Physics-Informed Proper Orthogonal Decomposition for Data Reconstruction

Kemas Zakaria, Pramudita Satria Palar, Lavi Rizki Zuhal, Joseph Morlier

AIAA SciTech Forum
23-27 January 2023, National Harbor, MD & Online
AIAA SCITECH 2023 Forum

10.2514/6.2023-0538

Physics-Informed Proper Orthogonal Decomposition for Data Reconstruction

Kemas Zakaria*, Pramudita Satria Palar†, Lavi Rizki Zuhal‡
Institut Teknologi Bandung, Bandung, West Java, Indonesia, 40132

Joseph Morlier§
ICA, Université de Toulouse, ISAE–SUPAERO, INSA, CNRS, MINES ALBI, UPS, Toulouse, France

Many engineering problems are governed by complex governing equations that are difficult and typically require high computational costs to solve. Machine learning and surrogate modelling aid such an endeavour by providing a cheap-to-evaluate prediction model that acts as a replacement of the original model. While most research focuses on predicting scalar values (e.g., lift and drag), predicting the solution field is also of interest in many practical engineering and scientific applications. This paper proposes a Physics-Informed Proper Orthogonal Decomposition (POD) technique that improves the solution field prediction by enforcing governing equations as a loss penalty. The proposed idea utilizes a reduced-order modeling technique based on POD to decompose solution snapshots into singular vectors and values. A Gaussian Process Regression is then utilized to predict the singular values from variable parameters. The predicted singular values from the data of the problem are then adjusted via optimization to minimize the physics-informed loss and achieve better prediction. In this paper, we illustrate the efficacy of the proposed method on simple two-dimensional partial differential equations. The result clearly shows that the proposed physics-informed POD outperforms the conventional POD in terms of approximation error.

I. Introduction

One of the exciting current research trends in engineering focuses on utilizing machine learning and data science technology to aid design optimization and exploration [1–4], acquire hidden physical insights [5] or even solve the governing equation of the problem itself [6, 7]. Currently, two machine learning paradigms are extensively used and studied in engineering: (1) data-driven approaches, which purely exploit available data, and (2) physics-informed approaches, which utilize a priori information about the physics of the problem. Data-driven approaches are generally used for predicting quantities of interests [8, 9], multi-fidelity modeling [10, 11], and flowfield reconstruction [12] by solely relying on data (i.e., no prior physics information are involved). On the other hand, the physics-informed approach leverages physical laws to improve the performance of the data-driven approach by imposing certain physical laws into surrogate models [13, 14], replacing specific modeling to help numerical solver [15], or even acts as the solver of the governing equations by using machine learning approach techniques such as neural networks [6, 16]. Moreover, unlike a traditional numerical approach, the physics-informed machine learning method can exploit the solution of a simpler equation to help predict the solution of a more complex governing equation [17].

Two main points motivate the development of the proposed method. First, although predicting scalar values is important, fast prediction of a physical field is also essential in many engineering applications. In this regard, reduced-order models aim to provide a quick-to-predict solution that significantly helps in the design, analysis, and optimization process [18]. Furthermore, uncertainty quantification of a physical field is another crucial application that requires fast prediction of multiple outputs [19]. The second motivation is that many of the existing machine learning methods in this field are purely data-driven; thus, ignoring existing prior information or knowledge such as the governing equation of the physical problem [20–22]. However, as mentioned before, most existing prediction or reconstruction methods do not incorporate physical laws. The main aim of injecting physical laws into a machine learning method is to reduce the required data for accurate prediction since engineering data is typically expensive.

*Graduate Student, Department of Aeronautics and Astronautics, Institut Teknologi Bandung.
†Assistant Professor, Faculty of Mechanical and Aerospace Engineering, Institut Teknologi Bandung.
‡Associate Professor, Faculty of Mechanical and Aerospace Engineering, Institut Teknologi Bandung.
§Professor.

The proper orthogonal decomposition (POD) is a versatile and robust method that has been widely applied for field reconstruction [23, 24], modal analysis [25, 26], design optimization [27–29], and uncertainty quantification [30, 31]. POD is a model order reduction technique that decomposes a physical field into the linear combination of orthogonal eigenfunctions multiplied by the coefficients. To construct POD modes, one needs to collect the snapshots obtained by varying the input parameters or from temporal evolution if the goal is to decompose a transient physical field. The orthogonal nature of POD decomposition allows easy interpretation of the produced model, which is especially insightful in modal analysis. The conventional POD is purely data-driven because no physics information is involved during the process. However, recent advancements in machine learning have shown that injecting prior knowledge or physics information can improve the accuracy of surrogate model [5]. The most notable algorithm is the physics-informed neural network that takes advantage of physical laws to constrain the machine learning prediction [6]. Another example of algorithms is the physics-informed Gaussian process (GP) [32, 33]. Besides the aforementioned well-known algorithm, there is also a potential for utilizing the physics-informed framework for other data-driven algorithms such as POD.

In this paper, we propose a Physics Informed POD (PI-POD) method that takes advantage of physical laws to reconstruct the solution of the engineering problem. Our method leverages a GP surrogate model [34] to aid in predicting POD coefficients. Similar strategies have been explored previously in the context of fluid-structure interaction [9] and multidisciplinary optimization [4]. Notice that no physics is involved during the construction of our GP model since it primarily acts as a tool to map POD coefficients. However, there are variants of physics-GP in which the physics is incorporated into the GP formulation [13, 14, 33]. This paper explains the working mechanism of the proposed method and presents some preliminary results on a relatively simple problem.

The rest of this paper is structured as follows: Section II explains the details of our proposed method. Section III presents the demonstration of PI-POD in a two-dimensional partial differential equation (PDE). Section IV discusses the result of our proposed method and its comparison with conventional POD. Section V concludes the current results of PI-POD with a pointer of future works.

II. Method

In this section, we explain the main idea behind our proposed method. This section are divided into three parts: the data-driven POD part, the proposed PI-POD part which depends on the governing equation, and the last part explains the proposed method.

