Local Non-Intrusive Reduced-Order Modeling Using Isomap

Rakesh Halder[*], Krzysztof J. Fidkowski[†], and Kevin J. Maki[‡]

University of Michigan, Ann Arbor, MI, USA 48109

Physics-based modeling and simulation has become an essential tool in science and engineering for analysis and design of complex physical systems. Computational fluid dynamics (CFD) is highly prevalent in aerospace applications, allowing for the simulation of highly accurate flow fields. Industrial processes such as design optimization often require simulations at many different sets of design parameters, leading to a very high computational cost for large problems. Reduced-Order Models (ROMs) can mitigate this issue, allowing for rapid, real-time surrogate modeling of flow fields using some high-fidelity simulations as training data. Non-intrusive ROMs, which use a purely data-driven approach, often use a proper orthogonal decomposition (POD) of the training data along with a regression model such as Gaussian process regression (GPR) to make predictions at unseen design parameters. In this work, we introduce a non-intrusive ROM using locally constructed POD bases found using Isomap, a versatile technique for nonlinear dimensionality reduction. When tested on a lid-driven cavity problem, results show that the ROM increases performance over a number of local ROM dimensions.

I. Introduction

Physics-based modeling and simulation is an essential tool across many applications in engineering and science. By solving a set of governing equations, often found in the form of parameterized partial differential equations (PDEs), highly accurate representations of physical systems can be obtained that may otherwise be difficult to evaluate. In aerospace applications, computational fluid dynamics (CFD) is widely used to obtain flow-field information. Parameterized PDEs discretized over a computational domain Ω can be associated with a set of design parameters $\boldsymbol{\mu}$ that control properties such as the boundary conditions, geometry of the domain, and physical properties. Design optimization processes often require high-fidelity simulation at many sets of design parameters, which leads to a large computational cost that may render the process infeasible. Reduced-Order Models (ROMs) are commonly used to drastically lower this computational cost. ROMs utilize training data from a small number of chosen high-fidelity simulations to create a surrogate model that can be rapidly evaluated at unseen design parameters to obtain accurate solutions. ROMs reduce the dimensionality of full-order models, which contain a large number of degrees of freedom. In one approach to generating ROMs, a *reduced basis* of the solution space is computed in a compression phase, for which a set of expansion coefficients can be used to obtain accurate approximations of full-order solutions. There exist many methods for computing the reduced basis, with a very popular one being the proper orthogonal decomposition (POD) [1, 2]. A singular value decomposition (SVD) of a matrix containing training data is used to obtain a *POD basis* which consists of a low-rank set of linearly independent basis vectors. A linear combination of these basis vectors can be used to approximate states lying within the solution space.

ROMs utilizing the POD involve both projection-based approaches [3, 4] and non-intrusive approaches [5, 6]. Projection-based ROMs project the physics of the governing equations onto the POD basis to solve a low-dimensional model. Although they have been shown to offer robust performance, they can be computationally expensive for certain nonlinear problems. In the case of non-intrusive ROMs, the physics of the model are only used to generate training solutions. A regression model is used to predict expansion coefficients at unseen design parameters. Gaussian process regression (GPR) is widely used in non-intrusive ROMs for this purpose [7, 8], and the combined method is referred to as POD-GPR. In problems associated with different physical regimes in the solution space, building ROMs with local POD bases has been shown to alleviate performance issues in both intrusive and non-intrusive approaches [9, 10]. In this work, we introduce a method for constructing local POD bases based on Isomap [11], a versatile technique for nonlinear dimensionality reduction. By using only training samples that are physically similar to the given prediction point, the regression of the expansion coefficients in POD-GPR can be more robust leading to

[*] Graduate Student, Department of Aerospace Engineering, AIAA Student Member

[†] Associate Professor, Department of Aerospace Engineering, AIAA Associate Fellow

[‡] Associate Professor, Department of Naval Architecture and Marine Engineering

higher predictive performance. Isomap provides low-dimensional representations of training data along a manifold, which can be used as a measure of similarity. The performance of this method will be tested on a lid-driven cavity problem solved using the steady incompressible Navier-Stokes equations.

II. Methods

1. Full-Order Model

In this work, the full-order model (FOM) is considered to be the solution of a state variable $\boldsymbol{x}(\boldsymbol{\mu})$ to a set of governing parameterized partial differential equations (PDEs) that are discretized over a computational domain $\Omega \in \mathbb{R}^d$. The design parameters $\boldsymbol{\mu} \in \boldsymbol{\mathcal{D}}$ are used to define the parameters of the PDEs as well as the computational domain. $\boldsymbol{\mathcal{D}} \subseteq \mathbb{R}^p$ denotes the design parameter space such that $\boldsymbol{x} : \boldsymbol{\mathcal{D}} \to \mathbb{R}^N$. The computational cost of obtaining a solution $\boldsymbol{x}$ increases with the size of the computational domain, which is in proportion to the number of degrees of freedom N. In engineering design, it is often the case that sufficiently accurate solutions are only obtained when N is very large, leading to a very large computational cost for a single solution. This computational cost can render industrial processes such as design optimization infeasible, as the need to evaluate many different designs in real-time becomes computationally intractable. The use of reduced-order models is motivated by this large computational cost, where training data from the solutions to a small number of FOMs are used to construct a cheap and accurate surrogate model. In this work, solutions to the steady incompressible Navier-Stokes equations, particularly the velocity, are considered.

1.1. Flow Solution

OpenFOAM [12], an open-source toolbox for multiphysics simulation, is used for CFD simulation. Steady incompressible laminar flow is simulated using simpleFoam, a standard OpenFOAM solver, by solving the Navier-Stokes equations,

$$\int_S \vec{U} \cdot \mathrm{d}\vec{S} = 0, \tag{1}$$

$$\int_S \vec{U}\vec{U} \cdot \mathrm{d}\vec{S} + \int_V \nabla p \, \mathrm{d}V - \nu \int_S (\nabla \vec{U} + \nabla \vec{U}^T) \cdot \mathrm{d}\vec{S} = 0, \tag{2}$$

where $\vec{U}$ = $[u, v, w]$ is the velocity vector and u, v, and w are the velocity components in the x, y, and z directions respectively; $\vec{S}$ is the face-area vector; V is the volume; ν is the kinematic viscosity, and p is the pressure. The finite volume method (FVM) is used to discretize the continuity and momentum equations over the computational domain. The semi-implicit method for pressure-linked equations (SIMPLE) algorithm [13] is used to couple the equations along with Rhie–Chow interpolation [14].

The SIMPLE algorithm solves the discretized momentum equation for an intermediate velocity field by using the pressure field obtained from the previous iteration or an initial guess (p^0) in the first iteration. The momentum equation is then semi-discretized as

$$a_P \vec{U}_P = -\sum_N a_N \vec{U}_N - \nabla p^0 = \vec{H}(\vec{U}) - \nabla p^0, \tag{3}$$

where a is the coefficient resulting from the FVM discretization, P and N denote a control volume cell and its neighboring cells respectively, and $\vec{H}(\vec{U}) = -\sum_N a_N \vec{U}_N$ measures the influence of the velocity from cell neighbors. A new variable ϕ^0 (the cell-face flux) is introduced to linearize the convective term,

$$\int_S \vec{U}\vec{U} \cdot \mathrm{d}\vec{S} = \sum_f \vec{U}_f \vec{U}_f \cdot \vec{S}_f = \sum_f \phi^0 \vec{U}_f, \tag{4}$$

where f denotes the cell face. ϕ^0 is obtained from a previous iteration or from an initial guess. After solving Eq. (3), the intermediate velocity field is obtained which does not satisfy the continuity equation.

