

Gaussian process surrogate with physical law-corrected prior for multi-coupled PDEs defined on irregular geometry

Pucheng Tang^a, Hongqiao Wang^{b,d},
 Qian Chen^c, Wenzhou Lin^{c*}, Heng Yong^c

^a*School of Artificial Intelligence
 Wuhan University, Wuhan 430072, China*

^b*School of Mathematics and Statistics
 Central South University, Changsha 410083, China*

^c*Institute of Applied Physics and Computational Mathematics
 Beijing 100094, China*

^d*Institute of Mathematics, Henan Academy of Sciences
 Zhengzhou 450046, China*

Abstract

Parametric partial differential equations (PDEs) are fundamental mathematical tools for modeling complex physical systems, yet their numerical evaluation across parameter spaces remains computationally intensive when using conventional high-fidelity solvers. To address this challenge, we propose a novel physical law-corrected prior Gaussian process (LC-prior GP) surrogate modeling framework that effectively integrates data-driven learning with underlying physical constraints to flexibly handle multi-coupled variables defined on complex geometries. The proposed approach leverages proper orthogonal decomposition (POD) to parameterize high-dimensional PDE solutions via their dominant modes and associated coefficients, thereby enabling efficient Gaussian process (GP) surrogate modeling within a reduced-dimensional coefficient space. A key contribution lies in the incorporation of physical laws together with a limited number of parameter samples to correct the GP posterior mean, thus avoiding reliance on computationally expensive numerical solvers. Furthermore, interpolation functions are constructed to describe the mapping from the full parameter space to the physics-based correction term. This mapping is subsequently backpropagated to constrain the original GP surrogate, yielding a more physically consistent conditional prior. To handle irregular geometries, the radial basis

function-finite difference (RBF-FD) method is incorporated during training set computation, with its inherent differentiation matrices providing both computational efficiency and numerical accuracy for physical constraint optimization. The effectiveness of the proposed method is demonstrated through numerical experiments involving reaction-diffusion model, miscible flooding models and Navier-Stokes equations with multi-physics coupling defined on irregular domains. Additionally, the framework's applicability to parameter estimation is verified through multi parameters miscible flooding models and Korteweg-de Vries equation.

Keywords: Parametric partial differential equations; Gaussian process regression; Physical laws; Surrogate model; Radial basis function finite difference; Parameter estimation

1 Introduction

Parametric differential equations (DEs), including parametric ordinary differential equations (ODEs) and parametric partial differential equations (PDEs), are fundamental tools for modeling a wide range of scientific and engineering phenomena, particularly in systems with uncertain or complex dynamics [1]. These parameters may arise from physical properties, geometric configurations, environmental factors, or other system characteristics. The inclusion of parameters in DEs is crucial for enabling uncertainty quantification, sensitivity analysis, and optimization in real-world applications. For example, in fluid dynamics, parameters such as viscosity and the Reynolds number govern flow behavior, whereas in structural mechanics, material stiffness and damping coefficients determine system stability. Moreover, parametric DEs facilitate the study of how small input variations propagate through a system, supporting robust design and control strategies. Techniques such as parameter estimation and inverse problems rely on these models to infer unknown quantities from observed data. Neglecting parametric dependencies may lead to model oversimplification, resulting in inaccurate predictions or sub-optimal performance in critical applications. Accurate estimation of unknown parameter values and quantification of their effects on specific systems typically require extensive computational sampling, often involving thousands to millions of realizations. Traditional mesh-based numerical methods, while ensuring solution accuracy, incur prohibitive computational costs that hinder their applicability to complex engineering scenarios [2–6]. In recent years, physics-informed neural networks and neural operator methods under deep learning frameworks have emerged as prominent research focuses [7, 8]. These approaches exhibit superior capabilities over conventional mesh-based methods in handling complex dynamical systems with multi-physics coupling. However, when solving parametric DEs, neural operators demand large training datasets consisting of parameter-solution pairs, which still require repeated calls to computationally expensive high-fidelity numerical solvers. Consequently, these methods often underperform in small-data regimes or scenarios where training data are scarce. Furthermore, operator learning approaches demonstrate limited generalization capability and reduced accuracy for certain classes of nonlinear, multi-scale PDEs, particularly for complex problems defined on irregular geometric domains [9–11].

