Computer Science Technical Report CSTR-1
January 16, 2017

Azam Moosavi, Razvan Stefanescu, Adrian Sandu

"Multivariate predictions of local reduced-order-model errors and dimensions "

Computational Science Laboratory
Computer Science Department
Virginia Polytechnic Institute and State University
Blacksburg, VA 24060
Phone: (540)-231-2193
Fax: (540)-231-6075
Email: razvan.stefanescu@spire.com
Web: http://csl.cs.vt.edu

COMPUTATIONAL SCIENCE LABORATORY

Innovative Computational Solutions

VirginiaTech
Invent the Future

Abstract

This paper introduces multivariate input-output models to predict the errors and bases dimensions of local parametric Proper Orthogonal Decomposition reduced-order models. We refer to these multivariate mappings as the MP-LROM models. We employ Gaussian Processes and Artificial Neural Networks to construct approximations of these multivariate mappings. Numerical results with a viscous Burgers model illustrate the performance and potential of the machine learning based regression MP-LROM models to approximate the characteristics of parametric local reduced-order models. The predicted reduced-order models errors are compared against the multi-fidelity correction and reduced order model error surrogates methods predictions, whereas the predicted reduced-order dimensions are tested against the standard method based on the spectrum of snapshots matrix. Since the MP-LROM models incorporate more features and elements to construct the probabilistic mappings they achieve more accurate results. However, for high-dimensional parametric spaces, the MP-LROM models might suffer from the curse of dimensionality. Scalability challenges of MP-LROM models and the feasible ways of addressing them are also discussed in this study.

keywords   Local reduced-order models, Proper Orthogonal Decomposition, regression machine learning techniques.

1   Introduction

Many physical phenomena are described mathematically by partial differential equations (PDEs), and, after applying suitable discretization schemes, are simulated on a computer. PDE-based models frequently require calibration and parameter tuning in order to provide realistic simulation results. Recent developments in the field of uncertainty quantification [42, 71, 27, 13] provide the necessary tools for validation of such models even in the context of variability and lack of knowledge of the input parameters. Techniques to propagate uncertainties through models include direct evaluation for linearly parametric models, sampling methods such as Monte Carlo [66], Latin hypercube [32] and quasi-Monte Carlo techniques [43], perturbation methods [10, 11, 12] and spectral representation [42, 21, 1]. While stochastic Galerkin methods [42] are intrusive in nature, Monte Carlo sampling methods [66] and stochastic collocations [21] do not require the modification of existing codes and hence they are non-intrusive. While uncertainty propagation techniques can measure the impact of uncertain parameters on some quantities of interest, they often become infeasible due to the large number of model realizations requirement. Similar difficulties are encountered when solving Bayesian inference problems since sampling from posterior distribution is required.

The need for computational efficiency motivated the development of surrogate models such as response surfaces, low resolution, and reduced-order models. Data fitting or response surface models [71] are data-driven models. The underlying physics remain unknown and only the input-output behavior of the model is considered. Data fitting can use techniques such as regression, interpolation, radial basis function, Gaussian Processes, Artificial Neural Networks and other supervised machine-learning methods. The latter techniques can automatically detect patterns in data, and one can use them to predict future data under uncertainty in a probabilistic framework [48]. While easy to implement due to the non-intrusive nature, the prediction abilities may suffer since the governing physics are not specifically accounted for.

Low-fidelity models attempt to reduce the computational burden of the high-fidelity models by neglecting some of the physical aspects (e.g., replacing Navier-Stokes and Large Eddy Simulations with inviscid Euler's equations and Reynolds-Averaged Navier-Stokes [25, 63, 78], or decreasing the spatial resolution [15, 74]). The additional approximations, however, may considerably degrade the physical solution with only a modest decrease of the computational load.

Reduced basis [53, 5, 26, 61, 18] and Proper Orthogonal Decomposition [39, 45, 36, 46, 47] are two of the popular reduced-order modeling (ROM) strategies available in the literature. Data analysis is conducted to extract basis functions from experimental data or detailed simulations of high-dimensional systems (method of snapshots [67, 68, 69]), for subsequent use in Galerkin projections that yield low dimensional dynamical models. While these type of models are physics-based and therefore require intrusive implementations, they are usually more robust than data fitting and low-fidelity models. However, since surrogate model robustness depends heavily on the problem, it must be carefully analyzed especially for large-scale nonlinear dynamical systems.

ROM robustness in a parametric setting can be achieved by constructing a global basis [34, 54], but this strategy generates large dimensional bases that may lead to slow reduced-order models. Local approaches have been designed for parametric or time domains generating local bases for both the state variables [56, 17] and non-linear terms [20, 52]. A recent survey of state-of-the-art methods in projection-based parametric model reduction is available in [6].

In this study, we propose multivariate data fitting models to predict the local parametric Proper Orthogonal Decomposition reduced-order models errors and bases dimensions. We refer to them as MP-LROM models. Let us consider a local parametric reduced-order model of dimension K_{POD} constructed using a high-fidelity solution associated with the parameter configuration μ_p.

Our first MP-LROM model consists in the mapping $\{\mu, \mu_p, K_{POD}\} \mapsto \log \varepsilon^{HF}_{\mu, \mu_p, K_{POD}}$, where $\varepsilon^{HF}_{\mu, \mu_p, K_{POD}}$ is the error of the local reduced-order model solution with respect to the high-fidelity solution for a viscosity parameter configuration μ. Our proposed approach is inspired from the multi-fidelity correction (MFC) [2] and reduced order model error surrogates method (ROMES) [19]. MFC [2, 22, 25, 37] has been developed for low-fidelity models in the context of optimization. The MFC model simulates the input-output relation $\mu \mapsto \varepsilon^{HF}_{\mu}$, where ε^{HF}_{μ} is the low-fidelity model error depending on a global reduced basis with a constant reduced-order model dimension. The ROMES method [19] introduced the concept of error indicators for global reduced-order models and generalized the MFC framework by approximating the mapping $\rho(\mu) \mapsto \log \varepsilon^{HF}_{\mu}$. The error indicators $\rho(\mu)$ include rigorous error bounds and reduced-order residual norms. No variation of the reduced basis dimension was taken into account. By estimating the log of the reduced-order

2

model error instead of the error itself, the input-output map exhibits a lower variance as shown by our numerical experiments as well as those in [19].

The second proposed MP-LROM model addresses the issue of a-priori selection of the reduced basis dimension for a prescribed accuracy of the reduced solution. The standard approach is to analyze the spectrum of the snapshots matrix, and use the largest singular value removed from the expansion to estimate the accuracy level [75]. To also take into account the error due to the full-order-model equations projection in the reduced space, here we propose the mapping $\{\mu_p, \log \varepsilon^{HF}_{\mu_p, \mu_p, K_{POD}}\} \mapsto K_{POD}$ to predict the dimension of a local parametric reduced-order model given a prescribed error threshold.

To approximate the mappings $\{\mu, \mu_p, K_{POD}\} \mapsto \log \varepsilon^{HF}_{\mu, \mu_p, K_{POD}}$ and $\{\mu_p, \log \varepsilon^{HF}_{\mu_p, \mu_p, K_{POD}}\} \mapsto K_{POD}$, we propose regression models constructed using Gaussian Processes (GP) [70, 44] and Artificial Neural Networks (ANN). In the case of one dimensional Burgers model, the resulted MP-LROM error models are accurate and their predictions are compared against those obtained by the MFC and ROMES models. The predicted dimensions of local reduced-order models using our proposed MP-LROM models are more accurate than those derived using the standard method based on the spectrum of snapshots matrix.

The remainder of the paper is organized as follows. Section 2 reviews the reduced-order modeling parametric framework. The MP-LROM models and the regression machine learning methods used in this study to approximate the MP-LROM mappings are described in details in Section 3. Section 4 describes the viscous 1D-Burgers model and compares the performances of the MP-LROM and state of the art models. Conclusions are drawn in Section 5.

2 Parametric reduced-order modeling

Proper Orthogonal Decomposition has been successfully applied in numerous applications such as compressible flow [?] and computational fluid dynamics [40, 60, 79], to mention a few. It can be thought of as a Galerkin approximation in the state variable built from functions corresponding to the solution of the physical system at specified time instances. A system reduction strategy for Galerkin models of fluid flows based on a partition in slow, dominant, and fast modes, has been proposed in [51]. Closure models and stabilization strategies for POD of turbulent flows have been investigated in [64, 77].

In this paper we consider discrete inner products (Euclidean dot product), though continuous products may be employed as well. Generally, an unsteady problem can be written in semi-discrete form as an initial value problem; i.e., as a system of nonlinear ordinary differential equations

$$\frac{d\mathbf{x}(\mu, t)}{dt} = \mathbf{F}(\mathbf{x}, t, \mu), \quad \mathbf{x}(\mu, 0) = \mathbf{x}_0 \in \mathbb{R}^{N_{\text{state}}}, \quad \mu \in \mathcal{P}. \tag{1}$$

The input-parameter μ typically characterizes the physical properties of the flow. By $\mathcal{P}$ we denote the input-parameter space. For a given parameter configuration μ_p we select an ensemble of N_t time instances of the flow $\mathbf{x}(\mu_p, t_1), \ldots, \mathbf{x}(\mu_p, t_{N_t}) \in \mathbb{R}^{N_{\text{state}}}$, where N_{state} is the total number of discrete model variables, and $N_t \in \mathbb{N}^*$. The POD method chooses an orthonormal basis $U_{\mu_p} = [\mathbf{u}_1^{\mu_p} \cdots \mathbf{u}_{K_{POD}}^{\mu_p}] \in \mathbb{R}^{N_{\text{state}} \times K_{POD}}$, such that the mean square error between $\mathbf{x}(\mu_p, t_i)$ and the POD expansion $\mathbf{x}_{\mu_p}^{\text{POD}}(t_i) = U_{\mu_p} \tilde{\mathbf{x}}_{\mu_p}(\mu, t_i)$, $\tilde{\mathbf{x}}_{\mu_p}(\mu, t_i) = U_{\mu_p}^T \mathbf{x}(\mu_p, t_i) \in \mathbb{R}^{K_{POD}}$, is minimized on average. The POD space dimension $K_{POD} \ll N_{\text{state}}$ is appropriately chosen to capture the dynamics of the flow. Algorithm 1 describes the reduced-order basis construction procedure [72].

Algorithm 1 POD basis construction

1: Compute the singular value decomposition for the snapshots matrix $[\mathbf{x}(\mu_p, t_1) \cdots \mathbf{x}(\mu_p, t_{N_t})] = \bar{U}_{\mu_p} \Sigma_{\mu_p} \bar{V}_{\mu_p}^T$, with the singular vectors matrix $\bar{U}_{\mu_p} = [\mathbf{u}_1^{\mu_p} \cdots \mathbf{u}_{N_t}^{\mu_p}]$.

2: Using the singular-values $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{N_t} \geq 0$ stored in the diagonal matrix Σ_{μ_p}, define $I(m) = \sum_{i=1}^{m} \lambda_i^2 / (\sum_{i=1}^{t_{N_t}} \lambda_i^2)$.

3: Choose K_{POD}, the dimension of the POD basis, such that $K_{POD} = \arg \min_m \{I(m) : I(m) \geq \gamma\}$ where $0 \leq \gamma \leq 1$ is the percentage of total information captured by the reduced space $\mathcal{X}^{K_{POD}} = \text{range}(U_{\mu_p})$. It is common to select $\gamma = 0.99$. The basis U_{μ_p} consists of the first K_{POD} columns of $\bar{U}_{\mu_p}$.