Next, the continuity and momentum equations are coupled to form a pressure Poisson equation, from which a new pressure field can be computed. The discretized form of the continuity equation is

$$\int_S \vec{U} \cdot \mathrm{d}\vec{S} = \sum_f \vec{U}_f \cdot \vec{S}_f = 0. \tag{5}$$

Rather than linearly interpolating, $\vec{U}_f$ is computed by interpolating the cell-centered velocity $\vec{U}_P$ obtained from equation (3) onto the cell face,

$$\vec{U}_f = \left(\frac{\vec{H}(\vec{U})}{a_P} \right)_f - \left(\frac{1}{a_P} \right)_f (\nabla p)_f. \tag{6}$$

Rhie and Chow [14] proposed this method of momentum interpolation, which is effective in mitigating the *checkerboard effect* found in collocated meshes, in which pressure oscillations arise. Substituting Eq. (6) into Eq. (5) gives us the pressure Poisson equation,

$$\nabla \cdot \left(\frac{1}{a_P} \nabla p \right) = \nabla \cdot \left(\frac{\vec{H}(\vec{U})}{a_P} \right). \tag{7}$$

Solving Eq. (7), an updated pressure field p^1 is computed. Finally, the cell-face flux is updated using the updated pressure field p^1,

$$\phi^1 = \vec{U}_f \cdot \vec{S}_f = \left[\left(\frac{\vec{H}(\vec{U})}{a_P} \right)_f - \left(\frac{1}{a_P} \right)_f (\nabla p^1)_f \right] \cdot \vec{S}_f. \tag{8}$$

A new velocity field is computed by solving Eq. (3) with the updated pressure and cell-face flux. This process is iteratively repeated until either a residual tolerance or maximum number of iterations is reached.

2. Reduced-Order Modeling

This section gives a brief overview of the POD-GPR method for constructing ROMs, also commonly referred to as POD-Kriging, which assumes that there exists a functional dependence between the design parameters and full-order solutions. This method involves calculating a proper orthogonal decomposition (POD) of a *snapshot matrix* containing solutions to the FOMs that comprise the training data using the singular value decomposition (SVD). A linear combination of a number of independent basis vectors found from the POD can be used to extract the underlying structure of the solution space [1]. Gaussian process regression is then used to predict the expansion coefficients associated with the basis vectors at a set of unrealized design parameters, allowing for accurate predictions of the full-order solution.

2.1. *Proper Orthogonal Decomposition*

Given a set of n solution snapshots calculated at chosen design points in the parameter space, a snapshot matrix $\boldsymbol{S} \in \mathbb{R}^{N \times n}$, is assembled,

$$\boldsymbol{S} \in \mathbb{R}^{N \times n} = [\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^n] = [\boldsymbol{x}(\boldsymbol{\mu}^1), \boldsymbol{x}(\boldsymbol{\mu}^2), \cdots, \boldsymbol{x}(\boldsymbol{\mu}^n)]. \tag{9}$$

There exists a subspace $\boldsymbol{\mathcal{V}}$ associated with $\boldsymbol{S}$, such that $\boldsymbol{\mathcal{V}} = \text{span}(\boldsymbol{S})$. It is assumed that $\boldsymbol{\mathcal{V}}$ provides a good approximation of the full-order solution manifold for $\boldsymbol{\mu} \in \boldsymbol{\mathcal{D}}$ if $\boldsymbol{S}$ contains a sufficient number of solution snapshots. A rank k set of orthonormal basis vectors $[\boldsymbol{\psi}^1, \boldsymbol{\psi}^2, \cdots, \boldsymbol{\psi}^k] \in \mathbb{R}^N$, where $N \gg k$, is associated with $\boldsymbol{\mathcal{V}}$ such that a solution snapshot $\boldsymbol{x}^i, i \in [1, 2, \cdots n]$ in $\boldsymbol{S}$ can be accurately represented as a linear combination of them

$$\boldsymbol{x}^i = a_1^i \boldsymbol{\psi}^1 + a_2^i \boldsymbol{\psi}^2 \cdots + a_k^i \boldsymbol{\psi}^k. \tag{10}$$

Where $\boldsymbol{a}^i$ consists of the *basis coefficients*, also referred to as expansion coefficients, for a given solution snapshot. The truncated singular value decomposition of $\boldsymbol{S}$ decomposes it into two orthonormal matrices $\boldsymbol{U} \in \mathbb{R}^{N \times n}$ and $\boldsymbol{V} \in \mathbb{R}^{n \times n}$ and a diagonal matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times n}$

$$\boldsymbol{S} = \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{V}^T. \tag{11}$$

We are interested in $\boldsymbol{U}$, which contains a set of n left singular vectors that form an orthonormal basis for the column space of $\boldsymbol{S}$ as well as diag($\boldsymbol{\Sigma}$) $\in \mathbb{R}^n = [\sigma_1, \sigma_2, \cdots, \sigma_n]$ which consists of the singular values in descending order, $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$. The POD basis $\boldsymbol{\Psi} \in \mathbb{R}^{N \times k} = [\boldsymbol{\psi}^1, \boldsymbol{\psi}^2, \cdots, \boldsymbol{\psi}^k]$ is formed by the first k left singular vectors of $\boldsymbol{U}$. The singular values are associated with the left singular vectors and often decay very rapidly. In many ROMs, it is common practice to use the first k singular vectors such that the POD basis preserves only dominant basis vectors. For any solution $\boldsymbol{x}$ the basis coefficients $\boldsymbol{a}$ that give the *projection* of $\boldsymbol{x}$ onto $\boldsymbol{\Psi}$ are computed as

$$\boldsymbol{a} = \boldsymbol{\Psi}^T \boldsymbol{x}. \tag{12}$$

Full-order solutions at unseen design parameters can be approximated using the POD basis

$$\boldsymbol{x}(\boldsymbol{\mu}^*) \approx \boldsymbol{\Psi} \boldsymbol{a}^* = a_1^* \boldsymbol{\psi}^1 + a_2^* \boldsymbol{\psi}^2 \cdots + a_k^* \boldsymbol{\psi}^k, \tag{13}$$

where $\boldsymbol{a}^*$ is estimated using a regression model in non-intrusive ROMs.

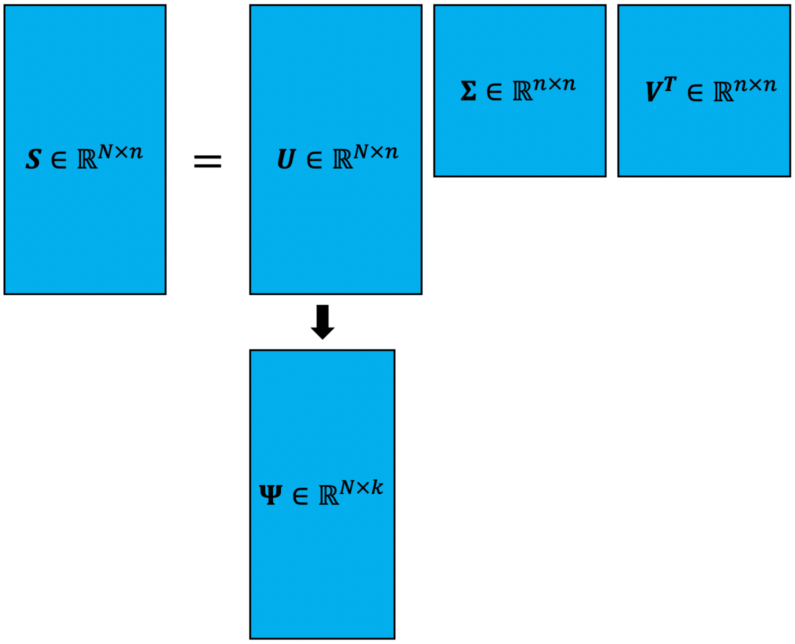

$S \in \mathbb{R}^{N\times n}$
=
$U \in \mathbb{R}^{N\times n}$
$\Sigma \in \mathbb{R}^{n\times n}$
$V^T \in \mathbb{R}^{n\times n}$
$\Psi \in \mathbb{R}^{N\times k}$

Figure 1: Schematic of the proper orthogonal decomposition (POD), where the snapshot matrix $\boldsymbol{S}$ is decomposed using the singular value decomposition (SVD) and the POD basis $\boldsymbol{\Psi}$ is obtained from $\boldsymbol{U}$.

