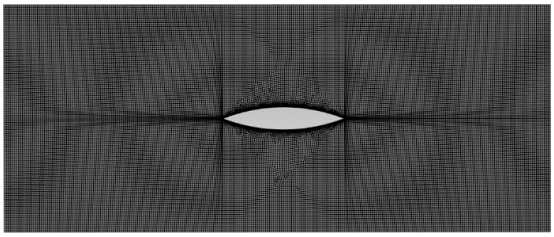

Fig. 1. Mesh Generation of Fluid Domain.

The mesh consists of a total of 38580 nodes, each node carrying the values of the velocities and pressure across the domain. A series of steady simulations are conducted using ANSYS Fluent, varying both density and velocity of fluid to analyze their effects on pressure and velocity distribution over a convex shape. The simulation data is carefully exported containing node numbers, spatial coordinates in the x and y directions, along with the relevant variables of interest.

TABLE 1 DATASET

Node number (Id)	x-coordinate (mm)	y-coordinate (mm)
10709	0.063636	0.019036
23525	0.0094683	0.033570
34025	0.0229060	0.052042

TABLE 2 DATASET TWO

x-velocity (m/s)	y-velocity (m/s)	total pressure (Pa)
55.85	-2.802	1605.152
50.00	-0.006	1604.587
50.26	-0.060	1610.362

The above table shows randomly selected data values from solution exported data points. Total data matrix consists of 38580 rows and 6 columns (38580,6), consuming 2.6MB storage. Similar, way we can extract 10 solution datasets with same number of nodes, each corresponding to different velocities and fluid densities.

TABLE 3 DENSITY AND VELOCITY

Density kg/m2	900	1000	1100	1200	1250
Velocity m/s	25	50	80	100	135

TABLE 4 DENSITY AND VELOCITY TWO

Density kg/m2	1300	1400	1500	1750	2000
Velocity m/s	150	200	225	250	300

The datasets comprise of 10 individual simulations solution data points, each with a shape of (38580, 6), resulting in a total storage of 26 MB. Each dataset is

flattened according to the parameters, the shape of flatten data (1 row x 115740 columns). The snapshot matrix or FOM is the collection and arrangement of these parameters, resulting in the shape of (10 row x 115740 columns), containing a total of 1157400 data points and occupying 8.83 MB of storage. This structure provides a detailed representation of the spatial data points.

B. Method of Dimensional Reduction

Dimensional reduction displays the approximation of high-fidelity data here snapshot matrix data points of shape (10, 1157400) points projecting into lower dimensional space which captures most of the essential features of snapshot matrix. Understanding the modes that captures most of the features and truncated the remaining modes by ranks which has lower performances easy to store data and increase in computational efficiency. The original snapshot matrix took 8.83MB storage by applying data reduction techniques, we can compare storage capacity. POD method is used to reduce the dimensionality of snapshot matrix, can ensure the accurately represents the system. Memory efficiency and high-fidelity data combination enables the handling of larger datasets still maintain precision in simulations and analyses. Single Value Decomposition (SVD) is a matrix decomposition tool to perform POD, POD method captures the important modes which has maximum variance, and the modes are orthogonal. Capturing the orthogonal matrix using the SVD decomposition of the snapshot matrix (m × n) where m is the number of snapshots and n is the number of spatial points in this case.

$$Snapshotmatrix\ \_X\ = U\Sigma V^T \qquad (1)$$

$U \in \mathbb{R}^{mxm}$ contain the left singular vector matrix U of shape m × m only contain the time depended on structures.

$\Sigma \in \mathbb{R}^{nxn}$ is a diagonal matrix n × n with singular values σ_i are the energy modes, POD utilizes this energy of the modes for the data compression or dimensionality reduction.

$V^T \in \mathbb{R}^{nxn}$ contains the right singular vectors matrix, V^T which consist of n × n spatial modes.

In a diagonal matrix, contains real and non-negative values in the diagonals, and zeros are present outside the diagonal. Singular values are arranged in descending order, from higher to lower. Truncated SVD based on ranks or modes, where the highest singular values are preserved, and the lower ones are discarded. If the singular values capture the higher singular values in first 3 or 4 singular values capturing the 99% of the physical system variance, truncating the matrix to rank 3 or mode 3 can effectively reduce dimensionality of the matrix, preserving the most significant modes and eliminating the less important ones.

$$Trancated_{snapshotmatrix_{X_r}} =$$

$$= U_r\Sigma_r{V^T}_r \qquad (2)$$

$U_r \in \mathbb{R}^{mxr}$ contains the left singular vectors matrix of shape (m, r) reduction based on rank columns.