Next, a Galerkin projection of the full model state (1) onto the space $\mathcal{X}^{K_{POD}}$ spanned by the POD basis elements is used to obtain the reduced-order model

$$\frac{d\tilde{\mathbf{x}}_{\mu_p}(\mu, t)}{dt} = U_{\mu_p}^T \mathbf{F}\left(U_{\mu_p} \tilde{\mathbf{x}}_{\mu_p}(\mu, t), t, \mu\right), \quad \tilde{\mathbf{x}}_{\mu_p}(\mu, 0) = U_{\mu_p}^T \mathbf{x}_0. \tag{2}$$

The notation $\tilde{\mathbf{x}}_{\mu_p}(\mu, t)$ expresses the solution dependence on the varying parameter μ and also on μ_p the configuration whose associated high-fidelity trajectory was employed to generate the POD basis. While being accurate for $\mu = \mu_p$, the reduced model (2) may lose accuracy when moving away from the initial setting. Several strategies have been proposed to derive a basis that spans the entire parameter space. These include the reduced basis method combined with the use of error estimates [61, 55, 54], global POD [73, 65], Krylov-based sampling methods [16, 76], and greedy techniques [29, 50]. The fundamental assumption used by these approaches is that a smooth low-dimensional global manifold characterizes the model solutions over the entire parameter domain. The purpose of our paper is to estimate the solution error and dimension of the reduced-order model (2) that can be subsequently used to generate a global basis for the parameter space.

3 Multivariate prediction of local reduced-order models characteristics (MP-LROM)

We propose multivariate input-output models

$$\phi : \mathbf{z} \mapsto y, \tag{3}$$

$\mathbf{z} \in \mathbb{R}^r$, to predict characteristics $y \in \mathbb{R}$ of local parametric reduced-order models (2).

3.1 Error Model

Inspired from the MFC and ROMES methodologies we introduce an input-output model to predict the level of error $\varepsilon_{\mu,\mu_p,K_{POD}}^{HF}$, where

$$\varepsilon_{\mu,\mu_p,K_{POD}}^{HF} = \\ \|\mathbf{x}(\mu,t_1) - U_{\mu_p}\tilde{\mathbf{x}}_{\mu_p}(\mu,t_1) \quad \mathbf{x}(\mu,t_2) - U_{\mu_p}\tilde{\mathbf{x}}_{\mu_p}(\mu,t_2) \quad \cdots \quad \mathbf{x}(\mu,t_{N_t}) - U_{\mu_p}\tilde{\mathbf{x}}_{\mu_p}(\mu,t_{N_t})\|_F. \tag{4}$$

Here $\| \cdot \|_F$ denotes the Frobenius norm, and K_{POD} is the dimension of the reduced-order model. In contrast with ROMES and MFC models that predict the error of global reduced-order models with fixed dimensions, using univariate functions, here we propose a multivariate model

$$\phi_{MP-LROM}^e : \{\mu, \mu_p, K_{POD}\} \mapsto \log \varepsilon_{\mu,\mu_p,K_{POD}}^{HF} \tag{5}$$

to predict the error of local parametric reduced-order models (2) of various dimensions. Since the dimension of basis usually influences the level of error we include it among the input variables. To design models with reduced variances we look to approximate the logarithm of the error as suggested in [19].

For high-dimensional parametric spaces, ROMES method handles well the curse of dimensionality with their proposing univariate models. In combination with active subspace method [14], we can reduce the number of input variables in case the amount of variability in the parametric space is mild. This will increase our error model feasibility even for high-dimensional parametric space.

3.2 Dimension of the reduced basis

The basis dimension represents one of the most important characteristic of a reduced-order model. The reduced manifold dimension directly affects both the on-line computational complexity of the reduced-order model and its accuracy [41, 33, 23]. By increasing the dimension of the basis, the projection error usually decreases and the accuracy of the reduced-order model is enhanced. However this is not necessarily valid as seen in [59, Section 5]. Nevertheless the spectrum of the snapshots matrix offers guidance regarding the choice of the reduced basis dimension when some prescribed reduced-order model error is desired. However the accuracy depends also on the 'in-plane' error, which is due to the fact that the full-order-model equations are projected on the reduced subspace [58, 35].

We seek to predict the dimension of the local parametric reduced-order model (2) by accounting for both the orthogonal projection error onto the subspace, which is computable by the sum of squares of singular values, and the 'in-plane' error. As such we propose to model the mapping

$$\phi_{MP-LROM}^d : \{\mu_p, \log \varepsilon_{\mu_p,\mu_p,K_{POD}}^{HF}\} \mapsto K_{POD}. \tag{6}$$

Once such model is available, given a positive threshold $\bar{\varepsilon}$ and a parametric configuration μ_p, we will be able to predict the dimension K_{POD} of the basis U_{μ_p}, such that the reduced-order model error satisfies

$$\|\mathbf{x}(\mu_p,t_1) - U_{\mu_p}\tilde{\mathbf{x}}_{\mu_p}(\mu_p,t_1) \quad \mathbf{x}(\mu_p,t_2) - U_{\mu_p}\tilde{\mathbf{x}}_{\mu_p}(\mu_p,t_2) \quad \cdots \quad \mathbf{x}(\mu_p,t_{N_t}) - U_{\mu_p}\tilde{\mathbf{x}}_{\mu_p}(\mu_p,t_{N_t})\|_F \approx \bar{\varepsilon}. \tag{7}$$

3.3 Supervised Machine Learning Techniques

In order to estimate the level of reduced-order model solution error $\varepsilon_{\mu,\mu_{p_j},K_{POD}}^{HF}$ (4) and the reduced basis dimension K_{POD}, we will use regression machine learning methods to approximate the maps $\phi_{MP-LROM}^e$ and $\phi_{MP-LROM}^d$ described in (5) and (6).

Artificial Neural Networks and Gaussian Processes are used to build a probabilistic model $\phi : \mathbf{z} \mapsto \hat{y}$, where ϕ is a transformation function that learns through the input features $\mathbf{z}$ to estimate the deterministic output y [48]. As such, these probabilistic models are approximations of the mappings introduced in (3). The input features $\mathbf{z}$ can be either categorical or ordinal. The real-valued random variable $\hat{y}$ is expected to have a low variance and reduced bias. The features of $\mathbf{z}$ should be descriptive of the underlying problem at hand [8]. The accuracy and stability of estimations are assessed using the K-fold cross-validation technique. The samples are split into K subsets ("folds"), where typically $3 \le K \le 10$. The model is trained on $K-1$ sets and tested on the K-th set in a round-robin fashion [48]. Each fold induces a specific error quantified as the average of the absolute values of the differences between the predicted and the K-th set values

$$\mathrm{E}_{\mathrm{fold}} = \frac{\sum_{i=1}^N |\hat{y}^i - y^i|}{N}, \quad \mathrm{VAR}_{\mathrm{fold}} = \frac{\sum_{i=1}^N \left(\hat{y}^i - \mathrm{E}_{\mathrm{fold}}\right)^2}{N-1}, \quad \mathrm{fold} = 1, 2, \ldots, \mathrm{K}, \tag{8a}$$

where N is the number of test samples in the fold. The error is then averaged over all folds:

$$\mathrm{E} = \frac{\sum_{\mathrm{fold}=1}^K \mathrm{E}_{\mathrm{fold}}}{K}, \quad \mathrm{VAR} = \frac{\sum_{\mathrm{fold}=1}^K \left(\mathrm{E}_{\mathrm{fold}} - \mathrm{E}\right)^2}{K-1}. \tag{8b}$$

The variance of the prediction results (8a) accounts for the sensitivity of the model to the particular choice of data set. It quantifies the stability of the model in response to the new training samples. A smaller variance indicates more stable predictions, however, this sometimes translates into a larger bias of the model. Models with small variance and high bias make strong assumptions about the data and tend to underfit the truth, while models with high variance and low bias tend to overfit the truth [49] . The trade-off between bias and variance in learning algorithms is usually controlled via techniques such as regularization or bagging and boosting [8].

In what follows we briefly review the Gaussian Process and Artificial Neural Networks techniques.

3.3.1   Gaussian process kernel method

A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution [57]. A Gaussian process is fully described by its mean and covariance functions

$$\phi(\mathbf{z}) \sim \mathrm{gp}\left(m(\mathbf{z}), \mathbf{K}\right),$$

(9)

where $m(\mathbf{z}) = \mathbb{E}\left[\phi(\mathbf{z})\right]$, and $\mathbf{K}$ is the covariance matrix with entries $K_{i,j} = \mathbb{E}\left[\left(\phi(\mathbf{z}^i) - m(\mathbf{z}^i)\right)\left(\phi(\mathbf{z}^j) - m(\mathbf{z}^j)\right)\right]$ [57].

In this work we employ the commonly used squared-exponential-covariance Gaussian kernel with

$$k : \mathbb{R}^r \times \mathbb{R}^r \to \mathbb{R}, \; k(\mathbf{z}^i, \mathbf{z}^j) = \sigma_\phi^2 \exp\left(-\frac{\left\|\mathbf{z}^i - \mathbf{z}^j\right\|}{2\,\hbar^2}\right) + \sigma_n^2\,\delta_{i,j},$$

(10)

and $K_{ij} = k(\mathbf{z}^i, \mathbf{z}^j)$ [57], where $\mathbf{z}^i$ and $\mathbf{z}^j$ are the pairs of data points in training or test samples, δ is the Kronecker delta symbol and $\|\cdot\|$ is some appropriate norm. The model (10) has three hyper-parameters. The length-scale $\hbar$ governs the correlation among data points. The signal variance $\sigma_\phi^2 \in \mathbb{R}$ and the noise variance $\sigma_n^2 \in \mathbb{R}$ govern the precision of variance and noise, respectively.

Consider a set of training data points $\mathbf{Z} = [\mathbf{z}^1\,\mathbf{z}^2\,\cdots\,\mathbf{z}^n] \in \mathbb{R}^{r \times n}$ and the corresponding noisy observations $\mathbf{y} = [y^1\,y^2\,\cdots\,y^n] \in \mathbb{R}^{1 \times n}$,

$$y^i = \phi(\mathbf{z}^i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}\left(0, \sigma_n^2\right), \quad i = 1, \ldots, n.$$

(11)

Consider also the set of test points $\mathbf{Z}^* = [\mathbf{z}^{*1}\,\mathbf{z}^{*2}\,\cdots\,\mathbf{z}^{*m}] \in \mathbb{R}^{r \times m}$ and the predictions $\hat{\mathbf{y}} = [\hat{y}^1\,\hat{y}^2\,\cdots\,\hat{y}^m] \in \mathbb{R}^{1 \times m}$,

$$\hat{y}^i = \phi\left(\mathbf{z}^{*i}\right), \quad i = 1, \ldots, m.$$

(12)

For a Gaussian prior the joint distribution of training outputs $\mathbf{y}$ and test outputs $\hat{\mathbf{y}}$ is

$$\begin{bmatrix} \mathbf{y}^T \\ \hat{\mathbf{y}}^{\mathbf{T}} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{m}(\mathbf{Z})^T \\ \mathbf{m}(\mathbf{Z}^*)^T \end{bmatrix}, \begin{bmatrix} \mathbf{K} & \mathbf{K}^* \\ \mathbf{K}^{*T} & \mathbf{K}^{**} \end{bmatrix}\right),$$

(13)

where

$$\mathbf{m}(\mathbf{Z}) = [m(\mathbf{z}^1)\,m(\mathbf{z}^2)\,\cdots\,m(\mathbf{z}^n)] \in \mathbb{R}^{1 \times n}, \; \mathbf{m}(\mathbf{Z}^*) = [m(\mathbf{z}^{*1})\,m(\mathbf{z}^{*2})\,\cdots\,m(\mathbf{z}^{*m})] \in \mathbb{R}^{1 \times m},$$

$$\mathbf{K}^* = (K_{ij}^*)_{i=1,\ldots,n;\,j=1,\ldots,m} = k(\mathbf{z}^i, \mathbf{z}^{j*}) \text{ and } \mathbf{K}^{**} = (K_{ij}^{**})_{i=1,\ldots,m;\,j=1,\ldots,m} = k(\mathbf{z}^{i*}, \mathbf{z}^{j*}).$$

The predictive distribution represents the posterior after observing the data [8] and is given by

$$p\left(\hat{\mathbf{y}}|\mathbf{Z}, \mathbf{y}, \mathbf{Z}^*\right) \sim \mathcal{N}\left(\mathbf{K}^{*T}\mathbf{K}^{-1}\mathbf{y}, \; \mathbf{K}^{**} - \mathbf{K}^{*T}\mathbf{K}^{-1}\mathbf{K}^*\right),$$

(14)

where superscript T denotes the transpose operation.