To address these challenges, surrogate modeling techniques have emerged as a powerful approach to reduce the computational cost of predictive modeling for large-scale, complex systems. Surrogate models are computationally efficient approximations of high-fidelity models that can accurately capture system behavior while maintaining controlled accuracy. These data-driven models are constructed based on assumptions about the underlying functional form and utilize system response data for training. Recent advances in machine learning have led to widespread adoption of techniques such as neural network (NN) and Gaussian process (GP) for constructing surrogate models [12]. For instance, Chen et al. [13] developed a data-free surrogate model based on deep neural networks for solving PDEs, while Radaideh et al. [14] integrated surrogate modeling with deep Gaussian processes. To further improve robustness, physics-informed approaches have been incorporated into surrogate modeling frameworks. Pestourie et al. [15] proposed physics-enhanced deep surrogates for PDEs that combines high-fidelity solutions with low-fidelity solutions, achieving higher accuracy with limited training data. Similarly, Li et al. [16] introduced an offline-online computational strategy that combines Markov chain Monte Carlo (MCMC) methods with physics-informed neural networks for Bayesian inverse problems. Recent studies have demonstrated that GP/kernel-based methods can also be competitive. Mora et al. [8] leveraged the strengths of both deep neural networks and kernel methods, proposed an operator learning with Gaussian processes to learn the mapping from input function space to solution space. However, the numerical experiments in this article require a relatively high training cost, and the predictive performance on complex geometries has not been demonstrated. Therefore, addressing problems flexibly under limited sample availability still poses a significant challenge.

In recent years, as researchers address increasingly complex systems, traditional full-order surrogate modeling approaches have exhibited growing computational inefficiencies. Consequently, the integration of reduced basis (RB) methods with surrogate modeling has emerged as a dominant paradigm. RB methods, belonging to the meta-modeling family, are widely employed as surrogates for parameterized large-scale systems [17–20]. The core principle of RB methods is to identify a low-dimensional subspace within the high-dimensional solution space and project the governing equations onto this reduced space, thereby enabling computationally efficient solutions. Among the most prevalent RB techniques is proper orthogonal decomposition (POD) [21, 22], also known as the Karhunen-Lo e expansion or principal component analysis (PCA) in certain contexts. POD operates on a set of solution vectors (snapshots) to derive an optimal basis for a reduced subspace. Its key advantage lies in its ability to truncate the basis optimally, retaining only the most energetically significant modes. Alternative basis construction methods include proper generalized decomposition [23, 24], balanced truncation [25], and rational interpolation [26]. Recently, the combination of RB techniques with data-driven models has shown promise for solving large-scale, complex systems. A major benefit of these hybrid methods is their independence from direct access to or modification of the original high-fidelity model's governing equations. For example, the hybrid POD-neural network approach proposed in [27, 28] employs a neural network to predict the low-dimensional

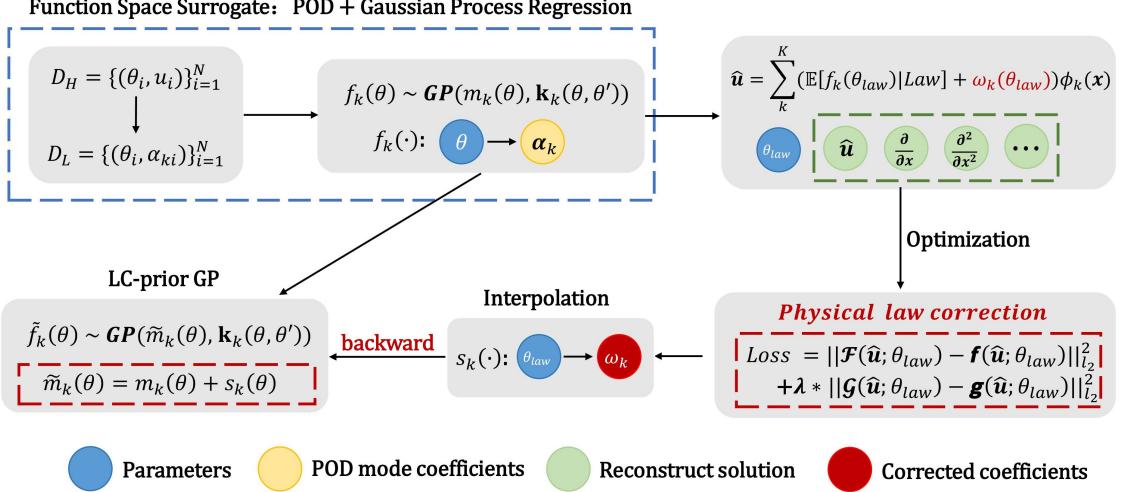


Figure 1: A Schematic of LC-prior GP for differential equations.