2.2. *Gaussian Process Regression*

Gaussian process regression (GPR) is commonly used in non-intrusive ROMs for expansion coefficient prediction. In addition to being computationally inexpensive and easy to implement, GPR provides reasonable accuracy with limited training data, making it a very flexible regression model. GPR is a supervised learning method and Bayesian approach that is used for predictions of continuous outputs and had one of its first uses in the field of geostatistics [15]. In ROMs using GPR, each expansion coefficient in $\boldsymbol{a}$ is predicted with an individual regression model,

$$f_i(\boldsymbol{\mu}) : \mathbb{R}^p \rightarrow \mathbb{R}, i \in [1, 2, \cdots k]. \tag{14}$$

ROMs using GPR allow for the number of expansion coefficients to be large as long as it does not degrade the quality of the solution subpsace; in the case of POD-based non-intrusive ROMs, $n = k$ usually provides the best accuracy when using GPR [16]. Given training data consisting of inputs $\boldsymbol{X} = [\boldsymbol{x}^1, \boldsymbol{x}^2 \cdots \boldsymbol{x}^n]$ and outputs $\boldsymbol{Y} = [y_1, y_2 \cdots y_n]$, where each input $\boldsymbol{x}^i \in \mathcal{P} \subset \mathbb{R}^p$ belongs to an input domain $\mathcal{P}$ corresponding to a single output $y_i \in \mathbb{R}$, a probability distribution over functions is inferred by GPR that is conditioned on the training data. A brief introduction to GPR is given in this section, while a more formal overview is given in the work of Rasmussen [17].

A Gaussian process (GP) is a set of random variables, of which any finite number follow a joint Gaussian distribution. GPR makes the assumption that data are generated according to a GP with a mean function $m(x)$ and covariance function κ,

$$f(x) \sim GP\left(m(x), \kappa(x, x^*)\right), \tag{15}$$

in addition to some Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma_y^2)$,

$$y = f(x) + \epsilon. \tag{16}$$

A joint Gaussian on the training data $\{\boldsymbol{X}, \boldsymbol{Y}\}$ and predictions at unseen points $\boldsymbol{x}^*$ are given by

$$\begin{bmatrix} \boldsymbol{Y} \\ f(\boldsymbol{x}^*) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(\boldsymbol{X}) \\ m(\boldsymbol{X}) \end{bmatrix}, \begin{bmatrix} \kappa(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2 I & \kappa(\boldsymbol{X}, \boldsymbol{x}^*) \\ \kappa(\boldsymbol{x}^*, \boldsymbol{X}) & \kappa(\boldsymbol{x}^*, \boldsymbol{x}^*) \end{bmatrix}\right). \tag{17}$$

The conditional expectation of $f(\boldsymbol{x}^*)$, derived from the properties of conditional Gaussian distributions, is given as

$$\mathbb{E}(f(\boldsymbol{x}^*)|\boldsymbol{Y}) = m(\boldsymbol{X}) + \kappa(\boldsymbol{x}^*, \boldsymbol{X})(\kappa(\boldsymbol{X}, \boldsymbol{X}) + \sigma_y^2 I)^{-1}\left(\boldsymbol{Y} - m(\boldsymbol{X})\right), \tag{18}$$

where I is the identity matrix and a small positive regularization constant σ_y^2 adds stability to the matrix inversion operation. The mean function $m(\boldsymbol{X})$ is in practice usually set to the mean of the training outputs

$$m(\boldsymbol{X}) = \frac{\sum_{i=1}^{n} y_i}{n}. \tag{19}$$

The inputs in $\boldsymbol{X}$ are also usually normalize before regression is performed, as this increases performance. There exist many $kernels$ that can be chosen to define the covariance function. The most well-known choice of kernel is the radial basis function (RBF) kernel

$$\kappa(\boldsymbol{X}, \boldsymbol{x}^*) = \text{exp}\left(-\frac{d(\boldsymbol{X}, \boldsymbol{x}^*)^2}{2l^2}\right). \tag{20}$$

Another choice of kernel, and the one that will be used in this work, is the Matern kernel,

$$\kappa(\boldsymbol{X}, \boldsymbol{x}^*) = \frac{1}{\Gamma(\nu)2^{\nu-1}}\left(\frac{\sqrt{2\nu}}{l}d(\boldsymbol{X}, \boldsymbol{x}^*)\right), \tag{21}$$

where d is the Euclidean distance function, Γ is the gamma function, and K_ν is the modified Bessel function of the second kind. l and ν comprise the set of hyperparameters θ of the Matern kernel, and control the length scale and smoothness respectively. The accuracy of the regression model is sensitive to the hyperparameter values, and an optimal set of hyperparameters θ_{opt} is found by using gradient-based optimizers to maximize the marginal log-likelihood of the training data

$$\theta_{\text{opt}} = \underset{\theta}{\text{argmax}} \log p(\boldsymbol{Y}|\boldsymbol{X}, \theta) = -\frac{1}{2}\boldsymbol{Y}^T(\kappa(\boldsymbol{X}, \boldsymbol{X}) + \sigma_y^2 I)^{-1} - \frac{1}{2}\text{log}\,|\kappa(\boldsymbol{X}, \boldsymbol{X}) + \sigma_y^2 I| - \frac{n}{2}\text{log}\,2\pi. \tag{22}$$

3. Local POD Basis Construction Using Isomap

In this section, the use of Isomap in constructing local POD bases is discussed. A brief introduction to the Isomap algorithm is given first, followed by the method used for selecting local POD bases for prediction points based on their design parameters. The original work Tenenbaum et al. [11] contains a more complete overview of the Isomap algorithm.

