Computational Mechanics by ROM

Giovanni Canali

August 4, 2025

1 Introduction

The primary objective of this project is to develop and validate a reduced-order modeling (ROM) framework capable of efficiently predicting flow fields in parametrically deformed geometries. Specifically, we focus on the backward-facing step problem, a classical benchmark in computational fluid dynamics known for exhibiting flow separation and recirculation phenomena sensitive to geometric changes.

By leveraging high-fidelity simulations generated via `OpenFOAM` version 2112 and geometric parameterization enabled through `PyGeM`[1], the aim is to create a compact, computationally inexpensive representation of the solution manifold across a range of geometric configurations. Such a model has significant practical relevance in scenarios where rapid evaluations of flow fields are required, including design optimization, control, and uncertainty quantification.

This work systematically explores the impact of smooth vertical deformations of the upper channel boundary, controlled by a scalar parameter μ, on the resulting flow patterns. The ROM is constructed through Proper Orthogonal Decomposition (POD) combined with Radial Basis Function (RBF) interpolation of modal coefficients, enabling accurate predictions for new, unseen geometric configurations without the need for repeated costly full-order simulations.

Overall, this study demonstrates the potential of projection-based ROM coupled with mesh morphing techniques to accelerate parametric CFD analysis while preserving essential flow physics.

2 Problem Description

We consider a two-dimensional, steady, incompressible, laminar flow of a Newtonian fluid through a geometrically parametrized domain featuring a backward-facing step. This classical configuration is widely used to investigate flow separation and recirculation phenomena, even in low-Reynolds-number regimes.

[1] `https://github.com/mathLab/PyGeM`

The computational domain consists of a channel with sudden expansion: the lower boundary includes a horizontal step near the inlet, while the upper boundary is vertically displaced by a geometric parameter $\mu \in [-1, 1]$. The parameter μ controls the vertical position of the upper wall in the inlet and outlet sections, introducing a smooth geometric deformation that influences the behavior of the flow. This parameterization captures essential variability in the domain's shape, making the problem suitable for parametric studies, as illustrated in Figure 1.

The fluid, assumed to be water with a constant kinematic viscosity of $\nu = 10^{-6}\,\mathrm{m}^2/\mathrm{s}$, enters the domain from the left boundary through a vertically oriented segment upstream of the step. A fully developed parabolic velocity profile is imposed at the inlet, with a maximum velocity of $U_{\max} = 10^{-5}\,\mathrm{m/s}$ at the centerline and zero velocity at the walls. The inlet velocity profile is given by

$$u_{\text{in}}(y) = 4U_{\max}\left(\frac{y - y_{\min}}{H}\right)\left(1 - \frac{y - y_{\min}}{H}\right), \tag{1}$$

where $H = 3 + \mu$ is the height of the inlet and $y_{\min} = 2$ is the lower extremum of the inlet segment. The vertical component of the velocity is zero at the inlet. A characteristic velocity $U = U_{\max}$ and characteristic length $L = 3$ (the mean inlet height) are chosen to define the Reynolds number:

$$\mathrm{Re} = \frac{UL}{\nu} = \frac{1 \times 10^{-5} \cdot 3}{1 \times 10^{-6}} = 30. \tag{2}$$

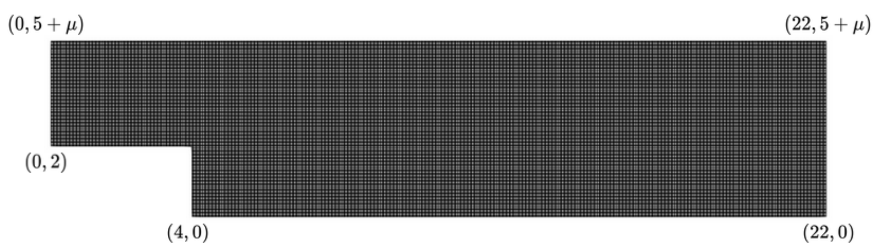

Figure 1: Schematic of the parametrized backward-facing step domain, highlighting the deformable upper boundary and the location of the step.

The flow is governed by the steady-state, incompressible Navier–Stokes equations in two spatial dimensions:

$$(\mathbf{u} \cdot \nabla)\mathbf{u} - \nu\Delta\mathbf{u} + \nabla p = \mathbf{0} \quad \text{in } \Omega_\mu, \tag{3}$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega_\mu, \tag{4}$$

where $\mathbf{u} = (u, v)$ is the velocity field, p is the pressure field, and Ω_μ is the parameter-dependent domain corresponding to a specific value of μ.