projection coefficients of the RB model. Song et al. [29] proposed a model reduction method in which the Koopman operator is extracted from the training set via dynamic mode decomposition (DMD), and the features of a subset of training samples are linearly combined using a nearest-neighbor weighting strategy to enable prediction. As a purely data-driven approach, this method achieves accurate predictions while significantly reducing computational costs. However, a significant limitation of these methods is their reliance on more model evaluations than intrusive approaches to initially construct a reliable surrogate model, particularly for multi-coupled parametric systems.

Inspired by the aforementioned POD method, data-driven Gaussian process surrogate and physics-informed method, we link these three approaches and propose a novel physical law-corrected prior GP (LC-prior GP) as shown in Fig.1. Firstly, we concatenate the training data into a snapshot matrix and apply POD to reduce the infinite-dimensional DEs solution space to a low-dimensional basis and coefficient space that can accurately capture the most important features of the original parametric DEs system. Subsequently, the GP surrogates model are constructed to map parameters in DEs to the coefficients of the reduced basis function, significantly reducing the computational cost associated with directly training the surrogate models in the original DEs solution space. By constructing the snapshot matrix, we aim to fully utilize these known samples and achieve effective prediction of unknown parametric systems through extraction and modeling of their key features - a process entirely driven by data. However, when dealing with limited training samples or extrapolation predictions, the accuracy of such data-driven models faces significant limitations. To address this drawback, we propose a physical law corrected method that leverages governing equations to refine predictions on small testing sets. Specifically, we optimize correction terms between surrogate model predictions and

physics-constrained optimal solutions. These correction terms are then characterized across the entire parameter space using interpolation functions, which are backpropagated to the original GP surrogate. This integration establishes a more physically reasonable prior expectation function that simultaneously satisfies both physical laws and training data constraints for standard data-driven GP surrogate.

To ensure flexible application of our proposed method to irregular computational domains, we employ the radial basis function finite difference (RBF-FD) method for forward simulations when generating training data for the surrogate model. RBF-FD method is a mesh-free numerical discretization method that extended from traditional finite difference methods. In contrast to conventional finite difference methods or finite element methods, RBF-FD does not rely on structured grids. Instead, it constructs differentiation weights through radial basis functions at local stencil, thereby achieving high-precision discretization of PDEs [30–32]. The core concept of RBF-FD lies in leveraging the interpolation properties of radial basis functions to establish local approximation models over arbitrarily distributed nodes within the computational domain. This approach is especially advantageous for irregular geometries, dynamic boundaries, and high-dimensional problems, circumventing the challenges associated with mesh generation in complex domains using traditional grid-based methods. Furthermore, RBF-FD exhibits high accuracy and flexible scalability, enabling the incorporation of various radial basis functions (e.g., Gaussian, multiquadric) to accommodate diverse numerical requirements. Moreover, when solving parametric DEs using RBF-FD method, the evaluation and differentiation matrices characterized by local stencil basis functions depend solely on the equation’s domain definition and nodes positions, remaining independent of system parameters. Leveraging this advantage, our physical law correction of data-driven surrogate models employs RBF-constructed matrices to approximate differential operators governing the equation solutions, significantly reducing both computational errors and time costs associated with numerical differentiation methods.

In this framework, when addressing parameter estimation inverse problems, the surrogate model eliminates the need for multiple queries to expensive numerical solvers. Instead, it uses the existing surrogate model to predict the coefficients of the reduced basis, which can then be employed to reconstruct solutions for DEs with different parameters and compute the likelihood function. We apply the Markov Chain Monte Carlo (MCMC) method to obtain posterior samples of the parameters from the likelihood function, facilitating uncertainty quantification (UQ) and measurement of the unknown parameters. Our approach integrates data-driven machine learning techniques with physical laws within the RB framework, ensuring computational efficiency in low-dimensional spaces while maintaining the surrogate model’s accuracy with small sample sizes.