A. Proper Orthogonal Decomposition

Consider a set of solutions $\mathbf{A}$ of a governing PDE equation, with N data for each solution and M solutions, denoted as

$$\mathbf{A} = [A_1, A_2, ..., A_M] \in \mathbb{R}^{N\times M} \tag{1}$$

In fluid dynamics literature, each solution in the solution matrix $\mathbf{A}$ is usually called a "snapshot", so we are going to refer to these solutions as "snapshots". In this regard, the M solutions are obtained from varying the boundary conditions or other parameters (e.g., Reynolds number). The N rows in solution matrix $\mathbf{A}$ refers to the number of points in the PDE solution.

We can perform the singular value decomposition (SVD) procedure on the solution matrix $\mathbf{A}$ to obtain the singular vectors $\mathbf{U}$ and $\mathbf{V}$, and singular values $\mathbf{\Sigma}$

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_{i=1}^{r} \sigma_i \xi_i \psi_i^T, \tag{2}$$

where

$$\begin{aligned} &\mathbf{U} = [\xi_1, ..., \xi_r] \in \mathbb{R}^{N\times r} \\ &\mathbf{V} = [\psi_1, ..., \psi_r] \in \mathbb{R}^{M\times r} \\ &\mathbf{\Sigma} = \mathrm{diag}(\sigma_1, ..., \sigma_r) \in \mathbb{R}^{r\times r} \\ &r \leq \min(N, M) \end{aligned} \tag{3}$$

Due to the unitary property of $\mathbf{U}$, we can calculate a coefficient matrix $\mathbf{\Delta}$ such that $\mathbf{A} = \mathbf{U}\mathbf{\Delta}$.

$$\begin{aligned} &\mathbf{\Delta} = \mathbf{U}^T\mathbf{A} \\ &\mathbf{\Delta} = [\delta_1, ..., \delta_r] \in \mathbb{R}^{M\times r} \end{aligned} \tag{4}$$

To reduce the computational cost of POD, the solution matrix $\mathbf{A}$ is usually approximated by truncating some columns in $\mathbf{U}$. The approximated solution matrix $\hat{\mathbf{A}}$ is computed as

$$\hat{\mathbf{A}} = \hat{\mathbf{U}}\hat{\mathbf{\Delta}}, \tag{5}$$

where

$$\hat{\mathbf{U}} = [\xi_1, ..., \xi_k] \in \mathbb{R}^{N\times k} \tag{6}$$

$$\hat{\mathbf{\Delta}} = \hat{\mathbf{U}}^T\mathbf{A} \tag{7}$$

The parameter k is determined by defining an error tolerance ϵ_{tol}

$$\mathbb{E}(k) = \frac{\sum_{i=1}^{k}\sigma_i^2}{\sum_{i=1}^{r}\sigma_i^2} \geq 1 - \epsilon_{tol}^2 \tag{8}$$

where $\mathbb{E}(k)$ denotes the energy percentage contained within the first k modes.

We can then reconstruct a solution, given the POD coefficients, by

$$\hat{u^*} = \hat{\mathbf{U}}\Delta^* \tag{9}$$

where $\hat{u^*}$ is the predicted field and Δ^* is the given POD coefficients for prediction. If we are given a solution u^* instead, we can calculate the corresponding POD coefficients by

$$\Delta^* = \hat{\mathbf{U}}^T u^*. \tag{10}$$

An example of situations where we want to reconstruct a solution field is predicting a flow field given untested flow conditions (e.g., untested Mach number or Reynolds number). To that end, one needs to estimate Δ^* so that the flow field can be reconstructed without running more computational fluid dynamics simulations. In this paper, we use the SVD implementation of MATLAB to perform the decomposition. As explained in the next section, our proposed method leverages a surrogate model based on GPR to estimate the POD coefficients for untested conditions. In this paper, we use MATLAB's SVD subroutine that removes extra rows and columns of zeros from the diagonal matrix of singular values and the corresponding rows and columns of the singular vector.

B. Gaussian Process Regression

GPR, or also known as Kriging, is a type of surrogate model that takes an input $\mathbf{x} \in \mathbb{R}^D$ to predict the outputs y, which are treated as realizations of Gaussian Processes. The prediction structure of a GPR model is given by

$$\hat{y}(\mathbf{x}) = \beta^T\phi(\mathbf{x}) + \mathbf{Z}(\mathbf{x}) \tag{11}$$

The first term $\beta^T\phi(\mathbf{x})$ represents the mean term, or also known as the trend function, while the second term $\mathbf{Z}(\mathbf{x})$ is the stochastic part of the GPR model, respectively. Each single point in the input space is treated as a Gaussian random variable, in which all points in the input are space are correlated with each other as modeled by the covariance function. The term $\phi(\mathbf{x})$ denotes a set of regression functions and β is the vector of the corresponding coefficients. In this paper, we utilize a constant trend (i.e., μ) for the sake of simplicity. The corresponding GPR prediction then becomes

$$\hat{y}(\mathbf{x}) = \mu + \mathbf{Z}(\mathbf{x}) \tag{12}$$

The stochastic term $\mathbf{Z}(\mathbf{x})$ with a constant trend can then be written as

$$\mathbf{Z}(\mathbf{x}) = \psi\mathbf{\Psi}^{-1}(\mathbf{y} - \mathbf{1}\mu) \tag{13}$$

where $\mathbf{\Psi}$ and ψ are correlation matrix and vector respectively. The covariance function used in this paper is the Gaussian correlation function, reads as

$$\Psi(i, j; \theta) = \exp\left(\frac{(x^i - x^j)^2}{2\theta}\right), \tag{14}$$

where θ is the length-scale of the correlation function. Notice that a multi-dimensional correlation function is simply constructed by the product of one-dimensional correlation functions. We then have a vector of lengthscale $\boldsymbol{\theta} = \{\theta_1, \theta_2, \ldots, \theta_D\}$ for a D-dimensional correlation for a D-dimensional approximation.