The boundary conditions are set as follows. A parabolic velocity profile is prescribed at the inlet, as defined in Equation (1), with no vertical velocity component. No-slip conditions are imposed along all solid walls of the domain, including the step surface. At the outlet, a zero normal gradient is imposed on the velocity field, and the pressure is fixed to zero to serve as a reference:

$$\frac{\partial \mathbf{u}}{\partial n} = 0, \quad p = 0 \quad \text{on the outlet.} \tag{5}$$

This configuration generates a primary recirculation zone just downstream of the step. The sensitivity of separation and reattachment to variations in μ makes this a valuable test case for studying the interaction between geometry and flow structure.

3 High-Fidelity Data Generation

To generate the high-fidelity snapshots required for the ROM, we begin with a reference simulation in `OpenFOAM`, corresponding to the undeformed geometry, i.e., $\mu = 0$. The mesh is generated using `blockMesh`, which creates the necessary files including the `points` file in the `constant/polyMesh` directory.

We then sample N values of the geometric deformation parameter $\mu \in [-1, 1]$, ensuring that both endpoints $\mu = -1$ and $\mu = 1$ are always included. For each sampled value, we apply a smooth geometric deformation to the mesh using the `PyGeM` library. This is done by modifying the mesh points stored in the `points` file, based on RBF interpolation. The deformation is applied only to selected regions of the mesh - specifically, the upper part of the domain - while the rest of the domain remains fixed. The deformation procedure ensures geometric consistency and smoothness.

Each deformed mesh is then used to initialize a new `OpenFOAM` case, created by copying the reference simulation directory. The `simpleFoam` solver is run for each deformed case to compute the steady-state velocity and pressure fields. The resulting high-fidelity data is converted to VTK format using `foamToVTK`, enabling post-processing and snapshot collection.

A summary of the pipeline is provided below.

1. Run `blockMesh` on the reference case with $\mu = 0$.
2. Sample N deformation parameters $\mu_i \in [-1, 1]$, including $\mu = \pm 1$.
3. For each μ_i, use `PyGeM` with RBF interpolation to deform the mesh.
4. Save the deformed mesh in the `points` file of a copied simulation folder.
5. Run `simpleFoam` and convert the results to VTK format.

Figure 2 shows an example of a deformed mesh corresponding to a non-zero value of μ.

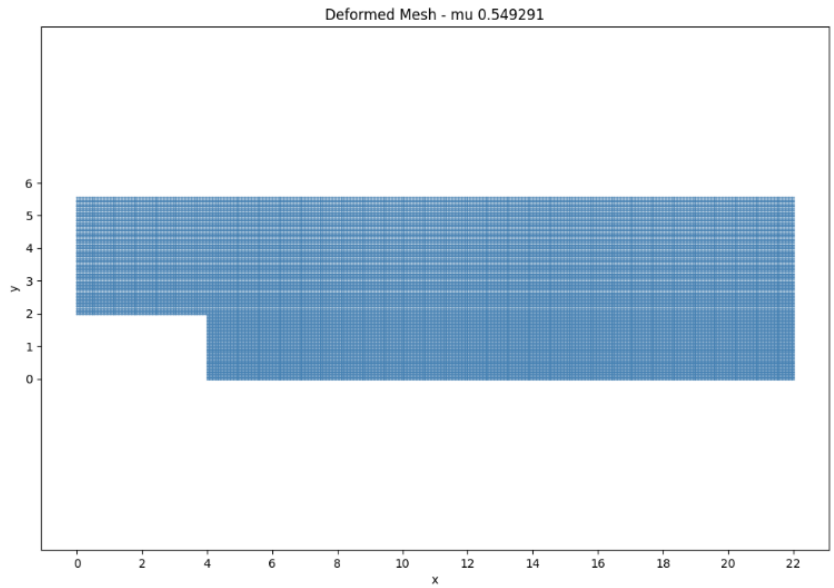

Figure 2: Example of a deformed mesh generated using RBF interpolation with $\mu = 0.549291$. The top boundary has been smoothly displaced upward.

4 Testing Phase and ROM Construction

To evaluate the predictive capabilities of the ROM, a comprehensive testing pipeline is executed, encompassing the construction of the reduced basis via POD, the application of RBF interpolation in the latent space, and the simulation of new test cases for rigorous comparison.