$\Sigma_r \in \mathbb{R}^{rxr}$ is a diagonal matrix r × r with singular values σ_r becomes a square matrix size based on rank. Since most energy modes are captured in initial singular values and remaining off diagonal values with lower impact in collection can be neglected, Consequently, truncating the smaller singular values leads to a reduction in the size of the diagonal matrix, effectively lowering the rank and reducing the dimensionality of the system while retaining the most significant features.

${V^T}_r \in \mathbb{R}^{rxn}$ contains the right singular vectors matrix of ${V^T}_r$ of shape of n × n, it plays a major role to project the reduced truncated matrix back to the original space and is aligned with the columns of the original matrix.

Truncated_snapshotmatrix_ X_r , provides an approximation of the original matrix on the low-rank space. Each component in the matrix decomposition has significant dimensionality reduction based on the rank r. The truncated left singular vectors and diagonal matrix $U_r\Sigma_r$, represent the transformation that projects data onto the low-rank subspace, with dimensions corresponding to the row of the original matrix. Truncated right singular vector ${V^T}_r$ correspond to the dimension of the column of the original matrix are used to reproject the low-rank space back onto the original space, reconstructing an approximation of the original matrix.

The coefficient matrix plays a crucial role in interpolation, as it is associated with right singular vectors and diagonal matrix. Coefficient matrix $\mathsf{C} \in \mathbb{R}^{rxn}$ contains parametric coefficients, reflexes the contribution of modes to the original data. It establishes the mapping between the input parameters and reduction model, leading to effective interpolation to predict solution for the unknown parameter values.

$$Coefficients\_matrix = C = \Sigma_r{V^T}_r \qquad (3)$$

For developing ROM, EZyRB [12], Pandas [13] and NumPy [14] Python-based libraries are used by converting to flattened data matrix to reduced order modeling. EZyRB: Easy Reduced Basis Method package divides the workflow into two sections. Section 1. Data generation and construction of data base based on the parameters and develop dimensional reduction using POD.

Section 2. Surrogate model construction on the parametric data using RBF interpolation for mapping between parameters and coefficients matrix for the fast and smooth approximation for the new parameters [15].

The following algorithm is used POD values extraction from the snapshot matrix and modes for visualization and followed by the truncation.

```
# pod
Rank = 5 # Retain 5 dominant modes
pod = POD (method='svd', rank=Rank)
pod.fit(snapshots_matrix)
reduced_snapshots          =          pod.
transform(snapshots_matrix)
#----------------------------------------
----------------------------
print     ("Singular     values:",     pod.
singular_values)
# Calculate the total variance
total     variance     =     np.sum     (pod.
singular_values**2)
# Calculate the explained variance for each
mode
Explained_variance          =          pod.
singular_values**2 / total_variance
# Calculate and print cumulative explained
variance
cumulative_variance          =          np.
cumsum(explained_variance)
print  ("Cumulative  Explained  Variance:",
cumulative_variance)
```

Extracting Singular values, Cumulative explained variance and explained variance along with additional POD shapes [16] and modes are vital for making decisions for the additional truncation required for retaining or removing additional modes.

The original shape of the matrix (10, 115740), with 10 parametric snapshots. The POD modes shape (10, 5) retains 5 significant modes. These modes are retained and projected back onto the original spatial points, resulting in a reduced snapshot of 5 modes over 115740 spatial points, resulting in a reduced snapshot that reconstructs modes on the original space.

TABLE 5 SINGULAR VALUES

Mode =5	Singular values	Explained variables per mode	Cumulative Explained Variance
1	2.23e+10	9.99e-01	0.99996171
2	1.37e+08	3.80e-05	0.99999977
3	7.05e+06	9.97e-08	0.99999987
4	5.94e+06	7.07e-08	0.99999994
5	5.50e+06	6.06e-08	1

From the above table, Mode1 has the higher singular values if 2.23e10 and explained variance 9.99e-01 the first mode almost captures the 99.9% of variance in the data, which become significant in capturing. Followed by the mode 2 singular value 1.37e+8 and explained variance 3.80e-5 very small remains some variance captured still not significant as mode l but remains explaining small fraction of variance. Mode 3 and beyond singular value continuous to decreases and minimal variance changes. cumulative explained variance shows the total percentage variance explained by the modes by 5 cumulative variances reached 1 says 100% variance are explained within 5 modes. Considering the adding modes beyond mode 5, occupies extra space still retain all the variance captured within mode 5.