The vector of lengthscale $\boldsymbol{\theta}$ is determined by maximizing the so-called likelihood function, as given by

$$\ln(L) = -\frac{n}{2}\ln 2\pi - \frac{n}{2}(\hat{\sigma}^2) - \frac{1}{2}\ln \boldsymbol{\Psi} - \frac{(\mathbf{y} - \mathbf{1}\mu)\boldsymbol{\Psi}^{-1}(\mathbf{y} - \mathbf{1}\mu)}{2\hat{\sigma}^2}, \tag{15}$$

where

$$\mu = \frac{\mathbf{1}^T\boldsymbol{\Psi}^{-1}\mathbf{y}}{\mathbf{1}^T\boldsymbol{\Psi}\mathbf{1}} \tag{16}$$

and

$$\hat{\sigma}^2 = \frac{(\mathbf{y} - \mathbf{1}\mu)\boldsymbol{\Psi}^{-1}(\mathbf{y} - \mathbf{1}\mu)}{n} \tag{17}$$

The optimization of Eq.15 is a D-dimensional optimization problem with respect to $\boldsymbol{\theta}$. Once the optimum hyperparameters are found (including $\boldsymbol{\theta}$, μ, and σ^2), we can then use GPR to make predictions at any point by using Eq.12.

GPR typically provides an accurate prediction for low sample size, a scenario that some supervised learning models cannot handle. Another reason why we picked GPR as the method of choice is that it naturally provides an uncertainty estimate for the mapping process in POD (explained in the next sections). Although not explored in this paper, the uncertainty estimate of GPR will be useful later for facilitating adaptive sampling to reduce the number of samples for the mapping.

C. Incorporating Physics Information

Suppose that partial-differential equation as the governing equation of the problem is given by

$$\mathcal{D}(u(\mathbf{x}, t)) = 0, \mathbf{x} \in \boldsymbol{\Omega}, t \in [0, T] \tag{18}$$

where $u(\mathbf{x}, t)$ represents the solution of the partial-differential equation 18, $\mathcal{D}$ represent the non-linear partial-differential operator, and $\boldsymbol{\Omega}$ is a subset of $\mathbb{R}^D$ in D-dimensional space. In this paper, we only consider a PDE defined in a spatial domain. The governing equation of the problem is be treated as loss function to adjust the singular values from the singular value decomposition. The adjustment variable, α is computed from the minimization problem 19.

$$\begin{aligned} \min_{\alpha} \quad & \mathcal{L}(\hat{u}) = \mathcal{D}(\hat{u}) + \gamma_{bc}\epsilon_{bc} \\ & \hat{u} = \hat{\mathbf{U}}^h(\delta^* + \alpha) \end{aligned} \tag{19}$$

where the term γ_{bc} is a constant to impose boundary condition from the governing equation and ϵ_{bc} denotes the error at the boundary condition. Note that we need to perform this minimization problem each time we want to make a prediction using PI-POD, so the term δ^* here denotes the predicted POD coefficient for this prediction. The term α has size equal to the number of truncated singular values, so the dimensionality of the minimization problem in Eq. 19 equals the size of the truncated singular values. The value of variable γ_{bc} is defined to be adaptive, in the sense that when the value of ϵ_{bc} gets too high when minimizing the term $\mathcal{D}(\hat{u})$, then the value of γ_{bc} is increased into significantly high value. In this paper, we set γ_{bc} equals 10^7 when $\epsilon_{bc} > 0.05$, and 1 otherwise. Such a condition will force the model not to violate the boundary condition during the optimization process. There is no definitive reason why we choose 10^7 value for γ_{bc}. Setting γ_{bc} to other value would result in roughly the same prediction, as long as the value is high enough. An adaptive setting of γ_{bc} is one subject of future works.

D. Physics-Informed POD

Our objective is to reconstruct the field solution from POD and tune it according to the governing equations. We can create a solution matrix by concatenating all snapshots as columns in the solution matrix:

$$\mathbf{A} = [u_1, ..., u_M] \in \mathbb{R}^{N \times M}, \tag{20}$$

where $\mathbf{A}$ is the solution matrix for our problem. The pseudo-algorithm of the PI-POD method is written in Algorithm 1

For the GPR-assisted mapping procedure, the next step is to build r GPR models where r is the number of bases retained in the POD model, and each GPR model predicts each POD coefficient as a function of all variable parameters in our problem. Up until this point, the proposed method is still data-driven. The main goal here is to predict $\boldsymbol{\Delta}$ for

Algorithm 1 Constructing Physics-Informed POD (Offline Phase)

Require: Solution Matrix $\mathbf{A}$
  Perform SVD to Solution Matrix $\mathbf{A}$
Require: User-defined ϵ_{tol} value
  Compute k by using Eq. 8.
  Truncate singular sector $\mathbf{U}^l$ and $\mathbf{U}^h$ by its corresponding k column
  Calculate truncated POD coefficients $\mathbf{\Delta}$ by using Eq. 7 by its corresponding singular vector $\mathbf{U}$
  Create GPR mappings between $\mathbf{\Delta}^l$ as inputs to $\mathbf{\Delta}^h$ as outputs

the desired solution at an untested parameter. The main novelty of the proposed method is the physics-informed part when we reconstruct the predicted solution. The central idea here is to refine the predicted POD coefficients so that the reconstructed solution, ideally, satisfies the physical laws and boundary conditions. The POD coefficients are first predicted using GPR models from unobserved parameters. Finally, the proposed method applies a gradient-based optimization to minimize the loss function by optimizing α (see Eq. 19), with predicted POD coefficients from the GPR model treated as the initial guess for the gradient-based minimization. The pseudo-algorithm for prediction using Physics-Informed POD is written in Algorithm 2.

Algorithm 2 Prediction using Physics-Informed POD (Online Phase)

Require: Parameters $\gamma_{bc}, \gamma_{\alpha}$
  Compute POD coefficients by GPR model from untested parameters
  Compute α by solving minimization problem Eq. 19
  Compute predicted solution $\hat{u}_*^h$ with Eq. 9 by using truncated singular vector $\hat{\mathbf{U}}$

The predicted solution $\hat{u}_*$ is the final output of the Physics-Informed POD. Notice that the procedure outlined in Algorithm 1 is only performed once using the training data, while the prediction procedure in Algorithm 2 is performed every time we want to predict a different solution.