The prediction of Gaussian process will depend on the choice of the mean and covariance functions, and on their hyper parameters $\hbar$, σ_ϕ^2 and σ_n^2 which can be inferred from the data

$$\boldsymbol{\theta}^* = [\hbar, \sigma_\phi^2, \sigma_n^2] = \arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}),$$

by minimizing the marginal negative log-likelihood function

$$L(\boldsymbol{\theta}) = -\log p(\mathbf{y}|\mathbf{Z}, \boldsymbol{\theta}) = \frac{1}{2}\log\det(\mathbf{K}) + \frac{1}{2}(\mathbf{y} - \mathbf{m}(\mathbf{Z}))\,\mathbf{K}^{-1}\,(\mathbf{y} - \mathbf{m}(\mathbf{Z}))^T + \frac{n}{2}\log(2\pi).$$

3.3.2 Artificial Neural Networks

The study of Artificial Neural Networks begins in the 1910s in order to imitate human brain's biological structure. Pioneering work was carried out by Rosenblatt, who proposed a three-layered network structure, the perceptron [30] . ANN detect the pattern of data by discovering the input–output relationships. Applications include the approximation of functions, regression analysis, time series prediction, pattern recognition, and speech synthesis and recognition [38, 4].

ANN consist of neurons and connections between the neurons (weights). Neurons are organized in layers, where at least three layers of neurons (an input layer, a hidden layer, and an output layer) are required for construction of a neural network.

The input layer distributes input signals $\mathbf{z} = [z_1 \; z_2 \; \cdots \; z_r]$ to the first hidden layer. For a neural network with L hidden layers and m^ℓ neurons in each hidden layer, let $\hat{\mathbf{y}}^\ell = [\hat{y}_1^\ell \; \hat{y}_2^\ell \; \cdots \; \hat{y}_{m^\ell}^\ell]$ be the vector of outputs from layer ℓ, $\mathbf{b}^\ell = [b_1^\ell \; b_2^\ell \; \cdots \; b_{m^\ell}^\ell]$ the biases at layer ℓ, and $\mathbf{w}_j^\ell = [w_{j_1}^\ell w_{j_2}^\ell \; \cdots \; w_{j_{m^l}}^l]$ the weights connecting the neuron j to the input of that layer (output of previous layer). The vectors $\hat{\mathbf{y}}^\ell$ and $\mathbf{w}_j^\ell$ share the same dimension which varies along the layers depending on the number of input features, neurons and outputs. Then the feed-forward operation is

$$x_j^{\ell+1} = \mathbf{w}_j^{\ell+1^T} \hat{\mathbf{y}}^\ell + b_j^{\ell+1}, \quad \hat{\mathbf{y}}^0 = \mathbf{z}, \quad j = 1, \ldots, m^\ell.$$
$$\hat{y}_j^{\ell+1} = \varphi\left(\mathbf{x}^{\ell+1}\right), \quad \ell = 0, 1, \ldots, L-1.$$

All products of previous layer output with current layer neuron weights will be summed and the bias value of each neuron will be added to obtain the vector $\mathbf{x}^\ell = [x_1^\ell \; x_2^\ell \; \cdots \; x_{m^\ell}^\ell]$. Then the final output of each layer will be obtained by passing the vector $\mathbf{x}^\ell$ through the transfer function φ, which is a differentiable function and can be log-sigmoid, hyperbolic tangent sigmoid, or linear transfer function.

The training process of ANN adjusts the weights and the biases in order to reproduce the desired outputs when fed the given inputs. The training process via the back propagation algorithm [62] uses a gradient descent method to modify weights and thresholds such that the error between the desired output and the output signal of the network is minimized [24]. In supervised learning the network is provided with samples from which it discovers the relations of inputs and outputs. The output of the network is compared with the desired output, and the error is back-propagated through the network and the weights will be adjusted. This process is repeated during several iterations, until the network output is close to the desired output [31].

4 Numerical experiments

We illustrate the application of the proposed MP-LROM models to predict the error and dimension of the local parametric reduced-order models for a one-dimensional Burgers model. The 1D-Burgers model proposed herein is characterized by the viscosity coefficient. To assess the performance of the MP-LROM models constructed using Gaussian Process and Artificial Neural Networks, we employ various cross-validation tests. The dimensions of the training and testing data sets are chosen empirically based on the number of samples. For Artificial Neural Networks models the number of hidden layers and neurons in each hidden layer vary for each type of problems under study. The squared-exponential-covariance kernel (10) is used for Gaussian Process models.

The approximated MP-LROM error models are compared against the ROMES and multi-fidelity correction models, whereas the MP-LROM models that predict the dimension of the reduced-order models are verified against the standard approach based on the spectrum of snapshots matrix.

4.1 One-dimensional Burgers' equation

Burgers' equation is an important partial differential equation from fluid mechanics [9]. The evolution of the velocity u of a fluid evolves according to

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \mu\frac{\partial^2 u}{\partial x^2}, \quad x \in [0, L], \quad t \in (0, t_f], \tag{15}$$

with $t_f = 1$ and $L = 1$. Here μ is the viscosity coefficient.

The model has homogeneous Dirichlet boundary conditions $u(0, t) = u(L, t) = 0$, $t \in (0, t_f]$. For the initial conditions, we used a seventh order polynomial constructed using the least-square method and the data set $\{(0,0); (0.2, 1); (0.4, 0.5); (0.6, 1); (0.8, 0.2); (0.9, 0.1); (0.95, 0.05); (1, 0)\}$. We employed the polyfit function in Matlab and the polynomial is shown in Figure 1.

The discretization uses a spatial mesh of N_s equidistant points on $[0, L]$, with $\Delta x = L/(N_s - 1)$. A uniform temporal mesh with N_t points covers the interval $[0, t_f]$, with $\Delta t = t_f/(N_t - 1)$. The discrete velocity vector is $\mathbf{u}(t_j) \approx [u(x_i, t_j)]_{i=1,2,\ldots,N_{\text{state}}} \in \mathbb{R}^{N_{\text{state}}}$, $j = 1, 2, \ldots, N_t$, where $N_{\text{state}} = N_s - 2$ (the known boundaries are removed). The semi-discrete version of the model (15) is

$$\mathbf{u}' = -\mathbf{u} \odot A_x \boldsymbol{u} + \mu A_{xx} \boldsymbol{u}, \tag{16}$$

where $\mathbf{u}'$ is the time derivative of $\mathbf{u}$, and $A_x, A_{xx} \in \mathbb{R}^{N_{\text{state}} \times N_{\text{state}}}$ are the central difference first-order and second-order space derivative operators, respectively, which take into account the boundary conditions, too. The model is implemented in Matlab and the backward Euler method is employed for time discretization. The nonlinear algebraic systems are solved using the Newton-Raphson method and the allowed number of Newton iterations per each time step is set to 50. The solution is considered to have converged when the Euclidean norm of the residual is less then 10^{-10}.

The viscosity parameter space $\mathcal{P}$ is set to the interval $[0.01, 1]$. Smaller values of μ correspond to sharper gradients in the solution, and lead to dynamics more difficult to accurately approximate using reduced-order models.
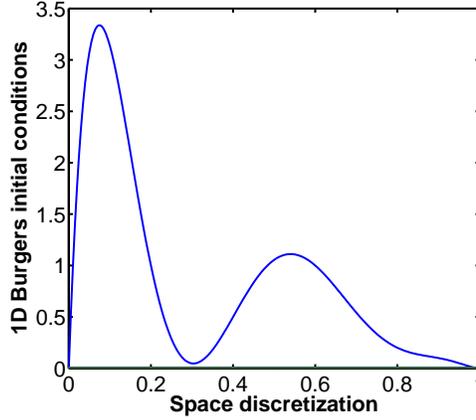

Figure 1: Seventh order polynomial used as initial conditions for 1D Burgers model.

The reduced-order models are constructed using POD method whereas the quadratic nonlinearities are computed via tensorial POD [72] for efficiency. A floating point operations analysis of tensorial POD, POD and POD/DEIM for p^{th} order polynomial nonlinearities is available in [72]. The computational efficiency of the tensorial POD 1D Burgers model can be noticed in Figure 2. Both on-line and off-line computational costs are shown. Here we selected $\mu = \mu_p = 0.7$, $N_t = 301$, POD dimension $K_{POD} = 9$, and we let the number of space points N_s to vary. For $N_s = 201$ and 701, the tensorial POD model is $5.17\times$ and $61.12\times$ times faster than the high-fidelity version. The rest of our numerical experiments uses $N_s = 201$ and $N_t = 301$.
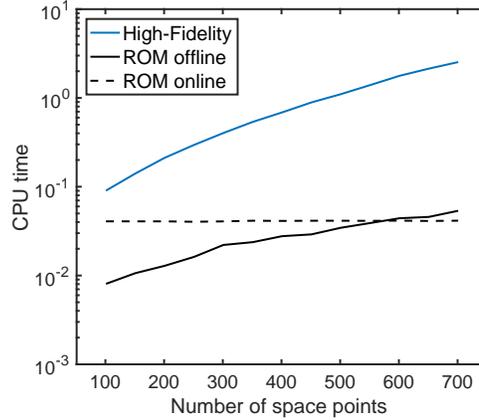

Figure 2: Computational efficiency of the tensorial POD 1D Burgers model. CPU time is given in seconds.