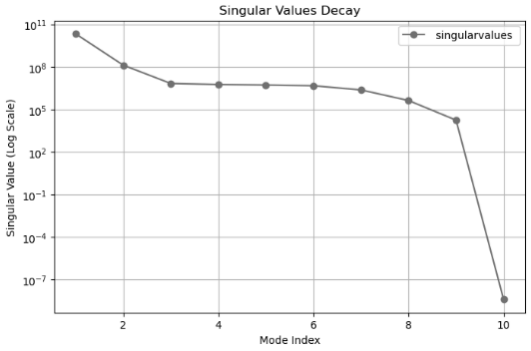

Fig. 2. Singular Values of the 10 snapshots.

From the above graph, we can see all 10 modes of the snapshot matrix dataset. After fifth mode, the singular values are decrease drastically, indicating their participation to capturing the variance is negligible.

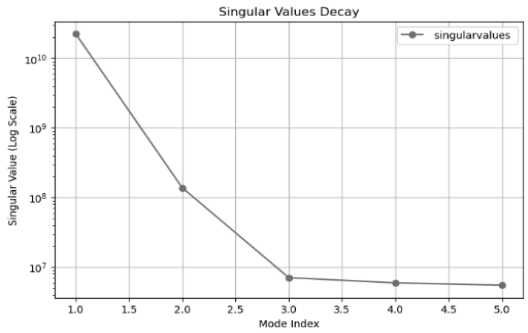

Fig. 3. Rank=5 Truncated Singular Values.

After mode 5 we can truncate for the data reduction this will justify truncation based on the singular values. Data storage capacity, the original data snapshot matrix required 8.83 MB storage capacity, ROM storage: 4.42 MB almost the half of the size and Storage Difference: 4.41 MB space is saved. This lightweight data set easy to handle and manipulation.

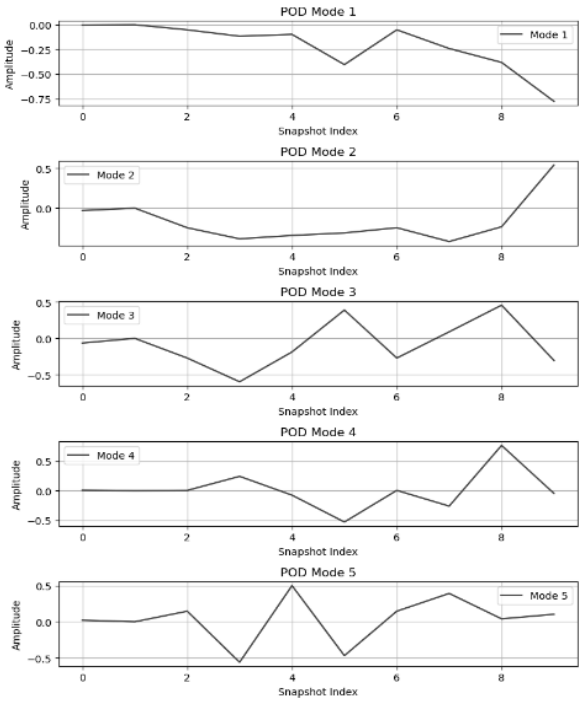

Fig. 4. POD Modes Rank = 5.

POD modes [17] represent the dominant features or spatial structure in the snapshot matrix, highlighting the most significant flow features. The x-axis represents the snapshot index, corresponding to 10 snapshots captured under various parameters. The y-axis value or amplitude, indicates the strong contribution of variance at different snapshots (spatial location or flow region) in the flow region. Amplitude of mode 1 exhibits smoothly and transitioning into negative value indicating an opposite influence this case high to low pressure in certain regions of the spatial domain.

Mode 2, amplitude start with 0 and moves towards positive value by the final snapshot index. This mode likely captures he spatial distribution of pressure or development of flow features over time. Modes 3 and 4 exhibit fluctuations, associated with transient effects or turbulent features which as smaller effects in the flow [18]. Higher-order modes primarily capture the small variation in the flow or secondary features. By truncation based on the amplitude avoiding the more oscillating amplitudes or focusing on dominant flow structure.