3.1. Isomap

Isomap is a nonlinear dimensionality reduction method that estimates an intrinsic low-dimensional manifold of high-dimensional data. Given a data matrix $\boldsymbol{X} \in \mathbb{R}^{n \times N}$ with n observations and dimensionality N, Isomap provides a *latent representation* matrix $\boldsymbol{W} \in \mathbb{R}^{n \times r}$, where $r \ll N$, and often $r = 2$. The latent representation $\boldsymbol{w} \in \mathbb{R}^r$ for each sample represents its position on the low-dimensional manifold. Isomap utilizes multidimensional scaling (MDS), a technique which preserves distances in a low-dimensional environment given a pairwise dissimilarity matrix of high-dimensional data points. The dissimilarity matrix can contain any metric of dissimilarity between points; in Isomap, this metric is related to the *geodesic distance* between points, which refers to the shortest distance along the manifold. Isomap first builds a neighborhood graph G from the training data, connecting only a given number of nearest neighbors for each observation, with the edges on the graph weighted by the Minkowski distance d between two points,

$$d(\boldsymbol{x}, \boldsymbol{y}) = \left(\sum_{i=1}^{N} |x_i - y_i|^q\right)^{\frac{1}{q}}. \tag{23}$$

When $q = 2$, this becomes the Euclidean distance. The number of nearest neighbors can be decided using various methods, including k-nearest-neighbors, kd trees, or ball trees [18]. The shortest path distances between all pairs of data points is then found by applying appropriate algorithms including Dijkstra's algorithm and the Floyd-Warshall algorithm [19] to G to obtain a dissimilarity matrix. MDS is then used to obtain a low-dimensional representation of the training data. Isomap is versatile algorithm for nonlinear dimensionality reduction, showing strong performance on datasets ranging from various images to gene and protein expressions [11, 20]. Alternative nonlinear dimensionality reduction techniques such as t-SNE and locally linear embedding (LLE) are also widely used in many fields. Although we do not compare these methods in this work, Isomap was shown to offer better performance for the ROM task. A popular benchmark for nonlinear dimensionality reduction methods is the Swiss roll dataset, an example of which is shown in Figure 2, where data points along the roll vary in color spectrally. A 2-dimensional latent representation

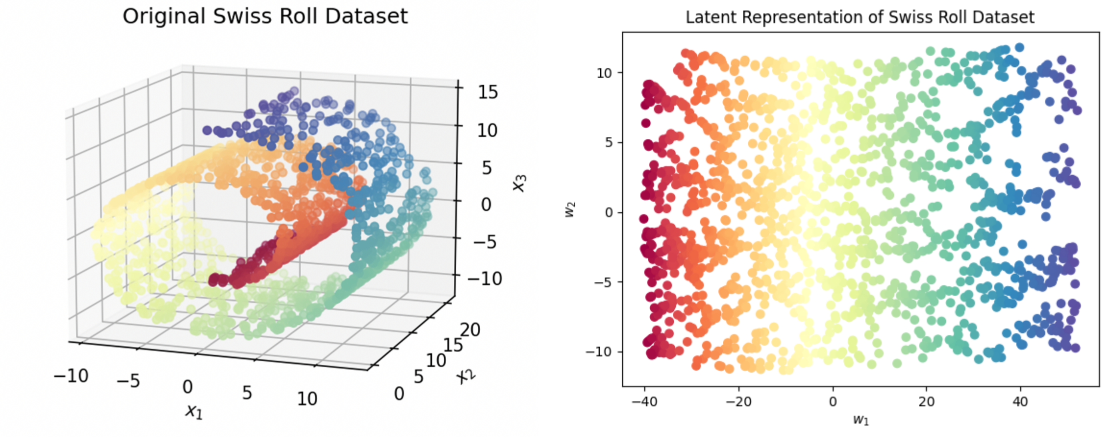


Figure 2: Swiss roll dataset and 2-dimensional latent representation produced by Isomap.

of the data produced by Isomap is also shown in Figure 2. The manifold produced by Isomap successfully separates points by their position along the roll by preserving their geodesic distances, rather than their Euclidean distances, as shown in Figure 3. While the Euclidean distance between points far away on the roll is small, their separation on the manifold produced by Isomap remains large.

3.2. Local POD Basis Construction

In constructing local POD bases, the goal is to use only training samples that will be physically similar to a given prediction point. The design parameters themselves may not be informative to this end, especially when the number of parameters is large. Applying Isomap to the transpose of the snapshot matrix, $\boldsymbol{S}^T$, we can estimate a low-dimensional manifold of our full-order solutions and obtain a latent representation $\boldsymbol{w}$ for each training sample. Using GPR, a regression model can be obtained to estimate the latent representations $\boldsymbol{w}^*$ for unseen design parameters $\boldsymbol{\mu}^*$. Guided by the general idea of nonlinear dimensionality reduction that points closer to each other in the lower dimensional space are more similar, we can create a local snapshot matrix $\boldsymbol{S}_L \in \mathbb{R}^{N\times l}$ for a prediction point with design parameters $\boldsymbol{\mu}^*$ consisting of the training samples corresponding to the l closest latent representations to $\boldsymbol{w}^*$ measured by Euclidean distance. For the Euclidean distance to be a strong metric of similarity, we set $r = 2$. The local POD basis $\boldsymbol{\Phi}_L \in \mathbb{R}^{N\times l}$ is found from the SVD of $\boldsymbol{S}_L$. Since ROMs using the POD-GPR method are most accurate when using all of the basis vectors, we choose $k = l$. Using this local snapshot matrix, the POD-GPR method can be used to make a prediction of the unknown state $\boldsymbol{x}^*$. This process is outlined in Algorithm 1.

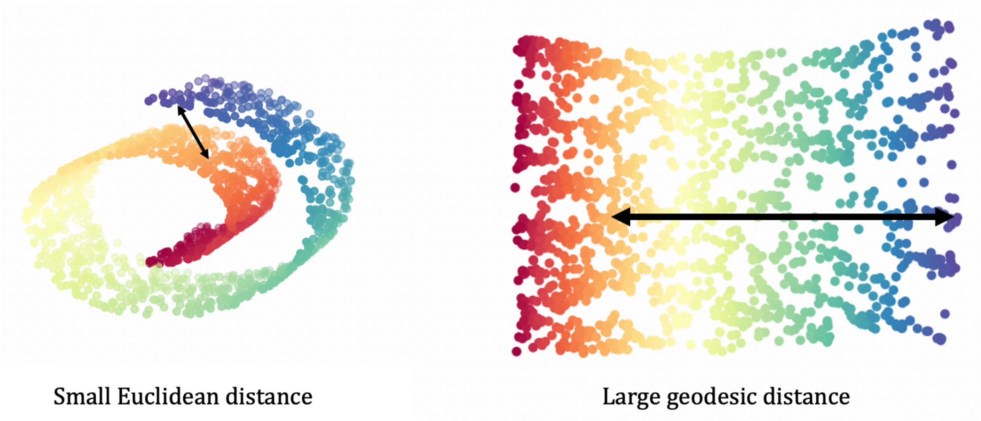
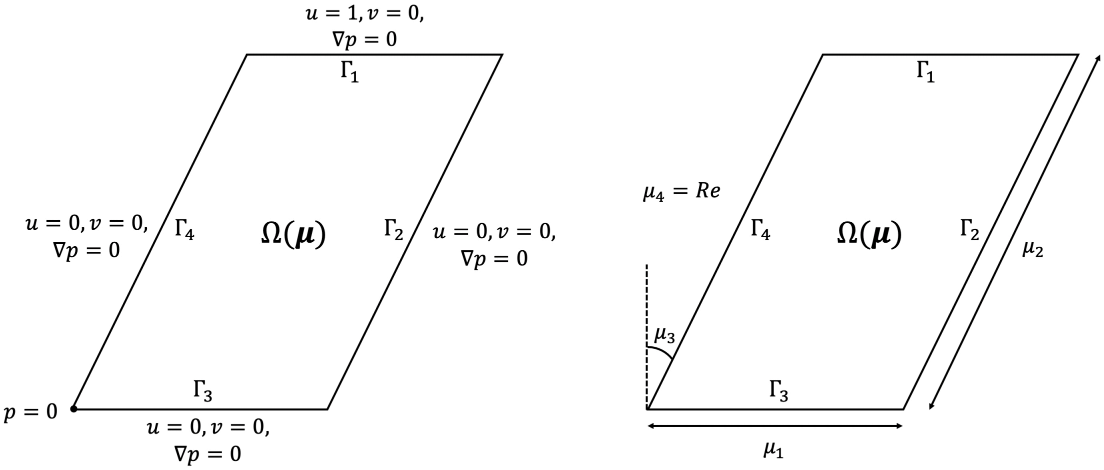


Figure 3: Comparison of Euclidean and geodesic distances between points for the Swiss roll dataset.