The main contributions of this paper are summarized as follows:

- We propose a novel approach that combines proper orthogonal decomposition and GP surrogate for the construction of infinite-dimensional function surrogate models. By fully utilizing the available data in both parametric representation and the surrogate model con-

struction, this method significantly enhances the approximation accuracy of the function.

- For the standard GP surrogate model, we propose a physical law correction for the prior expectation function. This model enables joint modeling of multiple physical variables while satisfying the constraints of the law in the context of DEs problems, that improving the prediction capability of data-driven surrogates and enhancing their generalizability in small sample settings.
- During the computation of the training set, we choose the meshless RBF-FD. Its combination with the POD method enables flexible handling of irregular geometry. Additionally, since the evaluation and differentiation matrices are represented by radial basis functions, repeated solving is unnecessary in the physical law correction part, greatly improving computational efficiency and accuracy during optimization.
- Building on the LC-prior GP, we infer the posterior distribution of unknown parameters from noisy observational data within a Bayesian framework, enabling effective uncertainty quantification. The posterior samples obtained through our method provide more accurate estimates of the parameters.

The structure of the paper is organized as follows: Section.2 formulates the problem setup and surrogate model construction based on parametric DEs system. In Section.3, we introduce proper orthogonal decomposition for parameterizing infinite-dimensional solutions and propose a physical law corrected prior function to improve the pure data-driven GP performance. We also briefly explains the parameter estimation process under LC-prior GP. Section.4 presents numerical examples, ranging from single-physical-quantity cases to multi-dimensional parameters with coupled multiple physical quantities to demonstrate the effectiveness of proposed method. Section.4 concludes with remarks and a discussion of future directions.

2 Problem setting in parametric PDEs

Parametric DEs are fundamental tools for modeling a wide range of phenomena in science and engineering. They describe how quantities change over time or space and are essential for understanding the underlying mechanisms of nature. Let Ω be the computational domain with the Lipschitz continuous boundary $\partial\Omega$. The general form for differential equation with parameters can be expressed as:

$$\begin{cases} \mathcal{F}(\mathbf{u}(x; \theta), \nabla \mathbf{u}, \nabla^2 \mathbf{u}, \dots; \theta) = \mathbf{f}(x; \theta), & x \in \Omega, \\ \mathcal{G}(\mathbf{u}(x; \theta), \nabla \mathbf{u}, \nabla^2 \mathbf{u}, \dots; \theta) = \mathbf{g}(x; \theta), & x \in \partial\Omega. \end{cases} \quad (1)$$

where $\mathbf{u}(x; \theta)$ is the solution vector and $\theta = (\theta_1, \dots, \theta_q)^T \in \mathbb{R}^q$ is the vector of parameters. The operator function \mathcal{F} and \mathcal{G} are represents a linear or nonlinear function involving \mathbf{u} and differential operator acting on \mathbf{u} defined in computational domain Ω and boundary

$\partial\Omega$ respectively. $\mathbf{f}(\mathbf{x}; \theta)$ and $\mathbf{g}(\mathbf{x}; \theta)$ are source term functions. In the following, we denote $\mathcal{F}(\mathbf{u}, \nabla\mathbf{u}, \nabla^2\mathbf{u}, \dots; \theta)$ by the short-hand notation $\mathcal{F}(\mathbf{u}; \theta)$. We can define an operator from θ to infinity-dimensional solution space of the DEs by :

$$\mathcal{M} : \theta \in \mathbb{R}^q \mapsto \mathbf{u}(\mathbf{x}; \theta).$$

2.1 RBF-FD method

Radial basis function (RBF) methods represent a class of mesh-free techniques that can be classified into several categories, including local RBF methods, global RBF methods, RBF-FD methods and RBF spectral methods. This study focuses specifically on the local RBF-FD approach integrated with the least squares technique. The local least squares RBF-FD method enables spatial discretization of PDEs without requiring a predefined mesh structure. This approach offers significant flexibility in handling complex geometries and is particularly effective for solving problems in arbitrarily shaped domains. The methodological details of the local RBF-FD scheme and its application to spatial discretization of parametric PDEs will be systematically presented in later sections.