4.2 Multivariate prediction of local reduced-order models characteristics (MP-LROM) using regression machine learning methods

4.2.1 Error estimation of local ROM solutions

Here, we will use GP and ANN to approximate the MP-LROM error model introduced in (5). The approximated models have the following form

$$\phi_{MP-LROM}^e : \{\mu, \mu_p, K_{POD}\} \mapsto \widehat{\log \varepsilon}_{\mu,\mu_p,K_{POD}}^{HF}, \tag{17}$$

where the input features include a viscosity parameter value μ, a parameter value μ_p associated with the full model run that generated the basis U_{μ_p}, and the dimension of the reduced manifold K_{POD}. The target is the estimated logarithm of error of the reduced-order model solution at μ using the basis U_{μ_p} and the corresponding reduced operators computed using the Frobenius norm

$$\log \varepsilon_{\mu,\mu_p,K_{POD}}^{HF} = $$
$$\log \left(\left\| \mathbf{x}(\mu, t_1) - U_{\mu_p}\tilde{\mathbf{x}}_{\mu_p}(\mu, t_1) \quad \mathbf{x}(\mu, t_2) - U_{\mu_p}\tilde{\mathbf{x}}_{\mu_p}(\mu, t_2) \quad \cdots \quad \mathbf{x}(\mu, t_{N_t}) - U_{\mu_p}\tilde{\mathbf{x}}_{\mu_p}(\mu, t_{N_t}) \right\|_F \right). \tag{18}$$

7

The probabilistic models described generically in equation (17) are just approximations of the MP-LROM model (5) and have errors. For our experiments, the data set includes 10 and 100 equally distributed values of μ_p and μ over the entire parameter region; i.e., $\mu_p \in \{0.1, 0.2, \ldots, 1\}$ and $\mu \in \{0.01, \ldots, 1\}$, 12 reduced basis dimensions K_{POD} spanning the interval $\{4, 5, \ldots, 14, 15\}$ and the reduced-order model logarithm of errors $\log \varepsilon_{\mu,\mu_p,K_{POD}}^{HF}$.

The entire data set contains 12000 samples, and for each 12 samples a high-fidelity model solution is calculated. Only one high-fidelity model simulation is required for computing the reduced solutions errors for the parametric configuration μ using reduced-order models of various K_{POD} constructed based on a single high-fidelity trajectory described by parameter μ_p. As such, 1000 high-fidelity simulations were needed to construct the entire data set. High-fidelity simulations are used to accurately calculate the errors associated with the existing reduced-order models for parametric configurations μ.

Figure 3 shows isocontours of the error $\varepsilon_{\mu,\mu_p,K_{POD}}^{HF}$ and $\log \varepsilon_{\mu,\mu_p,K_{POD}}^{HF}$ of the reduced-order model solution for various viscosity parameter values μ and POD basis dimensions. The design of the reduced-order models relies on the high-fidelity trajectory for $\mu_p = 0.8$. The target values $\varepsilon_{\mu,\mu_p,K_{POD}}^{HF}$ vary over a wide range (from 300 to 10^{-6}) motivating the choice of implementing models that target $\log \varepsilon_{\mu,\mu_p,K_{POD}}^{HF}$ to decrease the variance of the predicted results.

(a) Isocontours of the errors $\varepsilon_{\mu,\mu_p,K_{POD}}^{HF}$

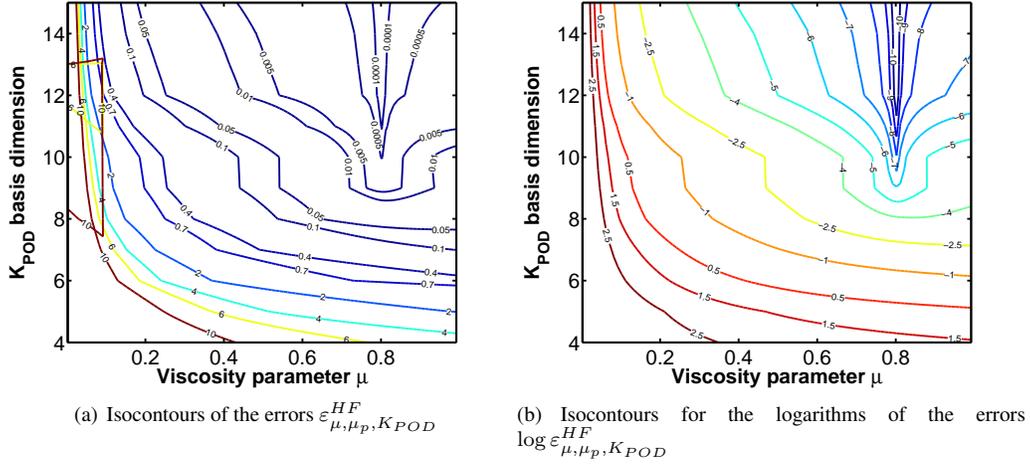(b) Isocontours for the logarithms of the errors $\log \varepsilon_{\mu,\mu_p,K_{POD}}^{HF}$

Figure 3: Isocontours of the reduced model errors for different POD basis dimensions and parameters μ. The reduced-order model uses a basis constructed from the full order simulation with parameter value $\mu_p = 0.8$.

A more detailed analysis, comparing models

$$\phi_{MP-LROM}^e : \{\mu, \mu_p, K_{POD}\} \mapsto \hat{\varepsilon}_{\mu,\mu_p,K_{POD}}^{HF} \tag{19}$$

that target no scaled data and model (17) is given in the following.

The approximated MP-LROM models for estimating the local parametric reduced-order model errors are constructed using a Gaussian Process with a squared-exponential covariance kernel (10) and a neural network with six hidden layers and hyperbolic tangent sigmoid activation function in each layer. Tables 1 and 2 show the averages and variances of errors in prediction of MP-LROM models for different sample sizes. Every subset of samples is selected randomly from a shuffled original data set. The misfit is computed using the same formulas presented in (8a) to evaluate the prediction errors. Table 1 shows the prediction errors of (19) computed via equation (8a) with $y = \varepsilon_{\mu,\mu_p,K_{POD}}^{HF}$ and $\hat{y} = \hat{\varepsilon}_{\mu,\mu_p,K_{POD}}^{HF}$; i.e., no data scaling; the predictions have a large variance and a low accuracy. Scaling the data and targeting $\log \varepsilon_{\mu,\mu_p,K_{POD}}^{HF}$ results using (17), reduce the variance of the predictions, and increase the accuracy, as shown in Table 2. The same formula (8a) with $y = \log \varepsilon_{\mu,\mu_p,K_{POD}}^{HF}$ and $\hat{y} = \widehat{\log \varepsilon}_{\mu,\mu_p,K_{POD}}^{HF}$ was applied. We notice that, for increasing sample sizes less or equal than 700 and for scaled data, the variances of GP and ANN predictions are not necessarily decreasing. This behavior changes and the variances of both regression models decrease for increasing sample sizes larger than 700 as seen in Table 2. The performance of the ANN and GP is highly dependent on the number of samples in the data set. As the number of data points grows, the accuracy increases and the variance decreases. The results show that GP outperforms ANN for small numbers of samples ≤ 1000 whereas, for larger data sets, ANN is more accurate than GP.

Sample size	GP MP-LROM		ANN MP-LROM	
	E_{fold}	VAR_{fold}	E_{fold}	VAR_{fold}
100	13.4519	5.2372	12.5189	25.0337
400	6.8003	31.0974	6.9210	26.1814
700	5.6273	14.3949	7.2325	19.9312
1000	3.7148	13.8102	5.6067	14.6488
3000	0.5468	0.0030	1.2858	1.2705
5000	6.0563	22.7761	3.8819	23.9059

Table 1: Average and variance of error in predictions of MP-LROM models (19) constructed via ANN and GP using errors $\varepsilon^{HF}_{\mu,\mu_p,K_{POD}}$ in training data for different sample sizes.

Sample size	GP MP-LROM		ANN MP-LROM	
	E_{fold}	VAR_{fold}	E_{fold}	VAR_{fold}
100	0.5319	0.0118	1.2177	0.1834
400	0.3906	0.0007	0.8988	0.2593
700	0.3322	0.0018	0.7320	0.5602
1000	0.2693	0.0002	0.5866	0.4084
3000	0.1558	0.5535×10^{-4}	0.01202	0.2744×10^{-4}
5000	0.0775	0.4085×10^{-5}	0.0075	0.3812×10^{-5}

Table 2: Average and variance of error in predictions of MP-LROM models (19) constructed via ANN and GP using logarithms of errors $\log \varepsilon^{HF}_{\mu,\mu_p,K_{POD}}$ in training data for different sample sizes.

Figures 4 and 5 show the corresponding histogram of the errors in prediction of MP-LROM models (17) and (19) using 100 and 1000 training samples for ANN and GP methods, respectively. The histograms shown in Figure 5 can assess the validity of GP assumptions (9), (11), (13). The difference between the true and estimated values should behave as samples from the distribution $\mathcal{N}(0, \sigma_n^2)$ [19]. In our case they are hardly normally distributed and this indicates that the data sets are not from Gaussian distributions.

Scaling the data and targeting $\log \varepsilon^{HF}_{\mu,\mu_p,K_{POD}}$ errors clearly improve the performance of the MP-LROM models. Consequently for the rest of the manuscript we will only use model (17).

To asses the quality of the MP-LROM models, we also implemented a five-fold cross-validation test over the entire dataset. The results computed using formula (8b) are shown in Table 3. ANN outperforms GP and estimates the errors more accurately. It also has less variance than the Gaussian Process which indicates it has more stable predictions.

	E	VAR
ANN MP-LROM	0.004004	2.16×10^{-6}
GP MP-LROM	0.092352	1.32×10^{-5}

Table 3: MP-LROM statistical results over five-fold cross-validation.

Figure 6 illustrates the average of errors in prediction of five different errors models computed using ANN and GP regression methods. The error models were constructed using a training set formed by 80% randomly selected data of the entire data set. The predictions were made using a fixed test set randomly selected from the entire data set and contains various values of μ, K_{POD} and μ_p shown in the x-axes of Figure 6. Building different GP and ANN MP-LROM error models, each trained on different part of the data set and then testing them with the same fixed test set, reduces the bias in prediction. Again, ANN outperforms GP having more accurate errors estimates.

We also compared the MP-LROM models with those obtained by implementing ROMES method [19] and MFC technique [2]. The ROMES method constructs univariate models

$$\phi_{ROMES} : \log \rho(\mu) \mapsto \log \varepsilon^{HF}_{\mu}, \tag{20}$$

where the input $\rho(\mu)$ consists of error indicators. Examples of indicators include residual norms, dual-weighted residuals and other error bounds. MFC implements input-output models