C. Interpolation

The EZyRB package enhances computational efficiency by incorporating the RBF [16] enabling rapid solution approximate for new parameters avoiding the need to compute full model simulation. This significantly reduces computational cost while maintaining accuracy. The following code demonstrates how to implement this approach:

```
from ezyrb import RBF
from ezyrb import Database
  rbf = RBF (kernel='gaussian', smooth=0,
                epsilon=1)
       rbf. Fit (parameters_matrix,
            reduced_snapshots)
         print (rbf. interpolator)
 new_parameters = np. array ([1232, 152]])
                # value
        newparameter_values = rbf.
          predict(new_parameters)
      print (predicted_values. shape)
    reshaped_values = predicted_values.
        flatten (). reshape (-1, 3)
```

RBF is commonly used for interpolation between known parameters using snapshot data points. The algorithm presented utilizes a Gaussian kernel [11], where the RBF compute's function values based on the distance from the center, ensuring smooth and accurate approximations.

$$\varphi(r) = e^{-\epsilon r^2} \quad (4)$$

where:
r = $\|x - c\|$ from the point x to center c
ϵ is a positive parameter that controls the width of the Gaussian function, ϵ =1 from the above code.

In the RBF framework, the center points represent the parametric locations where the solution matrix is constructed. POD is then applied to reduce data complexity. Each parameter corresponds to functional values, or snapshot data points, which depend on the parameter matrix, ensuring efficient representation of the system's behavior. RBF interpolation utilizes the snapshot data to construct a predictive model allowing for the estimation of functional values or generating new snapshots for unseen input parameters within the defined domain or parameter matrix range. The given new parameter values (1232, 152) for density and velocity of the water. After interpolation produces a new snapshot with a flattened shape of (1, 115740). This single row representation retains all the variables velocity in x, y direction and total pressure. The final step to recreate the spatial points for the visualization. Deflattening the data points, arrange the columns according to original data shape (38580, 6) and projecting onto the original nodes restore the visual representation of the ROM, closely approximates the original system.

These are comparative results between commercial software Ansys Fluent with the EZyRB package for the new parameters in terms of storage. The FOM from the Fluent took 482 MB of storage including mesh and solution data. The EZyRB based ROM model took 4.2MB of storage capacity.

1. Volume Occupation Difference (in MB)

$$\text{Difference} = \text{Full Model size} - \text{ROM Model size} = 482\ \text{MB} - 4.2\ \text{MB} = 477.8\ \text{MB}$$

2. Percentage occupation of ROM compared Full Model

$$\text{Percentage Occupation} = \left(\frac{ROM\ Model\ size}{Full\ Model\ size}\right) * 100 \quad (5)$$
$$= \left(\frac{4.2}{482}\right) * 100 \ = 0.87\%$$

ROM only occupies 0.87% of the FOM storage capacity. It reduced the storage capacity by approximately 99.13% to the FOM.

III. Results and Discussion

Reconstructed pressure for new parameters density 1232 kg/m^2 and velocity 152 m/s.

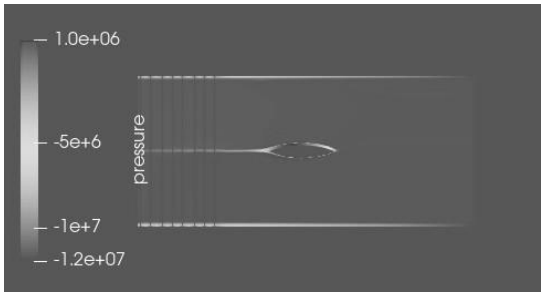

Fig. 5. Reconstructed Pressure Values in spatial points.

TABLE 6 PRESSURE VALUES

Max	Min
1.0e+06 Pa	-1.2e+7 Pa

Ansys Fluent FOM simulation for the new parameter's density 1232 kg/m^2 and velocity 152 m/s.

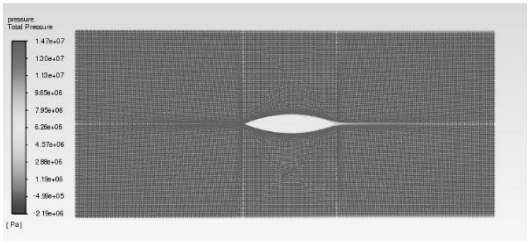

Fig. 6. FOM Simulation for the Pressure.