We begin by reviewing fundamental concepts of RBF interpolation, which form the basis for the RBF-FD methodology. Consider a set of scattered nodes $\mathbf{x}_i \in \mathbb{R}^d$, $i = 1, 2, \dots, n$ distributed in the neighborhood of a point \mathbf{x} , where these nodes are completely independent of any mesh or element structure. A localized RBF approximation of the function $u(\mathbf{x})$ can be constructed using the radial basis functions $\phi(\|\mathbf{x} - \mathbf{x}_i\|)$,

$$u_h(\mathbf{x}) = \sum_{i=1}^n c_i \phi(\|\mathbf{x} - \mathbf{x}_i\|),$$

where $\phi(\|\cdot\|)$ is some radial function, $\|\cdot\|$ is the standard Euclidean norm, and c_i are unknown coefficients. Using the interpolation condition $u_h(\mathbf{x}_i) = u(\mathbf{x}_i)$, we can obtain a linear system as:

$$\underbrace{\begin{pmatrix} \phi(\|\mathbf{x}_1 - \mathbf{x}_1\|) & \cdots & \phi(\|\mathbf{x}_1 - \mathbf{x}_n\|) \\ \vdots & \ddots & \vdots \\ \phi(\|\mathbf{x}_n - \mathbf{x}_1\|) & \cdots & \phi(\|\mathbf{x}_n - \mathbf{x}_n\|) \end{pmatrix}}_A \underbrace{\begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix}}_{\mathbf{c}} = \underbrace{\begin{pmatrix} u(\mathbf{x}_1) \\ \vdots \\ u(\mathbf{x}_n) \end{pmatrix}}_{\mathbf{u}}.$$

Let $\mathbf{c} = (c_1, \dots, c_n)^T$ and $\mathbf{u} = (u(\mathbf{x}_1), \dots, u(\mathbf{x}_n))^T$, we have the compact form $A\mathbf{c} = \mathbf{u}$.

To avoid the parameter optimization, we choose the piecewise smooth polyharmonic splines (PHS) for simplicity. Since PHS-RBF is a conditionally positive definite function, the interpolation using pure RBFs cannot guarantee convergence and solvability. By augmenting the approximation with a polynomial basis whose degree matches the order of conditional positive definiteness and enforcing orthogonality between the RBF coefficients \mathbf{c} and this polynomial

basis, we ensure that the quadratic form $\mathbf{c}^T A \mathbf{c}$ becomes strictly positive definite. This property is crucial for establishing optimality results. The resulting RBF interpolation takes the form:

$$\begin{cases} u_h(\mathbf{x}) = \sum_{i=1}^n c_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) + \sum_{k=1}^m \beta_k p_k(\mathbf{x}), \\ \sum_{i=1}^n c_i p_k(\mathbf{x}_i) = 0, \quad k = 1, \dots, m \end{cases} \quad (2)$$

The dimension m of the polynomial space is given by $m = \binom{D_m+d}{d}$, where D_m is the degree of the polynomial and d is the spatial dimension of \mathbb{R}^d . The coefficients c_i and β_k are determined by the collocation method and the additional constraints. The numerical optimal solution can be expressed as the solution of the following linear system:

$$\left(\begin{array}{ccc|ccc} \phi(\|\mathbf{x}_1 - \mathbf{x}_1\|) & \cdots & \phi(\|\mathbf{x}_1 - \mathbf{x}_n\|) & p_1(\mathbf{x}_1) & \cdots & p_m(\mathbf{x}_1) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \phi(\|\mathbf{x}_n - \mathbf{x}_1\|) & \cdots & \phi(\|\mathbf{x}_n - \mathbf{x}_n\|) & p_1(\mathbf{x}_n) & \cdots & p_m(\mathbf{x}_n) \\ \hline p_1(\mathbf{x}_1) & \cdots & p_1(\mathbf{x}_n) & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ p_m(\mathbf{x}_1) & \cdots & p_m(\mathbf{x}_n) & 0 & \cdots & 0 \end{array} \right) \begin{pmatrix} c_1 \\ \vdots \\ c_n \\ \hline \beta_1 \\ \vdots \\ \beta_m \end{pmatrix} = \begin{pmatrix} u(\mathbf{x}_1) \\ \vdots \\ u(\mathbf{x}_n) \\ \hline 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Let $\beta = (\beta_1, \dots, \beta_m)^T$, we also have the compact form as $\hat{A}\hat{\mathbf{c}} = \hat{\mathbf{u}}$:

$$\underbrace{\begin{pmatrix} A & P \\ P^T & \mathbf{0} \end{pmatrix}}_{\hat{A}} \underbrace{\begin{pmatrix} \mathbf{c} \\ \beta \end{pmatrix}}_{\hat{\mathbf{c}}} = \underbrace{\begin{pmatrix} \mathbf{u} \\ 0 \end{pmatrix}}_{\hat{\mathbf{u}}}.$$