$$\phi_{MFC} : \mu \mapsto \log \varepsilon^{HF}_{\mu}, \tag{21}$$

where the input of error models is the viscosity parameter μ. Both ROMES and MFC methods use a global reduced-order model with a fixed dimension in contrast to our method that employs local reduced-order models with various dimensions. ROMES and MFC models are univariate whereas the MP-LROM models are multivariate.
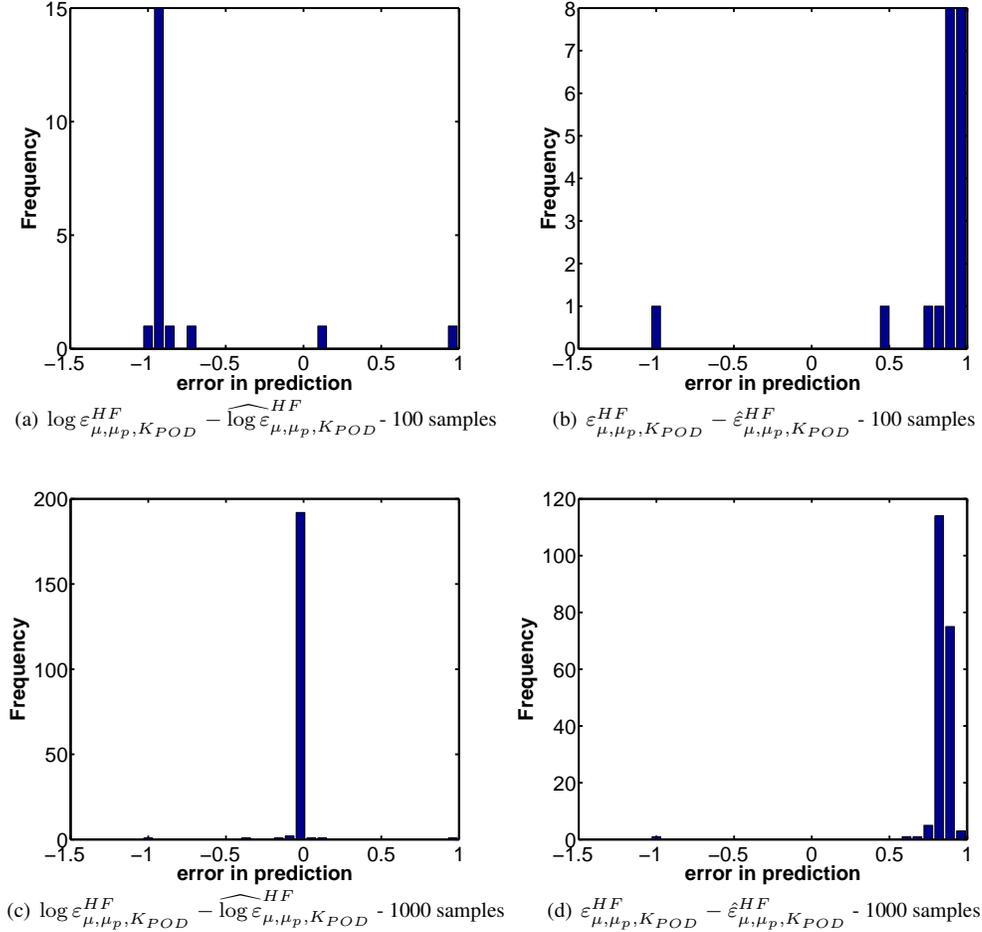
(a) $\log \varepsilon_{\mu,\mu_p,K_{POD}}^{HF} - \widehat{\log \varepsilon}_{\mu,\mu_p,K_{POD}}^{HF}$ - 100 samples

(b) $\varepsilon_{\mu,\mu_p,K_{POD}}^{HF} - \hat{\varepsilon}_{\mu,\mu_p,K_{POD}}^{HF}$ - 100 samples

(c) $\log \varepsilon_{\mu,\mu_p,K_{POD}}^{HF} - \widehat{\log \varepsilon}_{\mu,\mu_p,K_{POD}}^{HF}$ - 1000 samples

(d) $\varepsilon_{\mu,\mu_p,K_{POD}}^{HF} - \hat{\varepsilon}_{\mu,\mu_p,K_{POD}}^{HF}$ - 1000 samples

Figure 4: Histogram of errors in prediction using ANN MP-LROM.

To accommodate our data set to the requirements of the ROMES and MFC methods, we separated the data set into multiple subsets. Each of these subsets has 100 samples corresponding to a single μ_p and K_{POD} and 100 values of parameter $\mu \in \{0.01, 0.02 \ldots, 1\}$, so 100 high-fidelity simulations are required. For each subset we constructed ANN and GP models to approximate the input-output models defined in (20) and (21) using the same training set. In the case of ROMES method we employed the logarithms of residuals norms as inputs. We first computed the corresponding reduced-order solution and then the associated logarithm of residual norm by using the projected reduced order solution into the high-fidelity model for parameter μ. The output of both ROMES and MFC models approximates the logarithm of the Frobenius norm of the reduced-order-model errors.

Figures 7-10 shows the isocontours of the E_{fold} and VAR_{fold} computed using (8a) for different K_{POD} and μ_p using ROMES, MFC, and MP-LROM models constructed using GP and ANN methods. In total there are 12×10 configurations corresponding to different K_{POD} and μ_p and as many ROMES and MFC models. The MP-LROM models are global in nature and the training set is the whole original data set. The testing set is the same for all the compared models and differs from the training sets. We can see that MP-LROM models are more accurate than those obtained via ROMES and MFC models. Including more samples associated with various POD basis sizes and μ_p is benefic. We also trained and tested all the models using five-fold cross-validation. The average error and variance of all 120 E_{fold}s and VAR_{fold}s are compared against those obtained using MP-LROM error models and are summarized in tables 4 and 5. This shows that the MFC models outperform the ROMES ones, for our experiment, and the MP-LROM models are the most accurate. The MP-LROM models perform better since they employ more features and samples than the other models which help the error models tune the parameters better. We also notice the efficiency of the MFC models from accuracy point of view considering that they use very few samples. In the case of large parametric domains the MP-LROM error models may require a very large data set with a lot of features. By using only subsets of the whole data set near the vicinity of the parameters of interest and applying the active subset method [14] can help prevent the potential curse of dimensionality problem that MP-LROM might suffer.
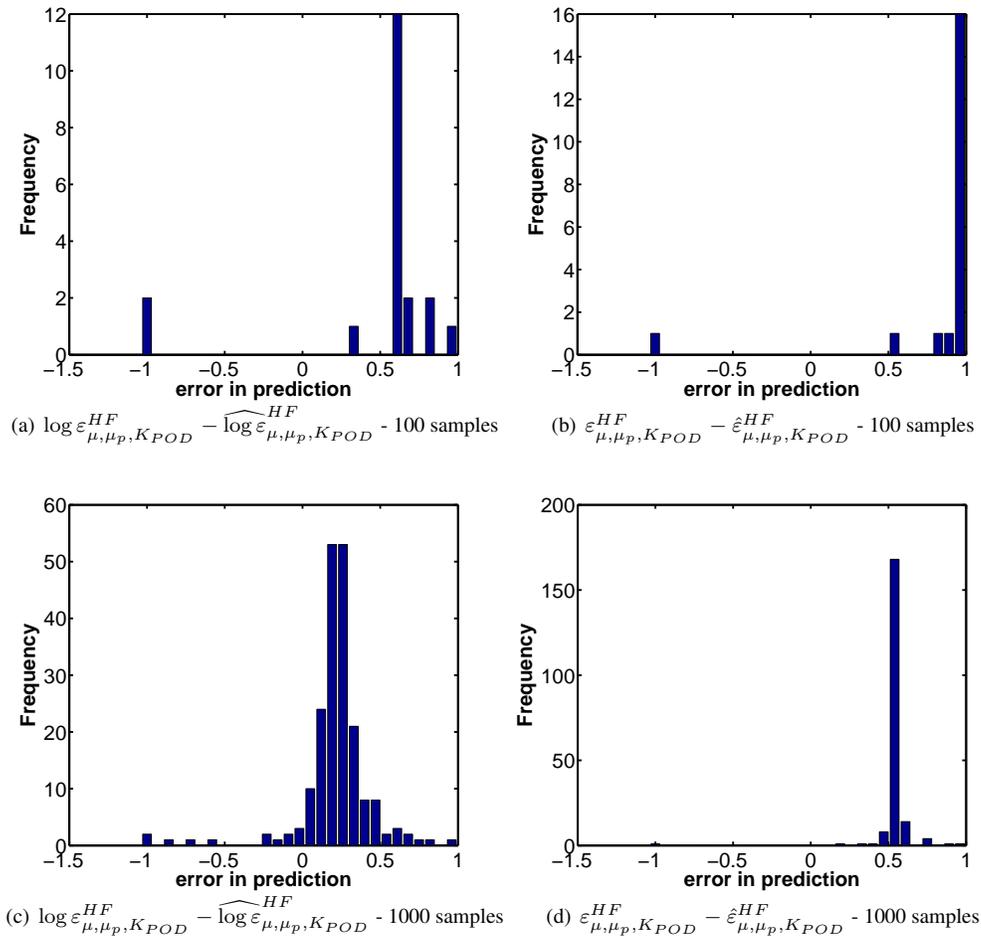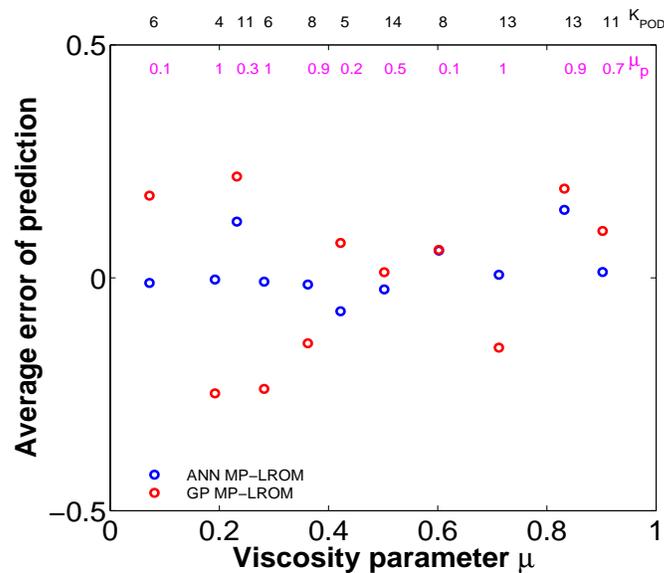
(a) $\log \varepsilon_{\mu,\mu_p,K_{POD}}^{HF} - \widehat{\log \varepsilon}_{\mu,\mu_p,K_{POD}}^{HF}$ - 100 samples

(b) $\varepsilon_{\mu,\mu_p,K_{POD}}^{HF} - \hat{\varepsilon}_{\mu,\mu_p,K_{POD}}^{HF}$ - 100 samples

(c) $\log \varepsilon_{\mu,\mu_p,K_{POD}}^{HF} - \widehat{\log \varepsilon}_{\mu,\mu_p,K_{POD}}^{HF}$ - 1000 samples

(d) $\varepsilon_{\mu,\mu_p,K_{POD}}^{HF} - \hat{\varepsilon}_{\mu,\mu_p,K_{POD}}^{HF}$ - 1000 samples

Figure 5: Histogram of errors in prediction using GP MP-LROM.

Figure 6: The average of errors in predictions using five different trained models. The top labels shows the corresponding μ_p and K_{POD} for each parameter μ.

	ROMES	MFC	MP-LROM
ANN	0.3844	0.0605	8.8468×10^{-4}
GP	0.2289	0.0865	0.0362

Table 4: Average error of all 120 E_{fold}s for three methods.

(a) MFC
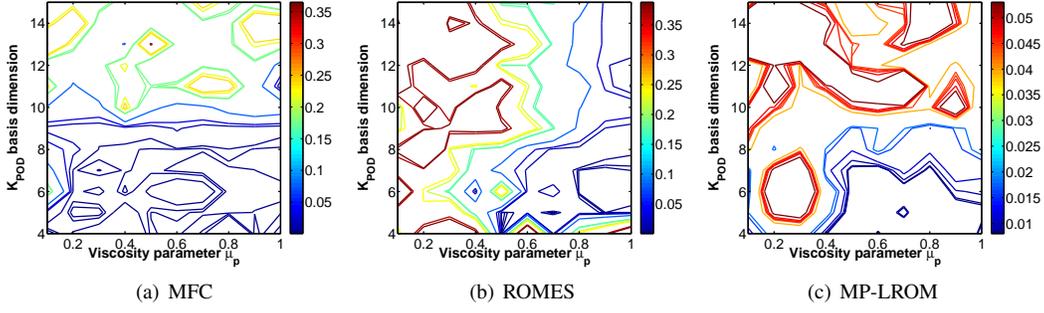
(b) ROMES

(c) MP-LROM

Figure 7: Isocontours for the E_{fold} using GP method.