Algorithm 1 POD-GPR method using local POD bases found using Isomap

```
Input: $\boldsymbol{S}, \boldsymbol{\mathcal{U}}_{\text{train}}, \boldsymbol{\mathcal{U}}_{\text{pred}}, l$    ▷ Snapshot matrix, design parameter matrices, local POD basis dimension
Output: $\boldsymbol{S}^*$    ▷ Snapshot matrix containing predictions of full-order states

$k = l$    ▷ ROM dimension is equal to local POD basis dimension
$\boldsymbol{W} \leftarrow \text{Isomap}(\boldsymbol{S}^T)$    ▷ Compute latent representations of training snapshots using Isomap
$\boldsymbol{W}^* \leftarrow \text{GPR}(\boldsymbol{W}, \boldsymbol{\mathcal{U}}_{\text{train}}, \boldsymbol{\mathcal{U}}_{\text{test}})$    ▷ Approximate latent representations of prediction points using GPR
for $n \in \{1, 2, \ldots, n_{\text{pred}}\}$ do    ▷ Loop over all predictions
    $\boldsymbol{w}_n^* \leftarrow \boldsymbol{W}^*[n, :]$    ▷ Obtain approximate latent representation for current prediction
    $\boldsymbol{d}_n \leftarrow \text{dist}(\boldsymbol{w}_n^*, \boldsymbol{W})$    ▷ Compute pairwise Euclidean distances between latent representations
    $\boldsymbol{i}_n \leftarrow \text{sort}(\boldsymbol{d}_n)[1:k]$    ▷ Find indices of $l$ closest training points to current prediction
    $\boldsymbol{S}_L \leftarrow \boldsymbol{S}[:, \boldsymbol{i}_n]$    ▷ Assemble local snapshot matrix
    $\boldsymbol{\Phi}_L \leftarrow \text{SVD}(\boldsymbol{S}_L)$    ▷ Compute local POD basis from SVD of local snapshot matrix
    $\boldsymbol{A}_L \leftarrow \left(\boldsymbol{\Phi}_L^T \boldsymbol{S}_L\right)^T$    ▷ Compute expansion coefficients of states in local snapshot matrix
    for $m \in \{1, 2, \ldots, k\}$ do
        $\boldsymbol{A}_L^*[:, m] \leftarrow \text{GPR}(\boldsymbol{A}_L[:, m], \boldsymbol{\mathcal{U}}_{\text{train}}, \boldsymbol{\mathcal{U}}_{\text{test}})$    ▷ Approximate each expansion coefficient using GPR
$\boldsymbol{S}^* \leftarrow \left(\boldsymbol{\Phi}_L \boldsymbol{A}_L^{*T}\right)$    ▷ Compute approximated snapshot matrix of predictions
```



a) Lid-driven cavity boundary conditions. **b) Lid-driven cavity design parameters.**

Figure 4: Schematics describing the lid-driven cavity problem.

III. Numerical Results

The CFD case used in this work is a steady, incompressible, and laminar lid-driven cavity flow parameterized by three geometric parameters controlling the computational domain Ω and one parameter controlling the kinematic viscosity ν by changing the Reynolds number. This problem has previously appeared in a work by Hesthaven and Ubbiali [5], which also involved the use of non-intrusive ROMs. Figure 4 shows the boundary conditions on each edge $\Gamma_i, i \in [1, 2, 3, 4]$; u, $v = 0$ on all of the edges except the top, where $u = 1, v = 0$, and $\nabla p = 0$ on all of the edges. The reference pressure is set to 0 at the bottom left corner. The parameterization of the computational domain is alo shown, which involves three parameters changing the length of the horizontal (μ_1) and slanting edges (μ_2) as well as the slanting angle (μ_3). The fourth parameter is the Reynolds number, Re (μ_4), which is found as

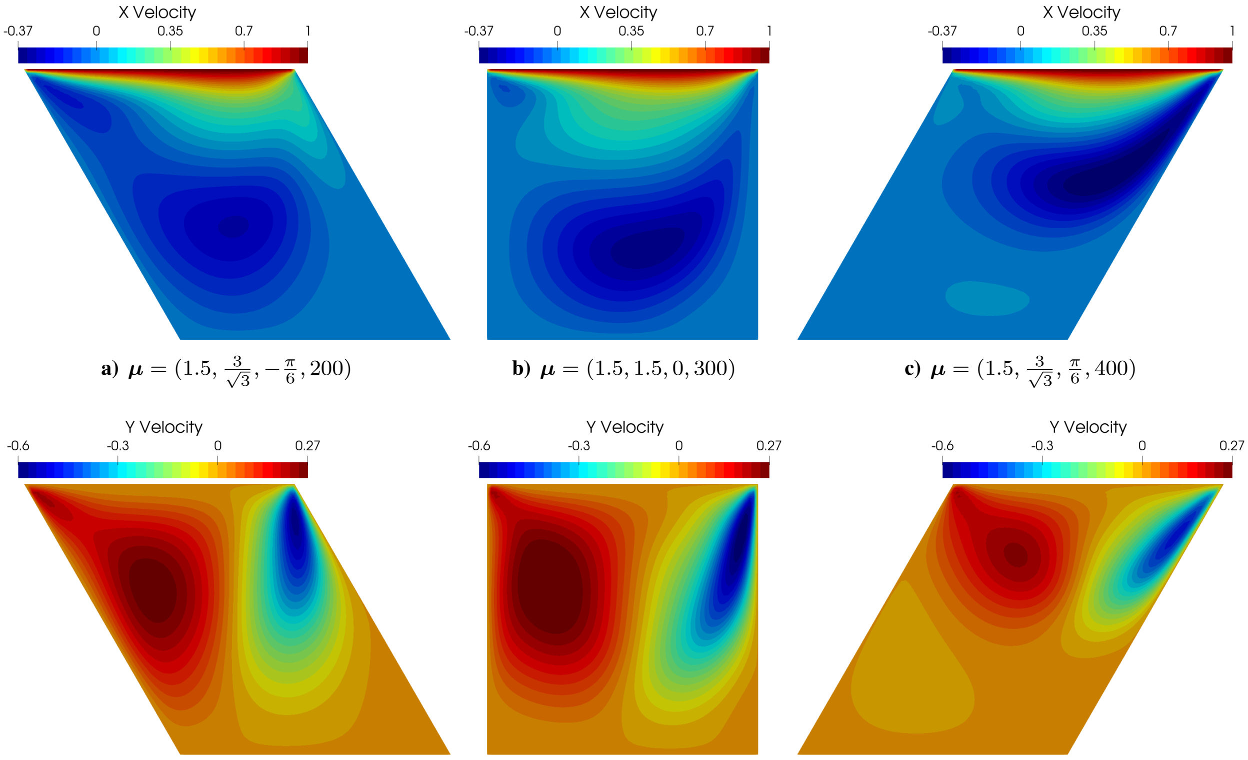
$$Re = \frac{\max(\mu_1, \mu_2)}{\nu(\boldsymbol{\mu})}. \tag{24}$$