In order to digitally discretize the problem Eq.(1), two sets of computational points are distributed over computational domain Ω :

- The interpolation point set $Y = \{\mathbf{y}_i\}_{i=1}^N$ for generating the cardinal functions.
- The evaluation point set $X = \{\mathbf{x}_j\}_{j=1}^M$ for sampling the PDE, and $M = qN$.

In the dataset Y , each data node \mathbf{y}_i corresponds to a local support domain which contributes to the stencil $Y_s = \{\mathbf{y}_i^s\}_{i=1}^n$.

$$u_h^s(\mathbf{y}) = \sum_{i=1}^n c_i \phi(\|\mathbf{y} - \mathbf{y}_i^s\|) + \sum_{k=1}^m \beta_k p_k(\mathbf{y}), \quad \text{if } \mathbf{y} \in X_s$$

To evaluate the RBF-FD approximation at a point \mathbf{x} , we choose the stencil associated with the point \mathbf{y}_i that is closest to \mathbf{x} . That is,

$$s(\mathbf{x}) = \arg \min_i \|\mathbf{x} - \mathbf{y}_i\|. \quad (3)$$

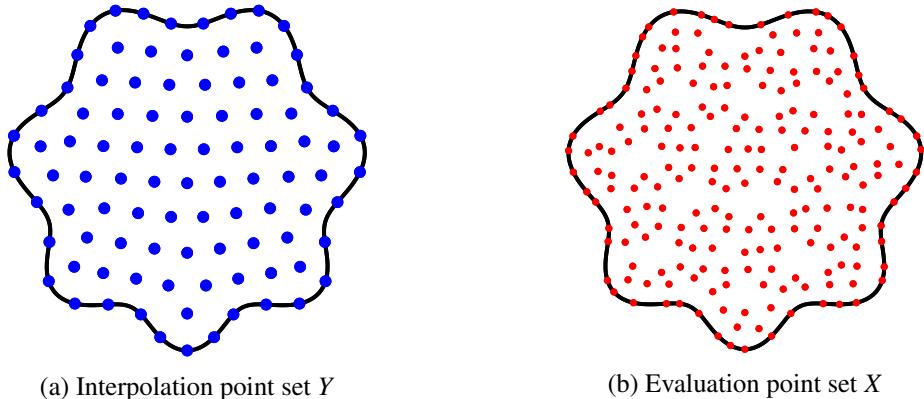


Figure 2: The interpolation point set Y and evaluation point set X .

Any point $\mathbf{x} \in \Omega$ is uniquely with one stencil Y_s by Eq.(3). Then the local interpolation for the solution of the PDE problem evaluated at the point \mathbf{x} with local stencil can be written as:

$$\begin{aligned}
u_h^s(\mathbf{x}) &= \sum_{i=1}^n c_i^s \phi(\|\mathbf{x} - \mathbf{y}_i^s\|) + \sum_{k=1}^m \beta_k^s p_k(\mathbf{x}) \\
&= \left(\phi(\|\mathbf{x} - \mathbf{y}_1^s\|), \dots, \phi(\|\mathbf{x} - \mathbf{y}_n^s\|), p_1(\mathbf{x}), \dots, p_m(\mathbf{x}) \right) \begin{pmatrix} \mathbf{c}^{(s)} \\ \boldsymbol{\beta}^{(s)} \end{pmatrix} \\
&= \left(\phi(\|\mathbf{x} - \mathbf{y}_1^s\|), \dots, \phi(\|\mathbf{x} - \mathbf{y}_n^s\|), p_1(\mathbf{x}), \dots, p_m(\mathbf{x}) \right) \begin{pmatrix} \mathbf{A}^{(s)} & \mathbf{P}^{(s)} \\ (\mathbf{P}^{(s)})^T & \mathbf{0} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{u}_h^{(s)} \\ \mathbf{0} \end{pmatrix} \\
&= \left(\Phi_1^s(\mathbf{x}), \Phi_2^s(\mathbf{x}), \dots, \Phi_n^s(\mathbf{x}), \delta_1^s(\mathbf{x}), \delta_2^s(\mathbf{x}), \dots, \delta_m^s(\mathbf{x}) \right) \begin{pmatrix} \mathbf{u}_h^{(s)} \\ \mathbf{0} \end{pmatrix} \\
&= \sum_{i=1}^n \Phi_i^s(\mathbf{x}) u_h(\mathbf{y}_i^s), \quad \text{where } \mathbf{y}_i^s \in Y_s
\end{aligned}$$