III. An Illustrative Example

A. Poisson Equation

For demonstration purposes, we use a simple linear two-dimensional partial differential equation. The governing equation used for this problem is the Poisson equation, reads as

$$\nabla^2 u = 10, \tag{21}$$

In this problem, the spatial domain is a 100x100 grid two-dimensional space within $[0, 1]^2$, with Dirichlet boundary conditions on the four sides surrounding the domain. The boundary conditions imposed are the parameters of interests; That is, we aim to predict the physical field under variations in the boundary conditions. The samples for POD and PI-POD are then taken by sampling from the domain of the input parameters.

There are two cases for this test problem. The first compares the predictive power of PI-POD and conventional POD with a small number of snapshots, and the second compares the robustness performance of PI-POD and conventional POD with more snapshots. Notice that the conventional POD here does not use the physics-informed correction. Instead, the conventional POD directly uses the mapped coefficients to reconstruct the high-fidelity solution.

The ϵ_{tol} value is chosen to be 0.01 for all cases in this problem. The root-mean-squared-error (RMSE) is chosen to be the error metric for this problem, written as

$$\epsilon_{rmse} = \sqrt{\frac{\sum_i^N (\hat{u}_i - u_i)^2}{N}} \tag{22}$$

where $\hat{u}$ represents the predicted field and N is the number of nodes in the spatial domain. Notice that ϵ_{rmse} is calculated for a single combination of untested parameters. For multiple combinations (e.g., as in optimization or uncertainty analysis), we can depict multiple ϵ_{rmse} is the form of a boxplot to analyze the robustness of the algorithm.

1. Case 1: Few snapshots case

We only use six snapshots as training data to create our model for the first case of this problem. The snapshots are created by using six variations of boundary conditions. The goal is then to reconstruct the solution field for boundary conditions outside the training snapshots. The boundary conditions for the training snapshots are written as follows:

$$\begin{aligned}\text{B.C. 1: } & u(x,0) = 0 \\ & u(x,1) = \sin(\pi x) \\ & u(0,y) = -\sin(\pi x) \\ & u(1,y) = 0\end{aligned} \qquad \begin{aligned}\text{B.C. 2: } & u(x,0) = -\sin(\pi x) \\ & u(x,1) = 0 \\ & u(0,y) = 0 \\ & u(1,y) = \sin(\pi x)\end{aligned} \qquad \begin{aligned}\text{B.C. 3: } & u(x,0) = 2\sin(\pi x) \\ & u(x,1) = -0.5\sin(\pi x) \\ & u(0,y) = 0 \\ & u(1,y) = 0\end{aligned}$$

$$\begin{aligned}\text{B.C. 4: } & u(x,0) = 0 \\ & u(x,1) = 0 \\ & u(0,y) = -2\sin(\pi x) \\ & u(1,y) = 0.5\sin(\pi x)\end{aligned} \qquad \begin{aligned}\text{B.C. 5: } & u(x,0) = -2\sin(\pi x) \\ & u(x,1) = 2\sin(\pi x) \\ & u(0,y) = 0.5\sin(\pi x) \\ & u(1,y) = -0.5\sin(\pi x)\end{aligned} \qquad \begin{aligned}\text{B.C. 6 : } & u(x,0) = 0.5\sin(\pi x) \\ & u(x,1) = -2\sin(\pi x) \\ & u(0,y) = 0.5\sin(\pi x) \\ & u(1,y) = -0.5\sin(\pi x)\end{aligned}$$

The accuracy of the constructed POD models is evaluated using the following three boundary conditions as our testing data:

$$\begin{aligned}\text{B.C. 1: } & u(x,0) = 0 \\ & u(x,1) = \sin(\pi x) \\ & u(0,y) = -2\sin(\pi x) \\ & u(1,y) = 0\end{aligned} \qquad \begin{aligned}\text{B.C. 2: } & u(x,0) = 0 \\ & u(x,1) = 1.3\sin(\pi x) \\ & u(0,y) = -0.8\sin(\pi x) \\ & u(1,y) = 0.6\sin(\pi x)\end{aligned} \qquad \begin{aligned}\text{B.C. 3: } & u(x,0) = -6\sin(\pi x) \\ & u(x,1) = 3\sin(\pi x) \\ & u(0,y) = 7\sin(\pi x) \\ & u(1,y) = -5\sin(\pi x)\end{aligned}$$

2. Case 2: Medium size snapshots case

In the second case, we generated 100 snapshots with random variations in the boundary conditions. The variations in the boundary conditions are modelled by varying the vector of constants $\boldsymbol{\lambda}$, where $\boldsymbol{\lambda} = \{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}$, as shown in the following:

$$\begin{aligned} u(x,0) &= \lambda_1 \sin(\pi x) \\ u(x,1) &= \lambda_2 \sin(\pi x) \\ u(0,y) &= \lambda_3 \sin(\pi y) \\ u(1,y) &= \lambda_4 \sin(\pi y) \end{aligned} \tag{23}$$

where λ_i, $i = 1, 2, 3, 4$, is a normally distributed random variable with the distribution of $\mathcal{N}(0, 2)$. In total, we generated 100 snapshots for this case, in which 70 and 30 snapshots were used as training and validation snapshots, respectively. The PI-POD and POD were constructed for three different sub-cases with different sizes of snapshots: 10, 20, and 30 snapshots for each sub-case (taken from 70 training snapshots with 50 random shuffles).

The metric that we use in the boxplot for the second case is the Averaged-RMSE, which is computed using the following equation

$$\text{Averaged-RMSE} = \frac{\sum_{i=1}^{N_v} \epsilon_{rmse}}{N_v}, \tag{24}$$

where N_v is the number of validation snapshots used.

Table 1 RMSE of POD and PI-POD for Case 1, first validation data.