TABLE 7 PRESSURE VALUES

Max	Min
1.47e+07 Pa	-2.19e+6 Pa

Reconstructed model x-velocity direction

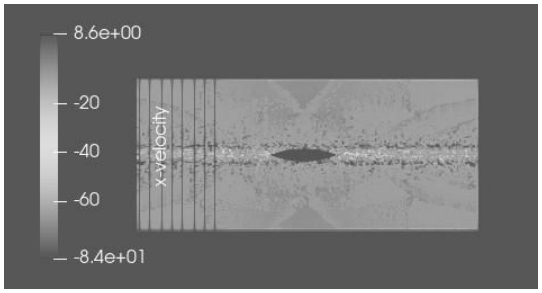

Fig. 7. Reconstructed x-velocity in spatial points.

TABLE 6 RECONSTRUCTED X-VELOCITY

Max	Min
1.47e+07 Pa	-2.19e+6 Pa

Ansys FOM simulation of the x-velocity direction

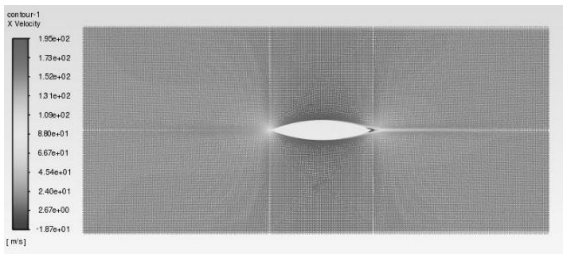

Fig. 8. FOM of x-velocity.

TABLE 7 FOM X-VELOCITY

Max	Min
1.47e+07 Pa	-2.19e+6 Pa

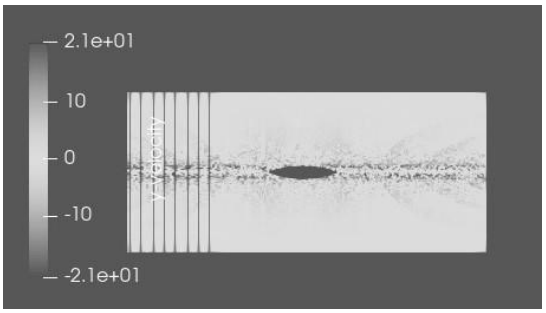

Fig. 9. Reconstructed Y- Velocity.

TABLE 8 RECONSTRUCTED Y-VELOCITY

Max	Min
1.47e+07 Pa	-2.19e+6 Pa

Ansys FOM simulation y velocity direction

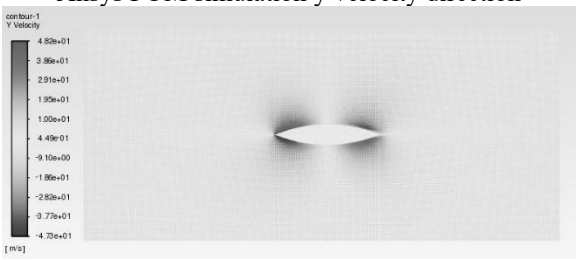

Fig. 10. FOM of Y- velocity.

TABLE 9 FOM Y-VELOCITY

Max	Min
1.47e+07 Pa	-2.19e+6 Pa

ROM captured the majority features and truncated small-scale fluctuations. In contrast, the FOM simulation

captures all the features, these changes significantly in the overall results. Reconstructed pressure has slightly deviated results compare to the FOM. Reconstructed pressure, x-velocity and y-velocity and compare with FOM, ROM, Pressure max value 1.0e+06 Pa and FOM 1.47e+07 Pa and the percentage difference is approximately 38.0%. ROM, velocities in x- direction has 8.6e+00 m/s and FOM 1.95e+02 m/s and the percentage difference is approximately 183.2%. y-direction 2.1e+01 m/s and FOM 4.82e+01 m/s and the percentage difference is approximately 78.6%, this effect demonstrates the truncation overall effective features are captured for preservation and perform multiple simulations with various parameters without going for FOM simulation other.