(a) MFC
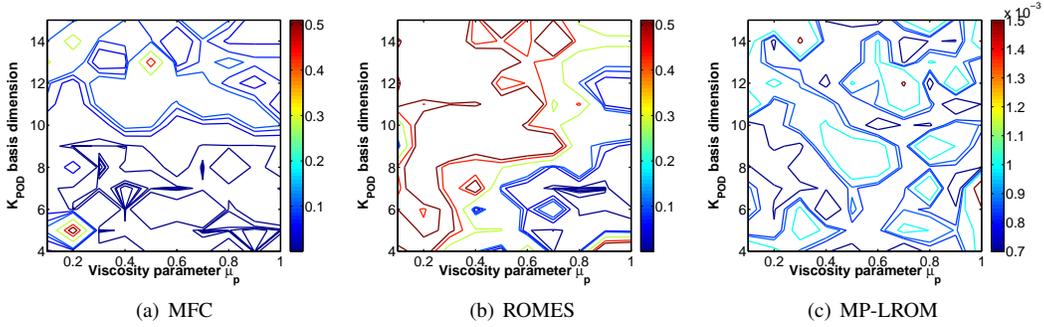
(b) ROMES

(c) MP-LROM

Figure 8: Isocontours for the E_{fold} using ANN method.

	ROMES	MFC	MP-LROM
ANN	0.0541	0.0213	4.9808×10^{-7}
GP	0.0051	0.0049	5.4818×10^{-4}

Table 5: Average variance of all 120 VAR$_{fold}$s for three methods

Finally we compare and show the average of the errors in prediction of five different errors models designed using ROMES, MFC, and MP-LROM methods for one of the subsets corresponding to $K_{POD} = 10$ and $\mu_p = 1$. The testing set is randomly selected from the samples and is not included in the training sets. The training set for both ROMES and MFC models are the same. In order to prevent the bias in prediction, each time the error models are trained on randomly selected 80% of the training sets and tested with the fixed test set. We repeated this five times and the average of error in prediction is obtained. Figure 11 shows the average of error in prediction for all models implemented using GP and ANN methods.

Since we are including more features in our mappings, we achieve more accurate predictions compared to other existing methods such as ROMES and MFC. However, there is always a trade-off between the computational complexity and the accuracy. For more accurate results, one can generate a bigger dataset with more samples taken from the parameter domain of the underlying model. This elevates the computational complexity since the dataset requires more high-fidelity model solutions and the probabilistic mappings are more costly to construct in the training phase. Techniques such as principal component analysis and active subspace can alleviate the curse of dimensionality for big datasets by selecting the most effective features and ignoring the less-effective ones.

4.2.2   Selecting the dimension of reduced-order model

Here we construct MP-LROM models to predict the reduced basis dimension that account for a-priori specified accuracy levels in the reduced-order model solution. The models are constructed using GP and ANN methods and have the following form

$$\phi_{MP-LROM}^d : \{\mu_p, \log \varepsilon_{\mu_p,\mu_p,K_{POD}}^{HF}\} \mapsto \widehat{K_{POD}}. \tag{22}$$

The input features of this model consist of the viscosity parameter $\mu_p \in [0.01, 1]$ and the log of the Frobenius norm of the error between the high-fidelity and reduced-order models (18). The searched output $\widehat{K_{POD}}$ is the estimation of the dimension of the reduced manifold K_{POD}. The data set contains equally distributed values of μ_p over the entire parametric domain $\mu_p \in \{0.01, 0.0113, 0.0126, \ldots, 0.9956\}$, reduced basis dimensions K_{POD} spanning the set $\{4, 5, \ldots, 14, 15\}$ and the logarithm of the reduced-order model error $\log \varepsilon_{\mu_p,\mu_p,K_{POD}}^{HF}$. We use GP and ANN methods to construct two MP-LROM models to predict the dimension of local reduced-order models given a prescribed accuracy level.

12
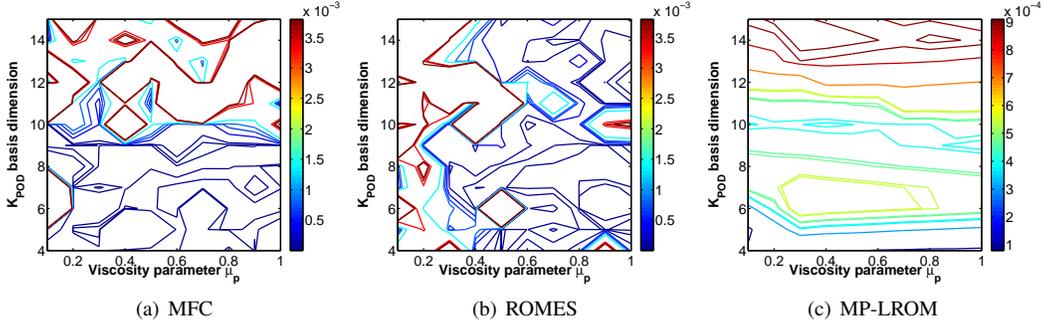
Figure 9: Isocontours for the VAR$_{\text{fold}}$ using GP method.
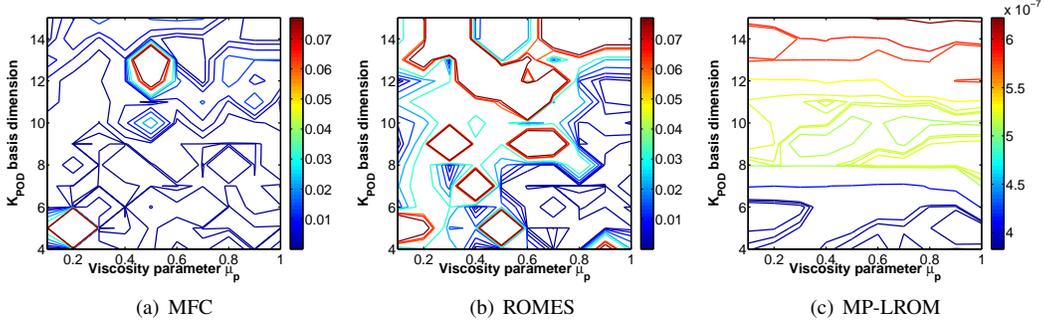
Figure 10: Isocontours for the VAR$_{\text{fold}}$ using ANN method.

During the training phase, the MP-LROM models will learn the dimensions of reduced-order basis K_{POD} associated with the parameter μ_p and the corresponding error $\log \varepsilon_{\mu_p,\mu_p,K_{POD}}^{HF}$. Later they will be able to estimate the proper dimension of reduced basis by providing it the specific viscosity parameter μ_p and the desired precision $\log \bar{\varepsilon}$. The computational cost is low once the models are constructed. The output indicates the dimension of the reduced manifold for which the ROM solution satisfies the corresponding error threshold. Thus we do not need to compute the entire spectrum of the snapshots matrix in advance which for large spatial discretization meshes translates into important computational costs reduction. Figure 12 illustrates the contours of the log of reduced-order model errors over all the values of the viscosity parameter $\mu_p \in \{0.01, 0.0113, 0.0126 \ldots 1\}$ and various POD dimensions $K_{POD} = \{4, 5, \ldots, 14, 15\}$.

A neural network with 5 hidden layers and hyperbolic tangent sigmoid activation function in each layer is used while for the Gaussian Process we have used the squared-exponential-covariance kernel (10). For both MP-LROM models, the results were rounded such as to generate natural numbers. Table 6 shows the average and variance of error in GP and ANN predictions using different sample sizes. ANN outperforms GP and as the number of data points grows, the accuracy increases and the variance decreases. The results are obtained using a conventional validation with 80% of the sample size dedicated for training data and the other 20% for the test data. The employed formula is described in equation (8a).

	MP-LROM GP		MP-LROM ANN	
sample size	E$_{\text{fold}}$	VAR$_{\text{fold}}$	E$_{\text{fold}}$	VAR$_{\text{fold}}$
100	0.2801	0.0901	0.1580	0.02204
1000	0.1489	0.0408	0.0121	0.0015
3000	0.1013	0.0194	0.0273	0.0009
5000	0.0884	0.0174	0.0080	0.0002

Table 6: Average and variance of errors in prediction of reduced basis dimension using MP-LROM models for different sample sizes

Figures 13 and 14 show the prediction errors using 100 and 1000 training samples for both MP-LROM models constructed via ANN and GP models. The histograms shown in Figure 14, as stated before, can assess the validity of GP assumptions. Once the number of samples is increased, the data set distribution shape is closer to the Gaussian profile $\mathcal{N}(0, \sigma_n^2)$ than in the case of the data set distribution shown in Figure 5 used for generation of MP-LROM models for the prediction of local reduced-order model errors.

To assess the accuracy of the MP-LROM models, the data set is randomly partitioned into five equal size sub-samples, and five-fold cross-validation test is implemented. The five results from the folds are averaged and they are presented in Table 7. The ANN model correctly estimated the dimension of the reduced manifold in 87% cases. GP correctly estimates the POD dimension 53% of the times. The variance results shows that the GP model has more stable predictions indicating a higher bias in the data.
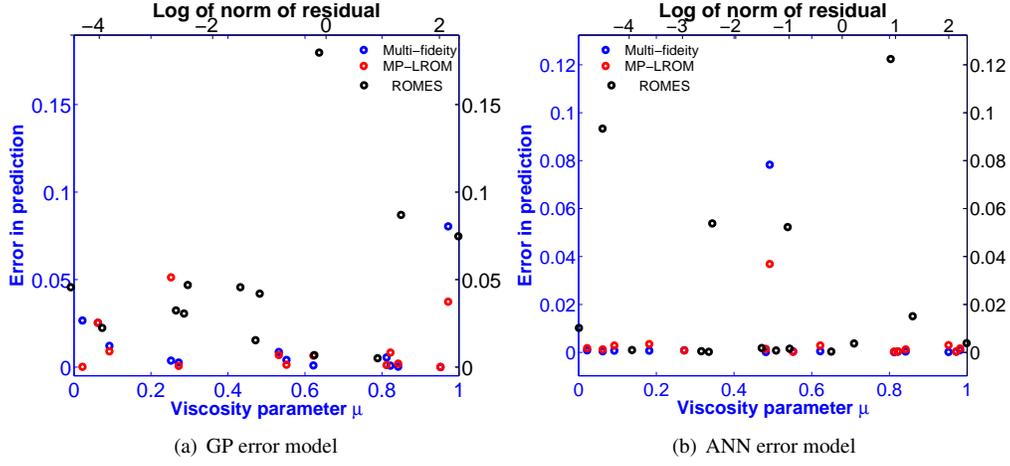
(a) GP error model

(b) ANN error model

Figure 11: The error in predictions of all methods using $K_{POD} = 10$ and $\mu_p = 1$. For GP error models the overall average of errors in predictions is $0.0131, 0.0487, 0.0095$ for MFC, ROMES and MP-LROM respectively. For ANN error models the overall average of errors in predictions is $0.0056, 0.0240, 0.0029$ for MFC, ROMES and MP-LROM respectively. The top x-axis shows the corresponding logarithms of the residuals norms used as inputs for the ROMES method.
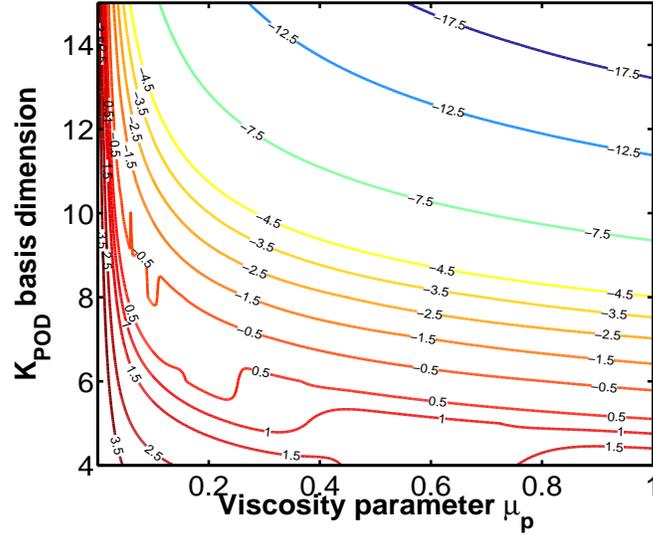

Figure 12: Isocontours of the reduced model errors for different POD basis dimensions and viscosity parameters μ_p.