Model	ϵ_{RMSE}	ϵ_{RMSE} at boundaries
PI-POD	3.35	1.12
POD	13.13	17.61

IV. Results

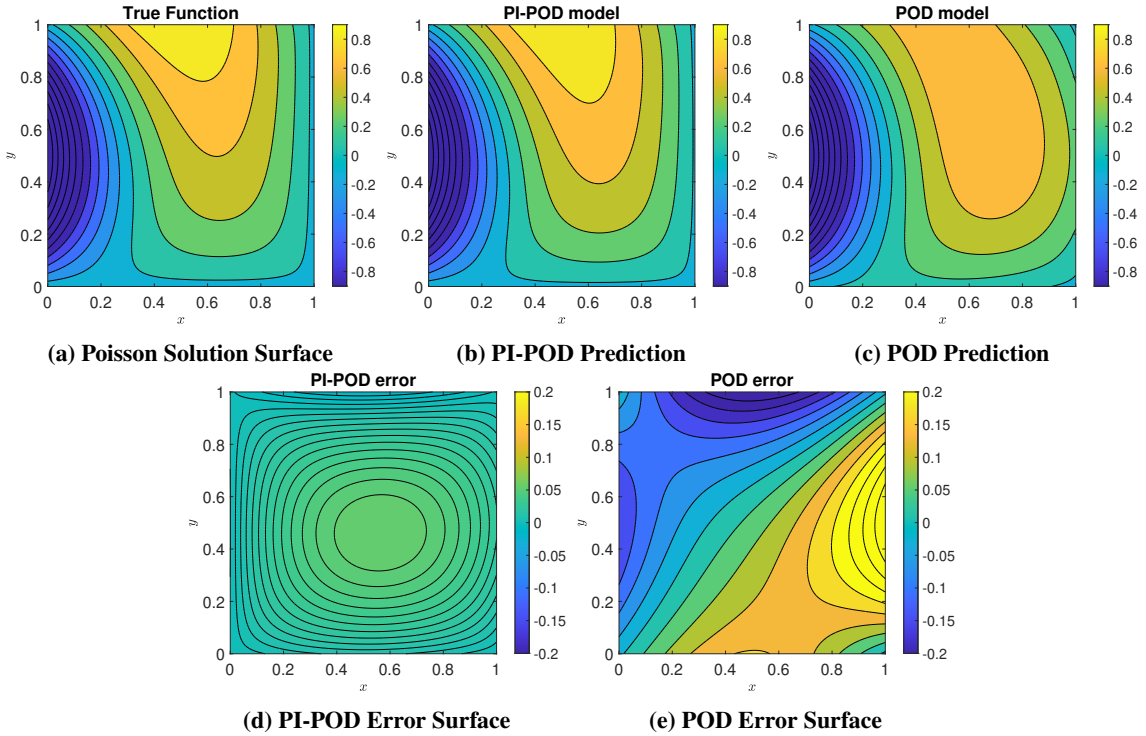
A. Poisson Equation: Case 1 Results

For case 1, we analyze the reconstructed solution for each validation data separately so that we can perform a more thorough analysis.

1. First validation data

For the first validation snapshot, the resulting prediction and the error surfaces for Physics-Informed POD and conventional POD are plotted in Fig. 1. Besides the RMSE in the whole domain, the RMSE at the boundary points is also calculated. For this result, the accuracy for both Physics-Informed POD and the conventional POD is summarized in Table 1. It is clear that the Physics-Informed POD outperforms the conventional POD in terms of overall accuracy, as indicated by the lower total RMSE and RMSE in the entire domain and at the boundary conditions. It can also be seen from Fig. 1 that the reconstructed solution from the PI-POD is closer to the true solution than the conventional POD.

The error level of the PI-POD is notably lower than the conventional POD. It can also be seen that the error from the PI-POD is the highest at the center of the computational domain, with the errors are lower at locations close to the boundaries. This indicates the success of PI-POD in better predicting the solution by an extra adjustment to satisfy the governing equations. On the other hand, the errors from the POD are more spread across the computational domain, with the highest errors observed at the boundaries.



(a) Poisson Solution Surface **(b) PI-POD Prediction** **(c) POD Prediction**

(d) PI-POD Error Surface **(e) POD Error Surface**

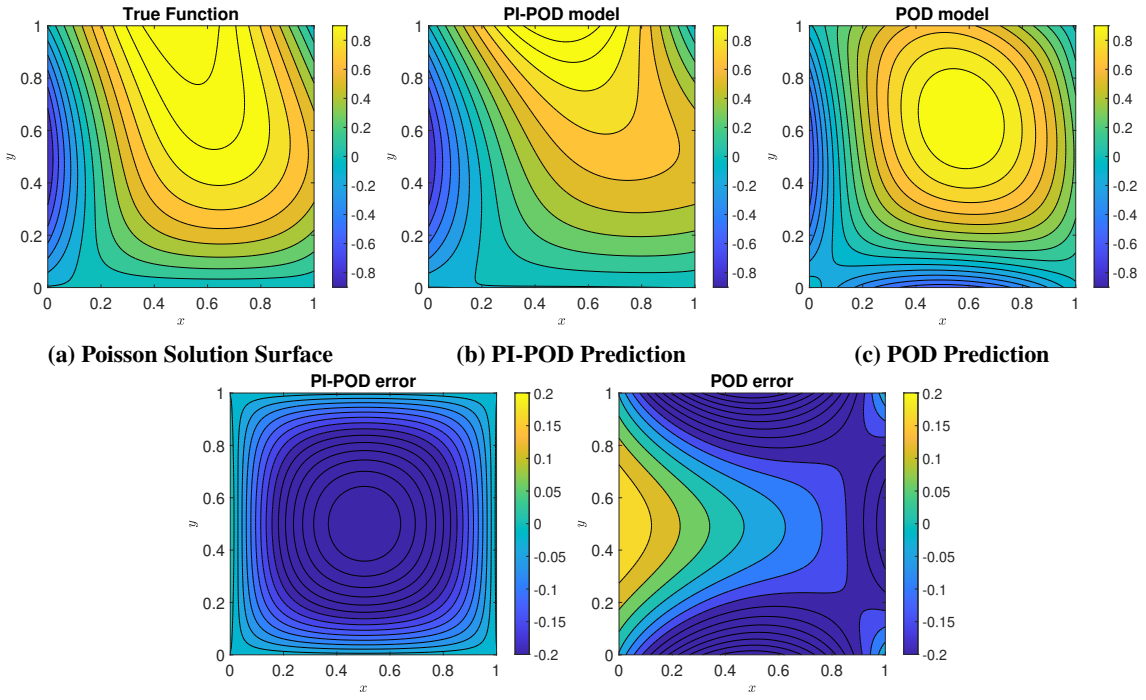
Fig. 1 Results for Case 1, first validation data, and the corresponding error surfaces.