Using the Latin hypercube sampling approach [21], 360 different sets of design parameters are generated with the bounds for each parameter given as

$$\begin{aligned}
\mu_1 &\in [1, 2], \\
\mu_2 &\in [1, 2], \\
\mu_3 &\in [-\frac{\pi}{4}, \frac{\pi}{4}], \\
\mu_4 &\in [100, 600].
\end{aligned}$$

The computational domain consists of 64×64 cells uniformly distributed in the x and y directions and one cell spanning the z direction, resulting in $N = 4096$. The full-order states of u and v will be used for testing the POD-GPR ROM using local POD bases, which will be referred to as the *local ROM*. Its performance will be tested against the ROM using all of the available training data and every basis vector, referred to as the *global ROM*. Separate ROMs are used for predicting u and v. Figure 5 shows contours of u and v for the lid-driven cavity problem at three different sets of design parameters. The metric of performance that will be used is the relative l^2 norm error e between the approximated state $\hat{\boldsymbol{x}}$ and true state $\boldsymbol{x}$

$$e = \frac{\sqrt{\sum_{i=1}^{N} (\hat{x}_i - x_i)^2}}{\sqrt{\sum_{i=1}^{N} x_i^2}}. \tag{25}$$



a) $\boldsymbol{\mu} = (1.5, \frac{3}{\sqrt{3}}, -\frac{\pi}{6}, 200)$

b) $\boldsymbol{\mu} = (1.5, 1.5, 0, 300)$

c) $\boldsymbol{\mu} = (1.5, \frac{3}{\sqrt{3}}, \frac{\pi}{6}, 400)$

d) $\boldsymbol{\mu} = (1.5, \frac{3}{\sqrt{3}}, -\frac{\pi}{6}, 200)$

e) $\boldsymbol{\mu} = (1.5, 1.5, 0, 300)$

f) $\boldsymbol{\mu} = (1.5, \frac{3}{\sqrt{3}}, \frac{\pi}{6}, 400)$

Figure 5: Contours of u (top) and v (bottom) for the lid-driven cavity problem at three sets of design parameters.

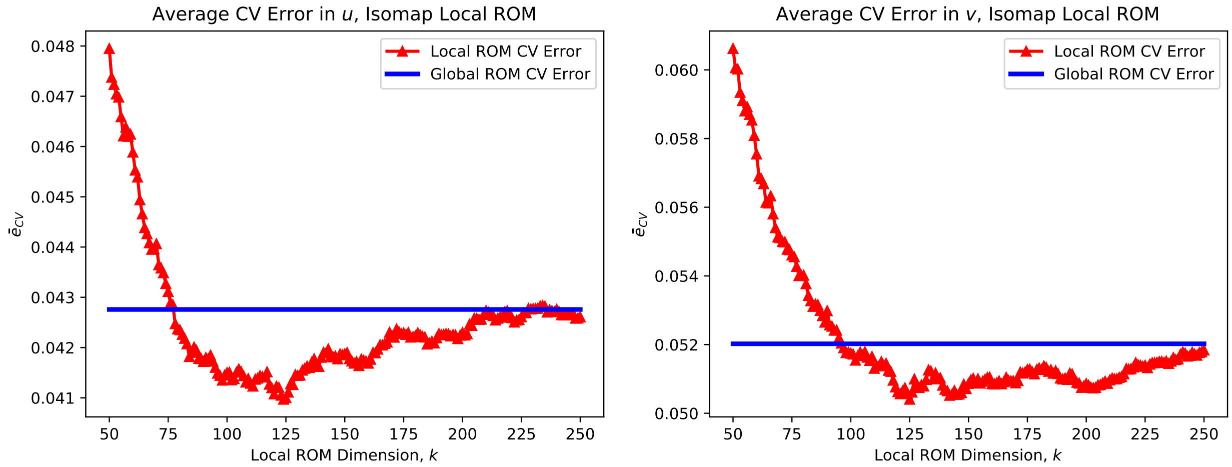
To assess the performance of the ROM over the entire dataset, 5-fold cross-validation will be used to create 5 different splits of the data into 288 training samples and 72 test samples. For each fold, the average relative error $\bar{e}_i, i \in [1, 2, 3, 4, 5]$ will be taken over all of the test samples. The mean of these average errors will be taken over all of the folds to report a cross-validation error $\bar{e}_{\text{CV}}$

$$\bar{e}_{\text{CV}} = \frac{\sum_{i=1}^{5} \bar{e}_i}{5}. \tag{26}$$

In addition to using local POD bases found using Isomap, results will be shown for local POD bases found randomly from the training set, where each test sample will be assigned a randomly selected local snapshot matrix. Isomap and GPR are implented using scikit-learn [22], an open-source machine learning library for Python. For Isomap, the Minkowski distance with $q = 5$ is used, the manifold dimension is set to $r = 2$, and the number of neighbors to consider is set to 5. The scikit-learn implementation of Isomap automatically selects an algorithm to use for deciding the number of nearest neighbors and computing the shortest path distances. Figure 5 shows contours of u and v for the lid-driven cavity problem at three different sets of design parameters. A sharp gradient in u exists at the top of the cavity and a vortex moves throughout the cavity as the design parameters change. The contours of v show a vortex which varies in shape and size with the design parameters. Both u and v exhibit very nonlinear behavior with changes in $\boldsymbol{\mu}$, making this a difficult prediction problem.

Figures 6 and 7 show plots of $\bar{e}_{\text{CV}}$ against the ROM dimension $k \in [50, 250]$ for the Isomap and random local ROMs respectively. The global ROM relative cross-validation error with $k = 288$ is also plotted for comparison. Compared to the global ROM, we can see that using Isomap to generate local POD bases offers better predictive performance in both u and v over a large number of ROM dimensions. When k is low, the amount of data available may not be sufficient in both the quality of the POD basis and GPR model to outperform the global ROM. As k grows larger, the performance offered by using local POD bases is greater than that of the global ROM, with peak performance occuring at around $k = 125$ for both u and v, with percent decreases in $\bar{e}_{\text{CV}}$ of 4.11% and 3.11% respectively. Generating random local POD bases does not exhibit this behavior; we can see that the global ROM outperforms this method over all selected values of k, especially when k is low. The random local ROM error approaches that of the global ROM as k grows large, with the decay in $\bar{e}_{\text{CV}}$ showing exponential behavior. This great disparity in performance at lower values of k shows that Isomap is highly effective in identifying training samples that are very similar to the current test sample based on their latent representations when combined with GPR. Although the local POD basis $\boldsymbol{\Psi}_L$ does not offer a lower projection error compared to $\boldsymbol{\Psi}$, the GPR model is more robust when working with states that show a higher degree of similarity amongst themselves.