Dimension discrepancies	zero	one	two	three	four	> four	VAR
ANN MP-LROM	87%	11%	2%	0	0	0	2.779×10^{-3}
GP MP-LROM	53%	23%	15%	5%	3%	1%	4.575×10^{-4}

Table 7: POD basis dimension discrepancies between the MP-LROM predictions and true values over five-fold cross-validation. The errors variance is also computed.

In Figure 15, we compare the output of the MP-LROM models against the singular values based estimation on a set of randomly selected test data. The estimation derived from the singular values is the standard method for selecting the reduced manifold dimension when a prescribed level of accuracy of the reduced solution is desired. Here the desired accuracy $\bar{\varepsilon}$ is set to 10^{-3}. The mismatches between the predicted and true dimensions are depicted in Figure 15. The predicted values are the averages over five different MP-LROM models constructed using ANN and GP methods. The models were trained on random 80% split of data set and tested on the fixed selected 20% test data. We notice that the snapshots matrix spectrum underestimates the true dimension of the manifold as expected since the 'in-plane' errors are not accounted. The ANN predictions were extremely accurate for most of the samples while the GP usually overestimated the reduced manifold dimensions.
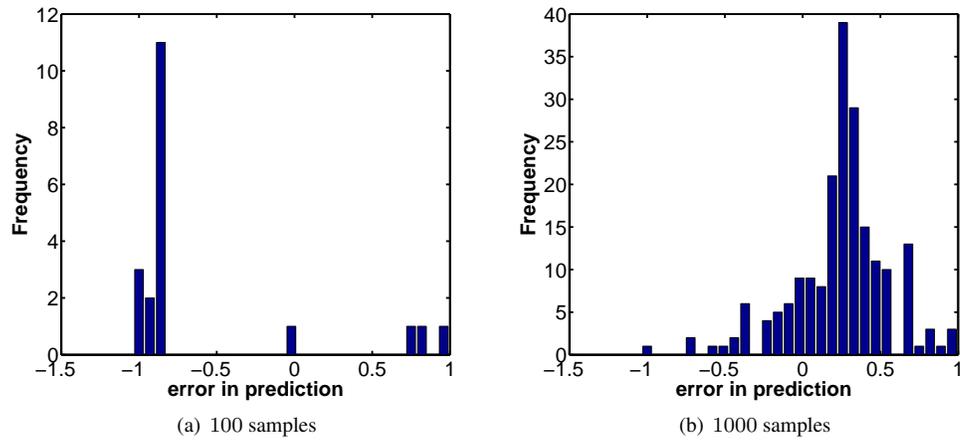
(a) 100 samples   (b) 1000 samples

Figure 13: Histogram of errors in prediction of the reduced basis dimension using ANN MP-LROM for different sample sizes

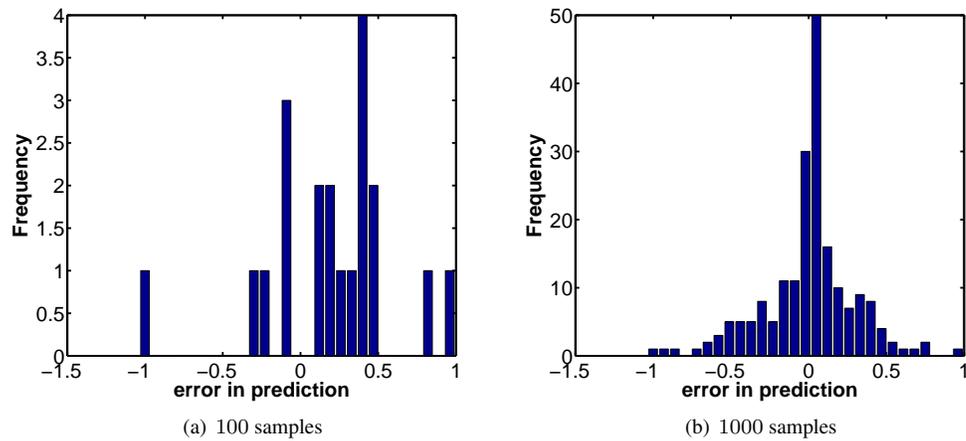(a) 100 samples   (b) 1000 samples

Figure 14: Histogram of errors in prediction of the reduced basis dimension using GP ML-ROM for different sample sizes
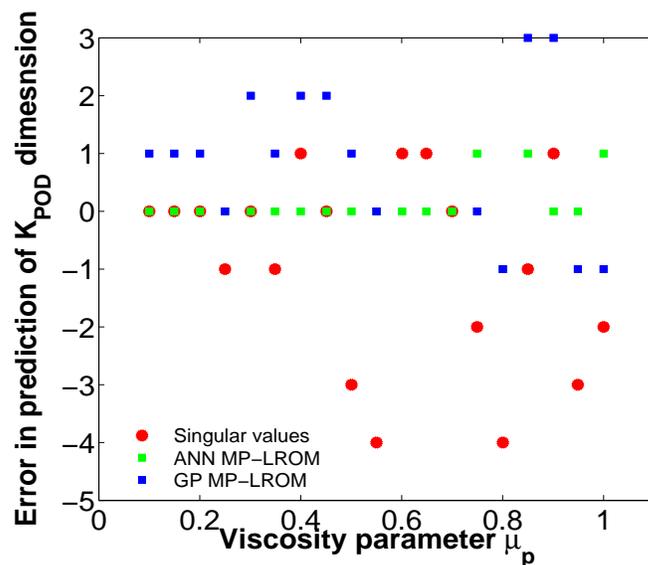

Figure 15: Average error of the POD dimension prediction on a randomly selected test data with desired accuracy of $\bar{\varepsilon} = 10^{-3}$. The average of the absolute values of the error in prediction are 1.31, 0.21 and 1.38 for singular value based method and MP-LROM models constructed using ANN and GP.

15

5 Conclusions

In this study, we introduced new multivariate input-output models (MP-LROM) to predict the errors and dimensions of local parametric reduced-order models. Approximation of these mappings were built using Gaussian Process and Artificial Neural Networks. Initially, we compared our MP-LROM error models against those constructed with multi-fidelity correction technique (MFC) and reduced order model error surrogates method (ROMES). Since global bases are used by MFC and ROMES methods, we implemented corresponding local error models using only small subsets of the data utilized to generate our MP-LROM models. In contrast, the MP-LROM models are global and rely on a global database. Moreover, our MP-LROM models differ from the ROMES [19] and MFC models [2], having more additional features such as reduced subspace dimension and are specially projected for accurate predictions of local parametric reduced-order models errors. As such, the MP-LROM models require significantly more and different data than MFC models. The numerical experiments revealed that our MP-LROM models are more accurate than the models constructed with MFC and ROMES methods for estimating the errors of local parametric reduced-order 1D-Burgers models with a single parameter. In the case of large parametric domains, the MP-LROM error models could be affected by the curse of dimensionality due to the large number of input features. In the future we plan to use only subsets of the global data set near the vicinity of the parameters of interest and combine our technique with the active subspace method [14] to prevent the potential curse of dimensionality that the MP-LROM models might suffer.

Next we addressed the problem of selecting the dimension of a local reduced-order model when its solution must satisfy a desired level of accuracy. The approximated MP-LROM models based on Artificial Neural Networks better estimated the ROM basis dimension in comparison with the results obtained by truncating the spectrum of the snapshots matrix.

In the future we seek to decrease the computational complexity of the MP-LROM error models. Currently the training data required by the machine learning regression MP-LROM models rely on many high-fidelity simulations. By employing error bounds, residual norms [19] and a-posteriori error estimation results [28, 50], this dependency could be much decreased. On-going work focuses on applications of MP-LROM error model. We are currently developing several algorithms and techniques that employ MP-LROM error model as a key component to generate decomposition maps of the parametric space associated with accurate local reduced-order models.

In addition, we plan to construct machine learning MP-LROM models to estimate the errors in quantities of interest computed with reduced-order models. The predictions of such error models can then be used to speed up the current trust-region reduced-order framework [3, 7] by eliminating the need of high-fidelity simulations for the quality evaluation of the updated controls.

Acknowledgements

References

[1] A.K. Alekseev, I.M. Navon, and M.E. Zelentsov. The estimation of functional uncertainty using polynomial chaos and adjoint equations. International Journal for Numerical Methods in Fluids, 67(3):328–341, 2011.

[2] Natalia M Alexandrov, Robert Michael Lewis, Clyde R Gumbert, Lawrence L Green, and Perry A Newman. Approximation and model management in aerodynamic optimization with variable-fidelity models. Journal of Aircraft, 38(6):1093–1101, 2001.

[3] E. Arian, M. Fahl, Sachs, and E.W. Trust-region Proper Orthogonal Decomposition for Flow Control, ICASE. Technical Report 2000-25, 2000.

[4] Ramin Ayanzadeh, Azam S Zavar Mousavi, and Saeid Setayeshi. Fossil fuel consumption prediction using emotional learning in amygdala. In Electrical Engineering (ICEE), 2011 19th Iranian Conference on, pages 1–6. IEEE, 2011.

[5] M. Barrault, Y. Maday, N.C. Nguyen, and A.T. Patera. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. Comptes Rendus Mathematique, 339(9):667–672, 2004.

[6] Peter Benner, Serkan Gugercin, and Karen Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. SIAM Review, 57(4):483–531, 2015.

[7] M. Bergmann and L. Cordier. Optimal control of the cylinder wake in the laminar regime by trust-region methods and POD reduced-order models. Journal of Computational Physics, 227(16):7813–7840, 2008.

[8] C.M. Bishop. Pattern Recognition and Machine Learning. Information Science and Statistics. Springer, 2006.

[9] Johannes Martinus Burgers. A mathematical model illustrating the theory of turbulence. Advances in Applied Mechanics, 1:171–199, 1948.

[10] Dan Cacuci. Sensitivity and Uncertainty Analysis: Theory. Chapman and Hall/CRC Press, Boca Raton, FL, 2003.

[11] Dan G. Cacuci. Second-order Adjoint Sensitivity Analysis Methodology (2nd-ASAM) for Computing Exactly and Efficiently First-and Second-order Sensitivities in Large-scale Linear Systems: I. Computational methodology. Journal of Computational Physics, 284(0):687 – 699, 2015.

[12] Dan G Cacuci. Second-order Adjoint Sensitivity Analysis Methodology (2nd-ASAM) for Computing Exactly and Efficiently First-and Second-order Sensitivities in Large-scale Linear Systems: II. Illustrative application to a paradigm particle diffusion problem. Journal of Computational Physics, 284:700–717, 2015.

[13] Dan G Cacuci, Mihaela Ionescu-Bujor, and Ionel Michael Navon. Sensitivity and uncertainty analysis, volume II: Applications to large-scale systems, volume 2. Chapman and Hall/CRC Press, Boca Raton, FL, 2005.