Model	ϵ_{RMSE}	ϵ_{RMSE} at boundaries
PI-POD	18.59	1.12
POD	20.09	31.97

Table 2 RMSE of POD and PI-POD for Case 1, second validation data.

2. Second validation data

For the second validation snapshot, the resulting prediction for PI-POD and conventional POD are plotted in Fig. 2. The accuracy of both PI-POD and conventional POD for the second validation data is summarized in Table 2. As shown in Table 2, the PI-POD model still outperforms POD in terms of overall accuracy, indicated by the higher RMSE value of the latter. Although there is still a mismatch between the true solution and PI-POD, the solution from PI-POD closely resembles the true solution better than POD. Furthermore, notice that the RMSE at the boundaries is higher for the conventional POD than the PI-POD. Another important result to notice is the difference in the error surface between the PI-POD and the POD model (see Fig. 2). The error surface of the PI-POD model spreads more evenly, with noticeably small errors around the boundaries. On the other hand, the error surface for the POD model is significantly high at locations close to the boundaries. Interestingly, the errors from POD are the highest at the boundary with zero values (i.e., $x = 0$). Although the error surface of PI-POD looks symmetric compared to the first validation data, there is no particular reason why it should be symmetric. In fact, despite not being visible from the error surface, there is a slight asymmetry in the error surface of PI-POD.



(a) Poisson Solution Surface **(b) PI-POD Prediction** **(c) POD Prediction**

(d) PI-POD Error Surface **(e) POD Error Surface**

Fig. 2 Results for Case 1, second validation data, and the corresponding error surfaces.

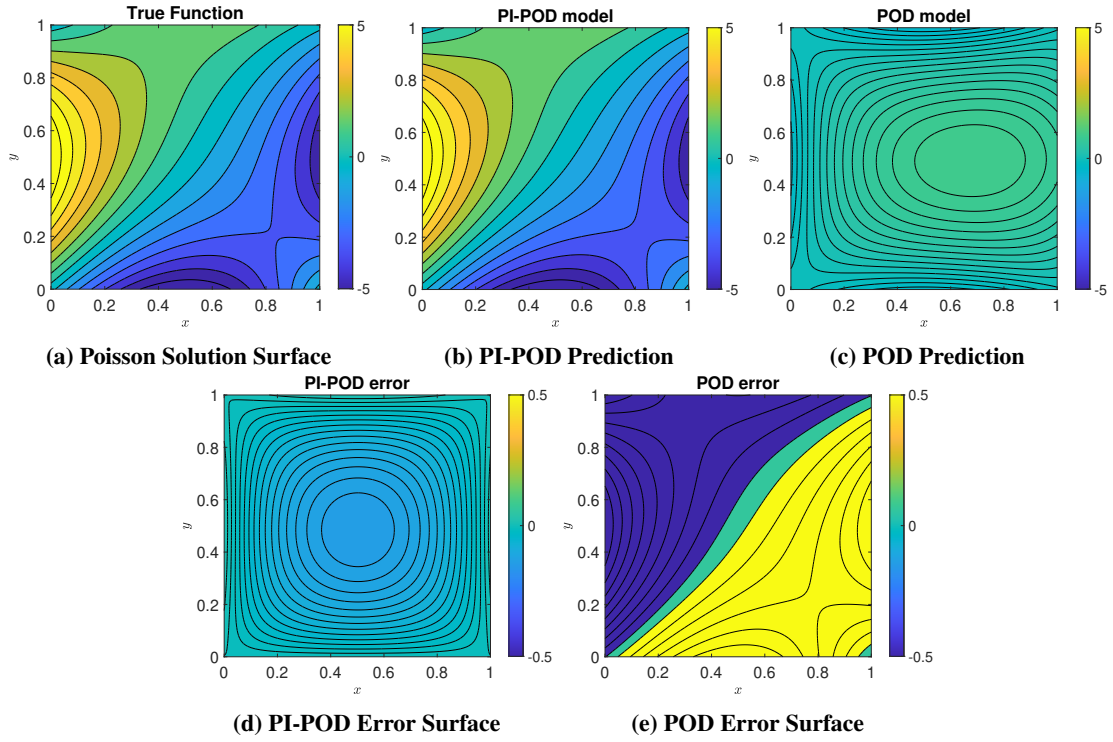
3. Third validation data

Finally, the resulting prediction for PI-POD and conventional POD for the third validation data are plotted in figure 3. For this result, the accuracy for PI-POD and conventional POD is summarized in Table 3. The boundary condition imposed on this case has a significantly higher amplitude compared to the boundary conditions of the training data, which is why we expected that predicting this third validation data would be more difficult. Indeed, from Table 3, it can

Model	ϵ_{RMSE}	ϵ_{RMSE} at boundaries
PI-POD	7.73	1.12
POD	283	391

Table 3 RMSE of POD and PI-POD for Case 1, third validation data.

be seen that the POD model produces a highly inaccurate prediction for this case. The reason why POD performs poor on the third validation data is because it needs to extrapolate outside the training data. On the other hand, PI-POD still managed to yield a highly accurate prediction for the third validation data. Such high accuracy is due to the incorporation of physics information that gives the ability to better extrapolate outside the training data by obeying the corresponding governing equation.



(a) Poisson Solution Surface **(b) PI-POD Prediction** **(c) POD Prediction**

(d) PI-POD Error Surface **(e) POD Error Surface**

Fig. 3 Results for Case 1, third validation data, and the corresponding error surfaces.