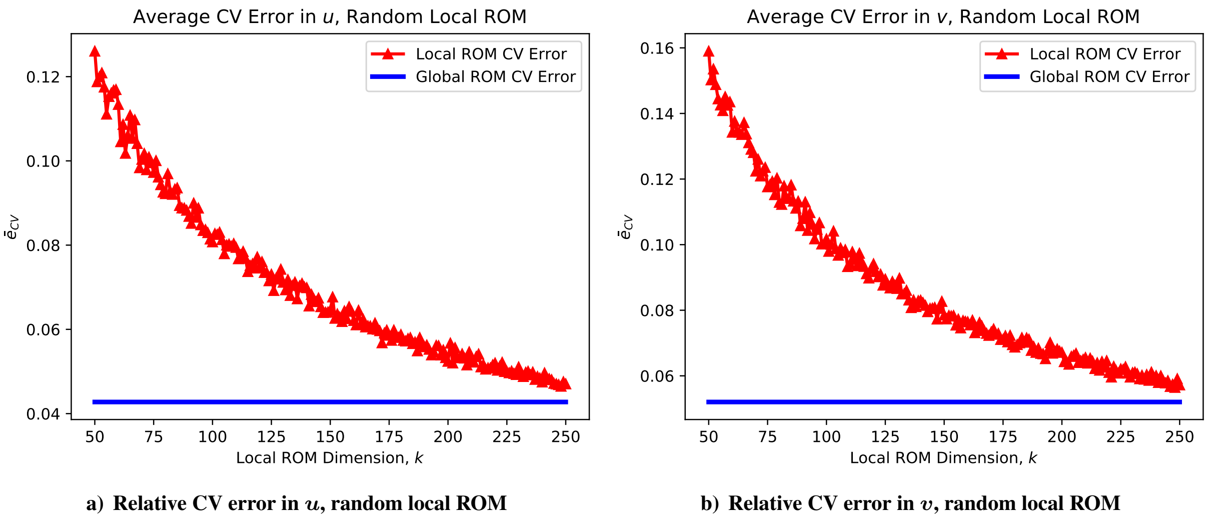
Although $\bar{e}_{\text{CV}}$ is lower for most of the selected values of k when using the Isomap local ROM, not all test samples will exhibit a decrease in relative error when using this method compared to the global ROM. The difference in relative error for test samples between the local and global ROMs will vary based on a number of factors including the values of $\boldsymbol{\mu}^*$, $\boldsymbol{\Psi}_L$, and the GPR model. There is no clear way of distinguishing which ROM will perform better given only $\boldsymbol{\mu}^*$. Figures 8 and 9 show two cases where the global ROM results in high relative errors and the Isomap local ROM reduces them significantly for both u and v. Although the values of μ_3 and Re are similar for these two cases, the contours of u and v show that they do not exhibit the same physical behavior. Figure 8 shows that a local ROM with $k = 125$ reduces the relative errors in both u and v significantly; with percent decreases in e of 40.6% and 47.6% respectively in the first case. For the second case, Figure 9 shows that a local ROM with $k = 150$ gives percent decreases in e for u and v of 27.9% and 30.8% respectively.



a) Relative CV error in u, Isomap local ROM

b) Relative CV error in v, Isomap local ROM

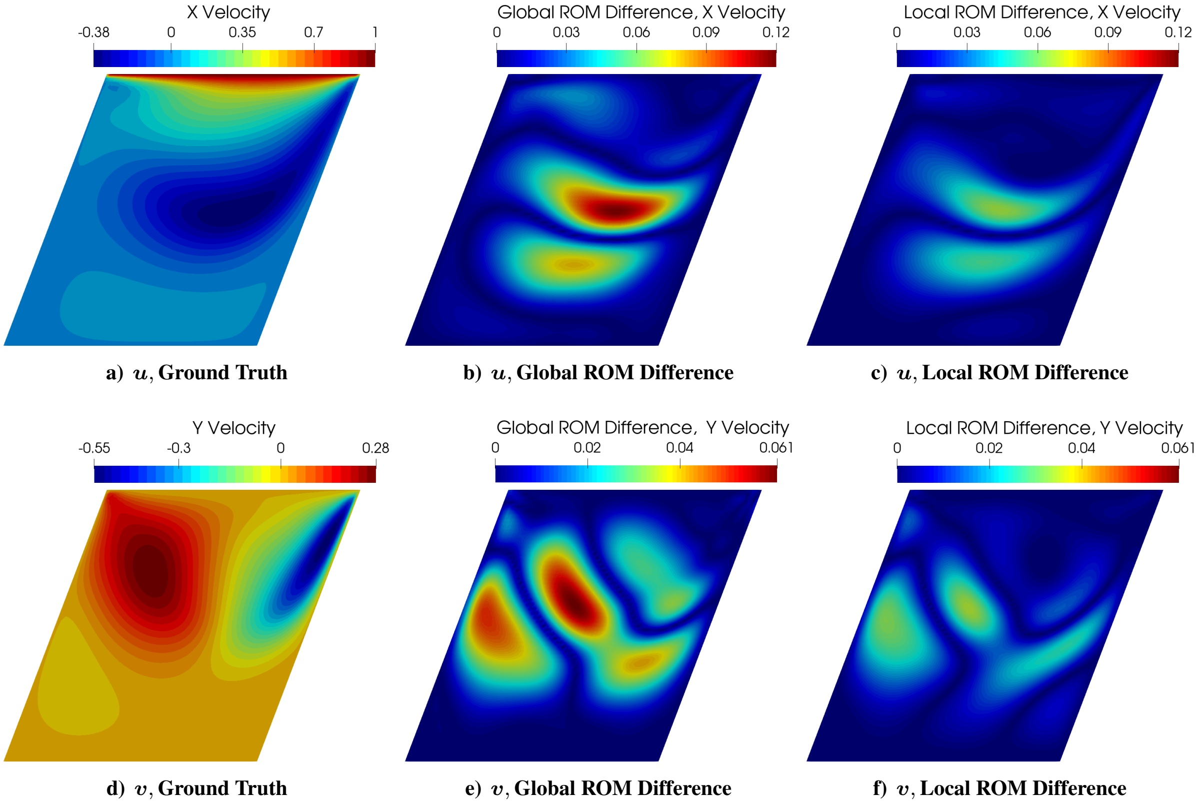
Figure 6: Relative cross-validation error plots for Isomap local ROM.



a) Relative CV error in u, random local ROM

b) Relative CV error in v, random local ROM

Figure 7: Relative cross-validation error plots for random local ROM.



a) u, Ground Truth

b) u, Global ROM Difference

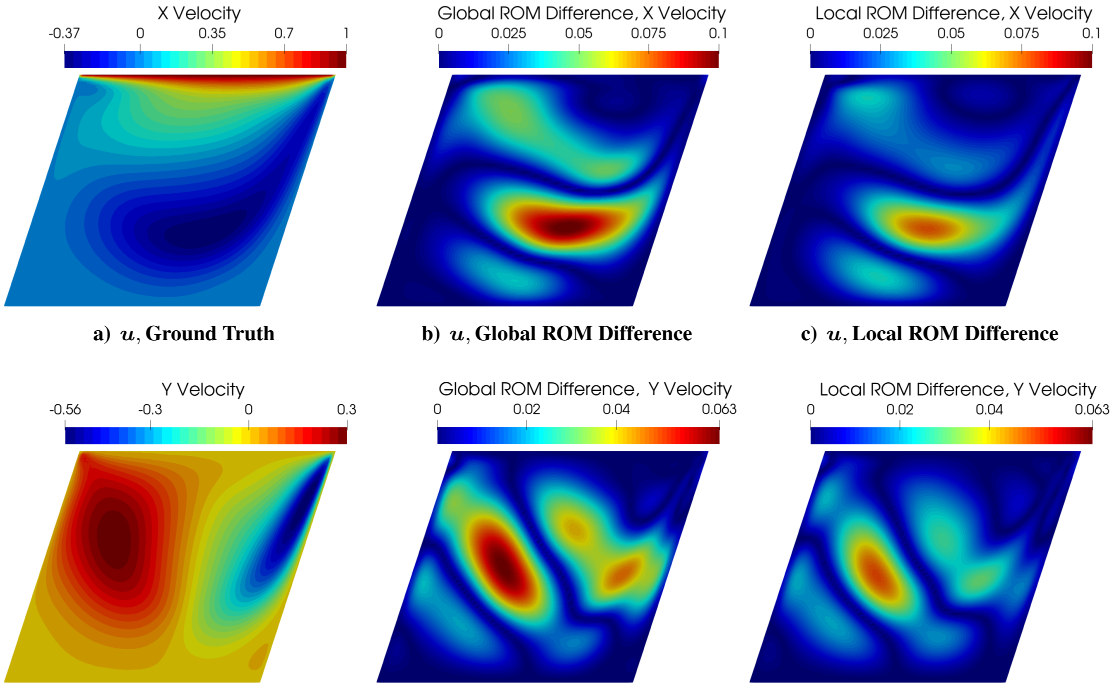
c) u, Local ROM Difference

d) v, Ground Truth

e) v, Global ROM Difference

f) v, Local ROM Difference

Figure 8: Comparison between global and Isomap local ROMs at $\boldsymbol{\mu} = (1.601, 1.832, 0.3643, 543.8)$ and $k = 125$.