[14] Paul G Constantine, Eric Dow, and Qiqi Wang. Active subspace methods in theory and practice: applications to kriging surfaces. SIAM Journal on Scientific Computing, 36(4):A1500–A1524, 2014.

[15] P. Courtier, J.N. Thepaut, and A. Hollingsworth. A strategy for operational implementation of 4D-VAR, using an incremental approach. Quarterly Journal of the Royal Meteorological Society, 120:1367–1387, 1994.

[16] Luca Daniel, Ong Chin Siong, Low Sok Chay, Kwok Hong Lee, and Jacob White. A multiparameter moment-matching model-reduction approach for generating geometrically parameterized interconnect performance models. Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, 23(5):678–693, 2004.

[17] M Dihlmann, M Drohmann, and B Haasdonk. Model reduction of parametrized evolution problems using the reduced basis method with adaptive time-partitioning. Proc. of ADMOS, 2011:, 2011.

[18] Markus A Dihlmann and Bernard Haasdonk. Certified PDE-constrained parameter optimization using reduced basis surrogate models for evolution problems. Computational Optimization and Applications, 60(3):753–787, 2015.

[19] Martin Drohmann and Kevin Carlberg. The ROMES method for statistical modeling of reduced-order-model error. SIAM/ASA Journal on Uncertainty Quantification, 3(1):116–145, 2015.

[20] Jens L Eftang and Benjamin Stamm. Parameter multi-domain 'hp' empirical interpolation. International Journal for Numerical Methods in Engineering, 90(4):412–428, 2012.

[21] MS Eldred and John Burkardt. Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification. AIAA paper, 976(2009):1–20, 2009.

[22] MS Eldred, AA Giunta, SS Collis, NA Alexandrov, and RM Lewis. Second-order corrections for surrogate-based optimization with model hierarchies. In Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, NY, Aug, pages 2013–2014, 2004.

[23] Marco Fahl and Ekkehard W Sachs. Reduced order modelling approaches to PDE-constrained optimization based on proper orthogonal decomposition. In Large-scale PDE-constrained optimization, pages 268–280. Springer, 2003.

[24] Ken-Ichi Funahashi. On the approximate realization of continuous mappings by neural networks. Neural networks, 2(3):183–192, 1989.

[25] Shawn E Gano, John E Renaud, and Brian Sanders. Hybrid variable fidelity optimization by using a kriging-based scaling function. AIAA Journal, 43(11):2422–2433, 2005.

[26] M.A. Grepl and A.T. Patera. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. ESAIM: Mathematical Modelling and Numerical Analysis, 39(01):157–181, 2005.

[27] Mircea Grigoriu. Stochastic systems: Uncertainty quantification and propagation. Springer Science & Business Media, London, 2012.

[28] Martin Gubisch, Ira Neitzel, and Stefan Volkwein. A-posteriori error estimation of discrete POD models for PDE-constrained optimal control. Technical report, Konstanzer Schriften in Mathematik, 03 2016.

[29] Bernard Haasdonk and Mario Ohlberger. Reduced Basis Method for Finite Volume Approximations of Parametrized Linear Evolution Equations. ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique, 42(2):277–302, 2008.

[30] M. Hagan, H. Demuth, M. Beale, and O. De Jesus. Neural Network Design (2nd Edition). Martin Hagan, 2014.

[31] S.S. Haykin. Neural Networks and Learning Machines. Number v. 10 in Neural networks and learning machines. Prentice Hall, 2009.

[32] Jon C Helton and Freddie Joe Davis. Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. Reliability Engineering & System Safety, 81(1):23–69, 2003.

[33] M. Hinze and S. Volkwein. Error estimates for abstract linear-quadratic optimal control problems using proper orthogonal decomposition. Computational Optimization and Applications, 39:319–345, 2008.

[34] Michael Hinze and Stefan Volkwein. Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: Error estimates and suboptimal control. In Dimension Reduction of Large-Scale Systems, pages 261–306. Springer, 2005.

[35] Chris Homescu, Linda R Petzold, and Radu Serban. Error estimation for reduced-order models of dynamical systems. SIAM Journal on Numerical Analysis, 43(4):1693–1714, 2005.

[36] H. Hotelling. Analysis of a complex of statistical variables with principal components. Journal of Educational Psychology, 24:417–441, 1933.

[37] Deng Huang, TT Allen, WI Notz, and RA Miller. Sequential kriging optimization using multiple-fidelity evaluations. Structural and Multidisciplinary Optimization, 32(5):369–382, 2006.

[38] J.S.R. Jang, C.T. Sun, and E. Mizutani. Neuro-fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. MATLAB curriculum series. Prentice Hall, 1997.

[39] K. Karhunen. Zur spektraltheorie stochastischer prozess. Annales Academiae Scientarum Fennicae, 37:, 1946.

[40] K. Kunisch and S. Volkwein. Galerkin Proper Orthogonal Decomposition Methods for a General Equation in Fluid Dynamics. SIAM Journal on Numerical Analysis, 40(2):492–515, 2002.

[41] Karl Kunisch and Stefan Volkwein. Galerkin proper orthogonal decomposition methods for parabolic problems. Numerische Mathematik, 90(1):117–148, 2001.

[42] OP Le Maıtre and OM Knio. Spectral methods for uncertainty quantification. Scientific Computation. Springer, New York, NY, 2010.

[43] Christiane Lemieux. Monte Carlo and quasi-Monte Carlo sampling. Springer Science & Business Media, New York, NY, 2009.

[44] M Lilley. Gaussian processes as neural networks. PhD thesis, Honours thesis, Victoria University of Wellington, 2004. Available from http://www. mcs. vuw. ac. nz/people/Marcus-Frean, 2004.

[45] M.M. Loève. Probability Theory. Van Nostrand, Princeton, NJ, 1955.

[46] E.N. Lorenz. Empirical Orthogonal Functions and Statistical Weather Prediction. Technical report, Massachusetts Institute of Technology, Dept. of Meteorology, 1956.

[47] John Leask Lumley. The structure of inhomogeneous turbulent flows. Atmospheric turbulence and radio wave propagation, pages 166–178, 1967.

[48] Kevin P Murphy. Machine learning: a probabilistic perspective. MIT press, 2012.

[49] Andrew Ng. Stanford CS229 , Machine Learning. Technical report, Stanford, 2008.

[50] Ngoc-Cuong Nguyen, Gianluigi Rozza, and Anthony T Patera. Reduced basis approximation and a posteriori error estimation for the time-dependent viscous Burgers' equation. Calcolo, 46(3):157–185, 2009.

[51] B.R. Noack, M. Schlegel, M. Morzynski, and G. Tadmor. System reduction strategy for Galerkin models of fluid flows. International Journal for Numerical Methods in Fluids, 63(2):231–248, 2010.

[52] Benjamin Peherstorfer, Daniel Butnaru, Karen Willcox, and Hans-Joachim Bungartz. Localized discrete empirical interpolation method. SIAM Journal on Scientific Computing, 36(1):A168–A192, 2014.

[53] TA Porsching. Estimation of the error in the reduced basis method solution of nonlinear equations. Mathematics of Computation, 45(172):487–496, 1985.

[54] Christophe Prud'homme, Dimitrios V Rovas, Karen Veroy, Luc Machiels, Yvon Maday, Anthony T Patera, and Gabriel Turinici. Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods. Journal of Fluids Engineering, 124(1):70–80, 2002.

[55] Alfio Quarteroni, Gianluigi Rozza, and Andrea Manzoni. Certified reduced basis approximation for parametrized partial differential equations and applications. Journal of Mathematics in Industry, 1(1):1–49, 2011.

[56] M.L. Rapún and J.M. Vega. Reduced order models based on local POD plus Galerkin projection. Journal of Computational Physics, 229(8):3046–3063, 2010.

[57] C.E. Rasmussen and C.K.I. Williams. Gaussian Processes for Machine Learning. Adaptive computation and machine learning series. University Press Group Limited, 2006.

[58] M. Rathinam and L. Petzold. A new look at proper orthogonal decomposition. SIAM J. Numer. Anal., 41(5):1893–1925, 2003.

[59] Clarence W Rowley, Tim Colonius, and Richard M Murray. Model reduction for compressible flows using POD and Galerkin projection. Physica D: Nonlinear Phenomena, 189(1):115–129, 2004.

[60] C.W. Rowley. Model Reduction for Fluids, using Balanced Proper Orthogonal Decomposition. International Journal of Bifurcation and Chaos (IJBC), 15(3):997–1013, 2005.

[61] Gianluigi Rozza, Dinh Bao Phuong Huynh, and Anthony T Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. Archives of Computational Methods in Engineering, 15(3):229–275, 2008.

[62] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, DTIC Document, 1985.

[63] Pierre Sagaut. Large eddy simulation for incompressible flows: an introduction. Springer Science & Business Media, 2006.

[64] O. San and T. Iliescu. Proper orthogonal decomposition closure models for fluid flows: Burgers equation. Technical Report arXiv:1308.3276, Cornell University, August 2013.

[65] Ryan Schmit and Mark Glauser. Improvements in low dimensional tools for flow-structure interaction problems: using global POD. In APS Division of Fluid Dynamics Meeting Abstracts, volume 1, 2003.

[66] Alexander Shapiro. Monte Carlo sampling methods. Handbooks in Operations Research and Management Science, 10:353–425, 2003.

[67] L. Sirovich. Turbulence and the dynamics of coherent structures. I. Coherent structures. Quarterly of Applied Mathematics, 45(3):561–571, 1987.

[68] L. Sirovich. Turbulence and the dynamics of coherent structures. II. Symmetries and transformations. Quarterly of Applied Mathematics, 45(3):573–582, 1987.

[69] L. Sirovich. Turbulence and the dynamics of coherent structures. III. Dynamics and scaling. Quarterly of Applied Mathematics, 45(3):583–590, 1987.

[70] Marek Słoński. Bayesian neural networks and Gaussian processes in identification of concrete properties. Computer Assisted Mechanics and Engineering Sciences, 18(4):291–302, 2011.

[71] Ralph C Smith. Uncertainty Quantification: Theory, Implementation, and Applications, volume 12. SIAM, Philadelphia, PA, 2014.

[72] Răzvan Ştefănescu, Adrian Sandu, and Ionel M Navon. Comparison of POD reduced order strategies for the nonlinear 2D shallow water equations. International Journal for Numerical Methods in Fluids, 76(8):497–521, 2014.

[73] JA Taylor and MN Glauser. Towards practical flow sensing and control via POD and LSE based low-dimensional tools. Journal of Fluids Engineering, 126(3):337–345, 2004.

[74] Yannick Trémolet. Incremental 4D-Var convergence study. Tellus A, 59(5):706–718, 2007.

[75] Stefan Volkwein. Proper orthogonal decomposition: applications in optimization and control. CEA-EDFINRIA Numerical Analysis Summer School, page , 2007.

[76] DS Weile, E Michielssen, Eric Grimme, and K Gallivan. A Method for Generating Rational Interpolant Reduced Order Models of Two-Parameter Linear Systems. Applied Mathematics Letters, 12(5):93–102, 1999.

[77] David Wells, Zhu Wang, Xuping Xie, and Traian Iliescu. Regularized reduced order models. Technical Report arXiv:1506.07555, Cornell University, 2015.

[78] David C Wilcox. Turbulence modeling for CFD, volume 2. DCW industries La Canada, CA, 1998.

[79] K. Willcox and J. Peraire. Balanced model reduction via the Proper Orthogonal Decomposition. AIAA Journal, pages 2323–2330, 2002.