B. Poisson Equation: Case 2 Results

The second case is a more realistic application of POD because it acts more like an interpolation rather than an extrapolation, as in the third validation data of Case 1. The boxplots of RMSE for both types of POD are plotted in Fig. 4. It is interesting that the proposed method consistently outperforms the conventional POD for all variations in the number of snapshots. The better accuracy of the PI-POD for the smallest number of snapshots (i.e., ten snapshots) is a clear evidence of how readjusting the coefficients with physics information can improve the accuracy of POD reconstruction. Furthermore, it is interesting to see that the PI-POD yields lower errors even for a high number of snapshots. The results for Case 2 then show the potential of PI-POD to reconstruct high-fidelity solutions, with potential applications in uncertainty quantification and design optimization.

V. Conclusion and Future Works

This paper proposes a method that reconstructs a PDE solution based on Physics-Informed Proper Orthogonal Decomposition (PI-POD). The proposed method improves the POD-based reconstruction method by incorporating physics information to increase accuracy and force the prediction to satisfy the physics and boundary conditions. Further, the proposed method also takes advantage supervised learning method (i.e., GPR) to map between POD coefficients and variable parameters. We demonstrate the efficacy of the proposed method on a simple two-dimensional PDE (i.e., Poisson equation) in a squared-shaped domain. Our proposed method consistently outperforms conventional POD in terms of accuracy and robustness. The PI-POD method especially yields better accuracy than POD at the boundaries since the former also incorporates boundary losses in the prediction process.

For future works, we plan to study the efficacy of PI-POD on a set of physical problems with more complex PDEs. In particular, we aim to apply PI-POD on a set of structural and fluid mechanics problems. Furthermore, the use of other methods than GPR should also be studied. It is also important to investigate the impact of PI-POD parameters (e.g., α_{BC}), and develop an adaptive method to automatically tune the parameters.

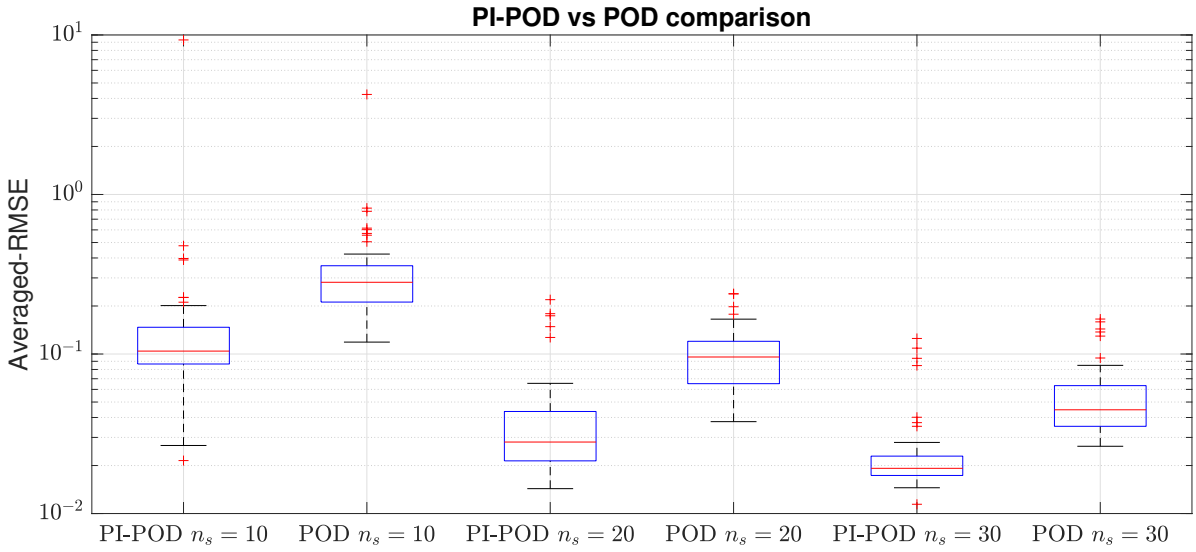


Fig. 4 The RMSE boxplot of PI-POD and POD for Case 2.

Acknowledgement

This work is supported by The Ministry of Finance of Republic of Indonesia, through Indonesia Endowment Fund for Education Institution (*Lembaga Pengelola Dana Pendidikan Indonesia*), under International Conference Funding Programme for doctoral students. The authors also acknowledge the financial support from Institut Teknologi Bandung.

References

[1] Li, M., and Wang, Z., "Surrogate model uncertainty quantification for reliability-based design optimization," *Reliability Engineering & System Safety*, Vol. 192, 2019, p. 106432.

[2] Zuhal, L. R., Zakaria, K., Palar, P. S., Shimoyama, K., and Liem, R. P., "Polynomial-chaos–kriging with gradient information for surrogate modeling in aerodynamic design," *AIAA journal*, Vol. 59, No. 8, 2021, pp. 2950–2967.

[3] Bouhlel, M. A., Hwang, J. T., Bartoli, N., Lafage, R., Morlier, J., and Martins, J. R., "A Python surrogate modeling framework with derivatives," *Advances in Engineering Software*, Vol. 135, 2019, p. 102662.

[4] Dubreuil, S., Bartoli, N., Berthelin, G., Vila, O. C., Gogu, C., Lefebvre, T., Morlier, J., and Salaün, M., "Development of MDO Formulations based on Discplinary Surrogate Models by Gaussian Processes," *AeroBest*, 2021.

[5] Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L., "Physics-informed machine learning," *Nature Reviews Physics*, Vol. 3, No. 6, 2021, pp. 422–440.

[6] Raissi, M., Perdikaris, P., and Karniadakis, G. E., "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational physics*, Vol. 378, 2019, pp. 686–707.

[7] Mao, Z., Jagtap, A. D., and Karniadakis, G. E., "Physics-informed neural networks for high-speed flows," *Computer Methods in Applied Mechanics and Engineering*, Vol. 360, 2020, p. 112789.

[8] Laurenceau, J., and Sagaut, P., "Building efficient response surfaces of aerodynamic functions with kriging and cokriging," *AIAA journal*, Vol. 46, No. 2, 2008, pp. 498–507.

[9] Chiplunkar, A., Bosco, E., and Morlier, J., "Gaussian process for aerodynamic pressures prediction in fast fluid structure interaction simulations," *World Congress of Structural and Multidisciplinary Optimisation*, Springer, 2017, pp. 221–233.