POD Basis Construction. The process begins by assembling a high-fidelity training dataset of velocity magnitude field snapshots, obtained from steady-state `OpenFOAM` simulations over a range of deformation parameters $\mu \in [-1, 1]$. To ensure consistency across samples, each velocity magnitude field is interpolated onto a common reference mesh. This produces a snapshot matrix $X \in \mathbb{R}^{N \times M}$, where N is the number of spatial degrees of freedom and M is the number of parameter samples. In this case, M is set to 15. A Singular Value Decomposition (SVD) is then performed:

$$X = U \Sigma V^T, \tag{6}$$

yielding the orthonormal POD modes in U, the singular values in Σ, and the parametric coefficients in V.

The singular values indicate the energy content of each mode and are used to determine an appropriate truncation rank r. This retains only the most energetic modes, significantly reducing the problem's dimensionality while preserving the dominant dynamics. The decay of the singular values, shown in the scree plot of Figure 3, demonstrates that a low-dimensional POD basis efficiently captures the dominant dynamics. Based on this decay, a POD rank of $r = 4$ is deemed sufficient to achieve an accurate approximation, balancing model complexity and predictive fidelity.

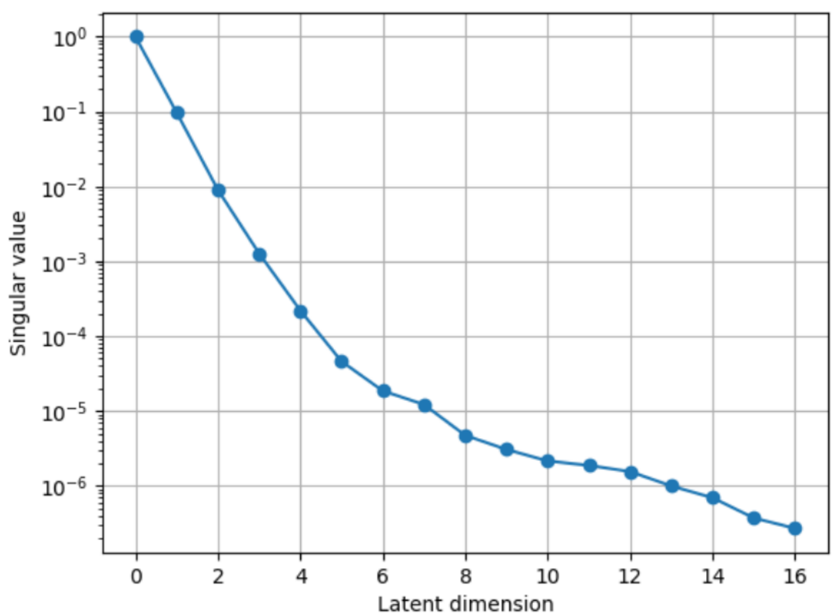

Figure 3: Decay of singular values obtained from the snapshot matrix of velocity magnitude fields. The rapid decay indicates that a low-dimensional POD basis captures most of the variance in the training data.

Latent-Space Interpolation with RBF. Once the POD basis is computed, each velocity magnitude field is projected onto the low-dimensional space spanned by the first r POD modes. This yields a set of latent-space coefficients $a_i \in \mathbb{R}^r$, associated with corresponding deformation parameters μ_i. A supervised learning problem is then defined to approximate the mapping:

$$\mu \mapsto a(\mu) \tag{7}$$

using RBF interpolation (specifically, a thin-plate spline kernel). The interpolator is trained on the parametric pairs $\{(\mu_i, a_i)\}$.

Prediction at Test Parameter. A new test parameter $\mu^{\text{test}} = 0.202369$ - not included in the training set - is selected. The trained RBF model predicts the corresponding latent-space coefficients a^{test}, which are then used to reconstruct the full velocity magnitude field via:

$$\tilde{u}(\mu^{\text{test}}) = \sum_{j=1}^{r} a_j^{\text{test}}\, u_j, \tag{8}$$

where u_j are the POD modes. The predicted solution is visualized and saved for subsequent comparison.

Simulation Setup for Validation. To validate the POD-RBF prediction, two additional `OpenFOAM` simulations are performed using the test parameter μ^{test}:

- **`foam_grid`:** A mesh is regenerated using `blockMesh`, with vertices directly modified to reflect the test deformation.
- **`pygem_grid`:** The original mesh is deformed using the same `PyGeM`-based RBF method used during training.

Both cases are solved with `simpleFoam`, and the velocity magnitude fields are exported to VTK format for detailed analysis.