a) u**, Ground Truth**
b) u**, Global ROM Difference**
c) u**, Local ROM Difference**

d) v**, Ground Truth**
e) v**, Global ROM Difference**
f) v**, Local ROM Difference**

Figure 9: Comparison between global and Isomap local ROMs at $\boldsymbol{\mu} = (1.957, 1.860, 0.3163, 513.2)$ and $k = 150$.

IV. Conclusions

This work presents a non-intrusive ROM utilizing the POD-GPR method based on local POD bases found using Isomap. GPR can be used to estimate the latent representations of unseen design parameters, with local POD bases generated from the closest training samples measured by Euclidean distance. When applied to a lid-driven cavity problem, this method shows greater accuracy for many local ROM dimensions, and greatly exceeds the performance found from selecting local POD bases randomly. Isomap is found to be highly effective in creating latent representations of the training data, with samples that are physically similar being closer together on the low-dimensional manifold. Local POD bases consisting of physically similar samples are shown to allow for more robust prediction of expansion coefficients when using GPR, even though the local POD bases do not provide a lower projection error. Although this method does show an increase in overall performance, it is not possible to tell if a given set of unseen design parameters will gain any boost in predictive performance. Future work will focus on increasing the utility of this method by developing a method for finding an optimal local ROM dimension to use for a set of training snapshots.

References

[1] Berkooz, G., Holmes, P., and Lumley, J., "The Proper Orthogonal Decomposition in the Analysis of Turbulent Flows," *Annual Review of Fluid Mechanics*, Vol. 25, 11 2003, pp. 539–575. doi:10.1146/annurev.fl.25.010193.002543.

[2] Holmes, P., Lumley, J. L., Berkooz, G., Mattingly, J. C., and Wittenberg, R. W., "Low-dimensional models of coherent structures in turbulence," *Physics Reports*, Vol. 287, 1997, pp. 337–384.

[3] Carlberg, K., Barone, M., and Antil, H., "Galerkin v. least-squares Petrov-Galerkin projection in nonlinear model reduction," *Journal of Computational Physics*, Vol. 330, 2017, pp. 693–734. doi:https://doi.org/10.1016/j.jcp.2016.10.033.

[4] Benner, P., Gugercin, S., and Willcox, K., "A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems." *SIAM Rev.*, Vol. 57, No. 4, 2015, pp. 483–531.

[5] Hesthaven, J. and Ubbiali, S., "Non-intrusive reduced order modeling of nonlinear problems using neural networks," *Journal of Computational Physics*, Vol. 363, 2018, pp. 55–78. doi:https://doi.org/10.1016/j.jcp.2018.02.037.

[6] Xiao, D., Fang, F., Buchan, A., Pain, C., Navon, I., and Muggeridge, A., "Non-intrusive reduced order modelling of the Navier-Stokes equations," *Computer Methods in Applied Mechanics and Engineering*, Vol. 293, 2015, pp. 522–541. doi:https://doi.org/10.1016/j.cma.2015.05.015.

[7] Xiao, M., Breitkopf, P., Coelho, R., Knopf-Lenoir, C., Sidorkiewicz, M., and Villon, P., "Model reduction by CPOD and Kriging: Application to the shape optimization of an intake port," *Structural and Multidisciplinary Optimization*, Vol. 41, 04 2010. doi:10.1007/s00158-009-0434-9.

[8] Guo, M. and Hesthaven, J. S., "Reduced order modeling for nonlinear structural analysis using Gaussian process regression," *Computer methods in applied mechanics and engineering*, Vol. 341, 2018, pp. 807–826.

[9] Dupuis, R., Jouhaud, J.-C., and Sagaut, P., "Surrogate Modeling of Aerodynamic Simulations for Multiple Operating Conditions Using Machine Learning," *AIAA Journal*, Vol. 56, No. 9, 2018, pp. 3622–3635. doi:10.2514/1.J056405.

[10] Amsallem, D., Zahr, M. J., and Farhat, C., "Nonlinear model order reduction based on local reduced-order bases," *International Journal for Numerical Methods in Engineering*, Vol. 92, No. 10, 2012, pp. 891–916. doi:https://doi.org/10.1002/nme.4371.

[11] Tenenbaum, J. B., De Silva, V., and Langford, J. C., "A global geometric framework for nonlinear dimensionality reduction," *science*, Vol. 290, No. 5500, 2000, pp. 2319–2323.

[12] Weller, H. G., Tabor, G., Jasak, H., and Fureby, C., "A tensorial approach to computational continuum mechanics using object-oriented techniques," *Computers in physics*, Vol. 12, No. 6, 1998, pp. 620–631.

[13] Patankar, S. V. and Spalding, D. B., "A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows," *International Journal of Heat and Mass Transfer*, Vol. 15, No. 10, 1972, pp. 1787–1806. doi:10.1016/0017-9310(72)90054-3.

[14] Rhie, C. and Chow, W. L., "Numerical study of the turbulent flow past an airfoil with trailing edge separation," *AIAA Journal*, Vol. 21, No. 11, 1983, pp. 1525–1532. doi:10.2514/3.8284.

[15] Krige, D., *A statistical approach to some mine valuation and allied problems on the Witwatersrand*, Ph.D. thesis, University of the Witwatersrand, 1951.

[16] Mrosek, M., Othmer, C., and Radespiel, R., "Reduced-Order Modeling of Vehicle Aerodynamics via Proper Orthogonal Decomposition," *SAE International Journal of Passenger Cars - Mechanical Systems*, Vol. 12, No. 3, oct 2019, pp. 225–236. doi:https://doi.org/10.4271/06-12-03-0016.

[17] Rasmussen, C. E., "Gaussian processes in machine learning," *Summer school on machine learning*, Springer, 2003, pp. 63–71.

[18] Abbasifard, M. R., Ghahremani, B., and Naderi, H., "A survey on nearest neighbor search methods," *International Journal of Computer Applications*, Vol. 95, No. 25, 2014.

[19] Magzhan, K. and Jani, H. M., "A review and evaluations of shortest path algorithms," *International journal of scientific & technology research*, Vol. 2, No. 6, 2013, pp. 99–104.

[20] Lee, G., Rodriguez, C., and Madabhushi, A., "Investigating the efficacy of nonlinear dimensionality reduction schemes in classifying gene and protein expression studies," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, Vol. 5, No. 3, 2008, pp. 368–384.

[21] McKay, M. D., Beckman, R. J., and Conover, W. J., "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, Vol. 21, No. 2, 1979, pp. 239–245.

[22] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, Vol. 12, 2011, pp. 2825–2830.