[10] Fischer, C. C., Grandhi, R. V., and Beran, P. S., "Bayesian low-fidelity correction approach to multi-fidelity aerospace design," *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2017, p. 0133.

[11] Ghoreyshi, M., Badcock, K., and Woodgate, M., "Integration of multi-fidelity methods for generating an aerodynamic model for flight simulation," *46th AIAA aerospace sciences meeting and exhibit*, 2008, p. 197.

[12] Yu, J., and Hesthaven, J. S., "Flowfield reconstruction method using artificial neural network," *Aiaa Journal*, Vol. 57, No. 2, 2019, pp. 482–498.

[13] Chiplunkar, A., Rachelson, E., Colombo, M., and Morlier, J., "Adding Flight Mechanics to Flight Loads Surrogate Model using Multi-Output Gaussian Processes," *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2016, p. 4000.

[14] Chiplunkar, A., Rachelson, E., Colombo, M., and Morlier, J., "Sparse physics-based Gaussian process for multi-output regression using variational inference," 2016.

[15] Yang, X., Zafar, S., Wang, J.-X., and Xiao, H., "Predictive large-eddy-simulation wall modeling via physics-informed neural networks," *Physical Review Fluids*, Vol. 4, No. 3, 2019, p. 034602.

[16] Kashinath, K., Mustafa, M., Albert, A., Wu, J., Jiang, C., Esmaeilzadeh, S., Azizzadenesheli, K., Wang, R., Chattopadhyay, A., Singh, A., et al., "Physics-informed machine learning: case studies for weather and climate modelling," *Philosophical Transactions of the Royal Society A*, Vol. 379, No. 2194, 2021, p. 20200093.

[17] Aliakbari, M., Mahmoudi, M., Vadasz, P., and Arzani, A., "Predicting high-fidelity multiphysics data from low-fidelity fluid flow and transport solvers using physics-informed neural networks," *International Journal of Heat and Fluid Flow*, Vol. 96, 2022, p. 109002.

[18] Yu, J., Yan, C., and Guo, M., "Non-intrusive reduced-order modeling for fluid problems: A brief review," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 233, No. 16, 2019, pp. 5896–5912.

[19] Tripathy, R. K., and Bilionis, I., "Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification," *Journal of computational physics*, Vol. 375, 2018, pp. 565–588.

[20] Chang, Y.-H., Wang, X., Zhang, L., Li, Y., Mak, S., Wu, C.-F. J., and Yang, V., "Reduced-Order Modeling for Complex Flow Emulation by Common Kernel-Smoothed Proper Orthogonal Decomposition," *AIAA Journal*, Vol. 59, No. 9, 2021, pp. 3291–3303.

[21] Kramer, B., and Willcox, K. E., "Nonlinear model order reduction via lifting transformations and proper orthogonal decomposition," *AIAA Journal*, Vol. 57, No. 6, 2019, pp. 2297–2307.

[22] Bertram, A., Bekemeyer, P., and Held, M., "Fusing distributed aerodynamic data using Bayesian Gappy Proper Orthogonal Decomposition," *AIAA Aviation 2021 Forum*, 2021, p. 2602.

[23] Qamar, A., and Sanghi, S., "Steady supersonic flow-field predictions using proper orthogonal decomposition technique," *Computers & fluids*, Vol. 38, No. 6, 2009, pp. 1218–1231.

[24] Bui-Thanh, T., Damodaran, M., and Willcox, K., "Proper orthogonal decomposition extensions for parametric applications in compressible aerodynamics," *21st AIAA applied aerodynamics conference*, 2003, p. 4213.

[25] Taira, K., Brunton, S. L., Dawson, S. T., Rowley, C. W., Colonius, T., McKeon, B. J., Schmidt, O. T., Gordeyev, S., Theofilis, V., and Ukeiley, L. S., "Modal analysis of fluid flows: An overview," *Aiaa Journal*, Vol. 55, No. 12, 2017, pp. 4013–4041.

[26] Taira, K., Hemati, M. S., Brunton, S. L., Sun, Y., Duraisamy, K., Bagheri, S., Dawson, S. T., and Yeh, C.-A., "Modal analysis of fluid flows: Applications and outlook," *AIAA journal*, Vol. 58, No. 3, 2020, pp. 998–1022.

[27] Toal, D. J., Bressloff, N. W., Keane, A. J., and Holden, C. M., "Geometric filtration using proper orthogonal decomposition for aerodynamic design optimization," *AIAA journal*, Vol. 48, No. 5, 2010, pp. 916–928.

[28] Zhu, Y., Ju, Y., and Zhang, C., "Proper orthogonal decomposition assisted inverse design optimisation method for the compressor cascade airfoil," *Aerospace Science and Technology*, Vol. 105, 2020, p. 105955.

[29] Oyama, A., Nonomura, T., and Fujii, K., "Data mining of Pareto-optimal transonic airfoil shapes using proper orthogonal decomposition," *Journal of Aircraft*, Vol. 47, No. 5, 2010, pp. 1756–1762.

[30] Mohammadi, A., and Raisee, M., "Efficient uncertainty quantification of stochastic heat transfer problems by combination of proper orthogonal decomposition and sparse polynomial chaos expansion," *International Journal of Heat and Mass Transfer*, Vol. 128, 2019, pp. 581–600.

[31] Mohammadi, A., Shimoyama, K., Karimi, M. S., and Raisee, M., "Efficient uncertainty quantification of CFD problems by combination of proper orthogonal decomposition and compressed sensing," *Applied Mathematical Modelling*, Vol. 94, 2021, pp. 187–225.

[32] Raissi, M., and Karniadakis, G. E., "Hidden physics models: Machine learning of nonlinear partial differential equations," *Journal of Computational Physics*, Vol. 357, 2018, pp. 125–141.

[33] Yang, X., Barajas-Solano, D., Tartakovsky, G., and Tartakovsky, A. M., "Physics-informed CoKriging: A Gaussian-process-regression-based multifidelity method for data-model convergence," *Journal of Computational Physics*, Vol. 395, 2019, pp. 410–431.

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