Quantitative Evaluation. For each POD rank $r \in \{1, \ldots, r_{\max}\}$, the relative error and mean squared error of the POD-RBF predictions and the `PyGeM`-deformed simulations are computed against the `foam_grid` reference solution:

- **Relative error:**
$$\mathrm{RE} = \frac{\|u^{\text{ref}} - \tilde{u}\|_2}{\|u^{\text{ref}}\|_2}, \tag{9}$$
  where u^{ref} is the velocity magnitude field from `foam_grid`, and $\tilde{u}$ is either the POD-RBF prediction or the `pygem_grid` result.
- **Mean squared error:**
$$\mathrm{MSE} = \frac{1}{N}\|u^{\text{ref}} - \tilde{u}\|_2^2. \tag{10}$$

The results are shown in Figure 4 and summarized numerically in Table 1.

Rank	MSE	Rel. Err.	Rank	MSE	Rel. Err.
1	6.95×10^{-4}	7.30×10^{-2}	11	3.20×10^{-5}	1.39×10^{-2}
2	4.55×10^{-5}	1.86×10^{-2}	12	3.20×10^{-5}	1.39×10^{-2}
3	3.52×10^{-5}	1.47×10^{-2}	13	3.20×10^{-5}	1.39×10^{-2}
4	3.19×10^{-5}	1.39×10^{-2}	14	3.20×10^{-5}	1.39×10^{-2}
5	3.19×10^{-5}	1.39×10^{-2}	15	3.20×10^{-5}	1.39×10^{-2}
6	3.20×10^{-5}	1.39×10^{-2}	16	3.20×10^{-5}	1.39×10^{-2}
7	3.20×10^{-5}	1.39×10^{-2}	17	3.20×10^{-5}	1.39×10^{-2}
8	3.20×10^{-5}	1.39×10^{-2}	18	3.20×10^{-5}	1.39×10^{-2}
9	3.20×10^{-5}	1.39×10^{-2}	19	3.20×10^{-5}	1.39×10^{-2}
10	3.20×10^{-5}	1.39×10^{-2}	20	3.20×10^{-5}	1.39×10^{-2}

Table 1: POD model errors for different ranks. MSE = Mean Squared Error, Rel. Err. = Relative Error.

Qualitative Comparison. A qualitative comparison of the reconstructed fields is shown in Figure 5. Visual inspection confirms that increasing the POD rank reduces discrepancies. Notably, even at $r = 4$, the ROM captures essential flow features well, while $r = 20$ yields a near-identical match to the reference.

5 Conclusion

This project successfully developed and validated a ROM framework for the efficient prediction of flow fields within parametrically deformed geometries. The study focused on the classic backward-facing step problem, where geometric variations were introduced by a smooth vertical deformation of the upper channel wall, controlled by a scalar parameter μ. By integrating high-fidelity computational fluid dynamics simulations from `OpenFOAM` with mesh morphing via the `PyGeM` library, a robust dataset was generated to train the ROM.

The core of the methodology involved POD to extract a low-dimensional basis from the high-fidelity data, followed by RBF interpolation to map the geometric parameter to the modal coefficients in the reduced space. The results demonstrate the framework's effectiveness, showing a rapid decay in the singular values of the snapshot matrix, which confirms that a small number of POD modes can capture the system's dominant dynamics.

Quantitative validation performed on an unseen test parameter $\mu^{\text{test}} = 0.202369$, revealed that a POD rank of just 4 was sufficient to achieve a relative error of 1.39×10^{-2}. This level of accuracy is comparable to the intrinsic error introduced by the mesh deformation tool itself, indicating the high fidelity of the low-rank prediction. Qualitative comparisons of the reconstructed velocity fields further confirmed that a rank 4 model accurately captures the primary flow features, including the critical recirculation zone, while a rank 20 model produces a result that is nearly indistinguishable from the ground-truth simulation.

In conclusion, this work effectively illustrates the power of combining projection-based model reduction with advanced mesh parameterization techniques. The resulting computational model offers a fast and accurate tool for parametric analysis, presenting significant potential for applications in design optimization, control, and uncertainty quantification where rapid flow field evaluations are essential.