IV. COUNCLUSION

The Reduced Order Model (ROM) captures effectively the most dominant features while reducing the storage. Solution matrix occupied 8.83 MB storage capacity, but ROM took 4.42MB nearly half of its size. Percentage occupation of ROM compared Full Model is 0.87% of storage capacity. Reduction of the computational complexity with less storage capacity yet providing the approximate results without simulating FOM. ROM approach beneficial in applications required rapid evaluations, such as optimization and faster iterations. ROM can integrated with various machine learning techniques for predictive models making them a valuable tool for multidisciplinary optimization and digital twin applications. The presented approach enables the fastening the prediction of flow behavior using reduced models. We can extent this approach over the 3-dimensional data sets typically 3d simulations such as UAV design simulations, ventilation system and turbomachinery flow. By compressing high dimensional 3D spatial and flow field variables into lower dimensions space capturing the dominant features, enabling the interpolation can perform seamlessly and allow for reconstruct the parametric scenario on full scale and project original space. This reduces the time consumption unlike full order model and reduction in data volume. The reduced order model (ROM) enables faster iteration in optimization, system performance evaluation and decision making, become valuable tool for storage.

REFERENCES

[1] P. L. J. L. &. B. G. Holmes, Turbulence, Coherent Structures, Dynamical Systems and Symmetry., United Kingdom.: Cambridge University Press., 2012

[2] J. Reddy, An introduction to the finite element method, McGraw-Hill, 2006. T. Jordan and P. A. Taylor, Hacktivism and Cyberwars: Rebels with a cause? London: Routledge, 2004.

[3] S. L. &. K. J. N. Brunton, Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control., Cambridge, UK: Cambridge University Press., 2019.

[4] P. G. S. &. W. K. Benner, "A survey of projection-based model reduction methods for parametric dynamical systems," SIAM Review, vol. 57, no. 4, pp. 483-531, 2015.

[5] T. &. R. G. Taddei, "Model order reduction based on data: A survey of existing approaches and new. perspectives.," Advances in Computational Mathematics, vol. 3, no. 47, pp. 1-34, 2021.

[6] L. Sirovich, "Turbulence and the dynamics of coherent structures.," Quarterly of Applied Mathematics, vol. 3, no. 45, pp. 561-571, 1987.

[7] G. H. P. &. L. J. L. Berkooz, "The proper orthogonal decomposition in the analysis of turbulent flows.," Annual Review of Fluid Mechanics, vol. 25, pp. 539-575, 1993.

[8] J. &. B. P. Salmist, "EZyRB: A Python-based library for Reduced Order Modeling with Radial Basis Function Interpolation," Computational Mechanics, vol. 68, pp. 343-367., 2021.

[9] J. S. &. U. M. Hesthaven, Introduction to Reduced Order Modeling for Computational Fluid Dynamics., Society for Industrial and Applied Mathematics, 2016.

[10] K. &. D. S. Taira, "Data-driven methods for model reduction and flow analysis: Insights from EZyRB and other ROM techniques.," Journal of Fluid Mechanics, vol. 894, pp. 1-30., 2020.

[11] X. &. H. L. Ma, "Efficient interpolation methods for reduced order models using Radial Basis Functions," Computers & Fluids, vol. 186, pp. 123-132., 2019.

[12] J. N. K. Steven L. Brunton, Data Driven Science & Engineering., Washington : Brunton & Kutz, 2017.

[13] E. Contributors, "EZyRB Documentation," 1 april 2021. [Online]. Available: https://mathlab.github.io/EZyRB/. [Accessed 13 march 2025].

[14] T. P. D. Team, "Pandas Documentation," [Online]. Available: https://pandas.pydata.org/docs/user_guide/reshaping.html. [Accessed 13 March 2025].

[15] T. N. Developers, "NumPy Documentation," 14 october 2022. [Online]. Available: https://numpy.org/doc/. [Accessed 13 march 2025].

[16] M. T. a. G. R. Nicola Demo, "EZyRB: Easy Reduced Basis Method," The Journal of Open Source Software, vol. lll, no. 24, p. 662, 2018.

[17] S. Volkwein, Proper Orthogonal Decomposition: Theory and Reduced-Order Modelling, Konstanz: Lecture Note (August 27, 2013), 2013.

[18] P. M. L. R. T. Regert, "Investigation of the Link between Physics and POD Modes," RTO-MP-AVT-124, no. 4, pp. 11- 2, 2005.