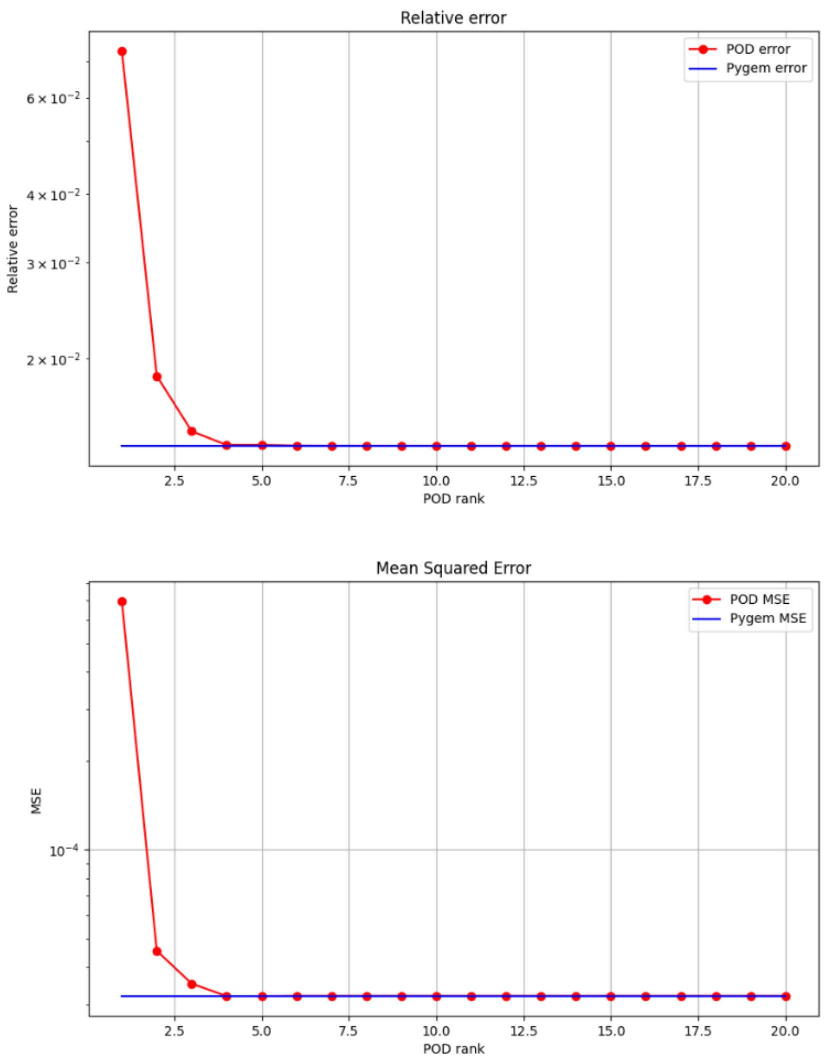

Figure 4: **Top**: Relative error of the POD prediction with respect to the high-resolution ground truth obtained with `OpenFOAM`. In blue: the relative error of the `OpenFOAM` simulation on the mesh deformed with `PyGeM`. **Bottom**: Mean squared error of the POD prediction.

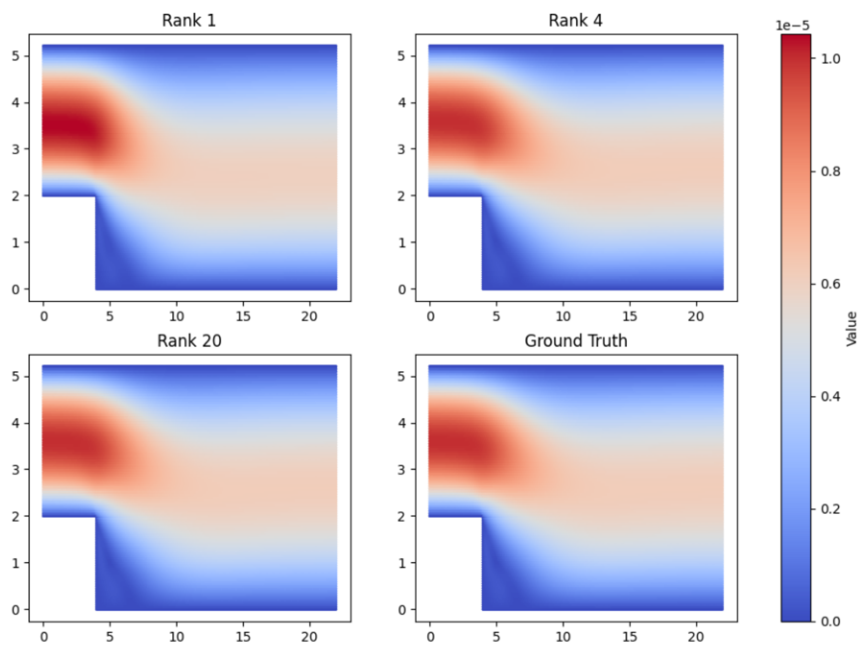

Figure 5: Comparison of predicted fields using POD-RBF reconstruction with increasing rank: Rank 1 (top left), Rank 4 (top right), and Rank 20 (bottom left). The bottom right shows the corresponding ground truth simulation. As the rank increases, the approximation quality improves, with the Rank 20 prediction closely resembling the ground truth. All plots share the same color scale and spatial domain.