where the function $\Phi_i^s(\cdot)$ and $\delta_k^s(\cdot)$ can be written as:

$$\begin{aligned}
\Phi_i^s(\mathbf{x}) &= \phi(\|\mathbf{x} - \mathbf{y}_i^s\|) \begin{pmatrix} \mathbf{A}^{(s)} & \mathbf{P}^{(s)} \\ (\mathbf{P}^{(s)})^T & \mathbf{0} \end{pmatrix}^{-1}, \quad i = 1, \dots, n \\
\delta_k^s(\mathbf{x}) &= p_k(\mathbf{x}) \begin{pmatrix} \mathbf{A}^{(s)} & \mathbf{P}^{(s)} \\ (\mathbf{P}^{(s)})^T & \mathbf{0} \end{pmatrix}^{-1}, \quad k = 1, \dots, m
\end{aligned}$$

For ease of analysis, we define the global function of the whole computational domain:

$$\Phi_i(\mathbf{x}) = \begin{cases} \Phi_i^s(\mathbf{x}), & \mathbf{y}_i \in Y_s \\ 0 & \mathbf{y}_i \notin Y_s \end{cases}.$$

For any point from evaluation point set X , we can use the global basis function $\Phi_i(\cdot)$ and Eq.(3) to construct the interpolation function:

$$u_h(\mathbf{x}) = \sum_{i=1}^N \Phi_i(\mathbf{x}) u_h(\mathbf{y}_i)$$

We construct a sparse global linear system:

$$\underbrace{\begin{pmatrix} \Phi_1(\mathbf{x}_1) & \cdots & \Phi_N(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \Phi_1(\mathbf{x}_M) & \cdots & \Phi_N(\mathbf{x}_M) \end{pmatrix}}_{E_h(X, Y)} \underbrace{\begin{pmatrix} u_h(\mathbf{y}_1) \\ \vdots \\ u_h(\mathbf{y}_N) \end{pmatrix}}_{u_h(Y)} = \underbrace{\begin{pmatrix} u_h(\mathbf{x}_1) \\ \vdots \\ u_h(\mathbf{x}_M) \end{pmatrix}}_{u_h(X)}.$$

And we have the compact form: $u_h(X) = E_h(X, Y)u_h(Y)$. The expression for the action of a differential operator \mathcal{L} on the RBF approximation $u_h(\mathbf{x})$ as follow:

$$\mathcal{L}u_h(\mathbf{x}) = \sum_{i=1}^N \mathcal{L}\Phi_i(\mathbf{x}) u_h(\mathbf{y}_i),$$

In the same way of the global sampling set X :

$$\mathcal{L}u_h(X) = D_h^{\mathcal{L}}(X, Y)u_h(Y).$$

The evaluation matrix $E_h(X, Y)$ and differentiation matrix $D_h^{\mathcal{L}}(X, Y)$ are both $M \times N$ sparse matrices. Using the radial basis function method for the spatial discretization of the Eq.(1) system, we can obtain:

$$\begin{cases} \mathcal{F}(E_h(X, Y)u_h(Y), D_h^{\nabla}(X, Y)u_h(Y), D_h^{\nabla^2}(X, Y)u_h(Y), \dots; \theta) = \mathbf{f}(X; \theta), & X \in \Omega \\ \mathcal{G}(E_h(X, Y)u_h(Y), D_h^{\nabla}(X, Y)u_h(Y), D_h^{\nabla^2}(X, Y)u_h(Y), \dots; \theta) = \mathbf{g}(X; \theta), & X \in \partial\Omega \end{cases}$$

In order to solve the solution $u_h(Y)$, we can construct a linear system and use least squares method. For the sake of intuitive expression, the boundary points are not distinguished here:

$$\underbrace{\begin{pmatrix} \mathcal{F}(\Phi_1(\mathbf{x}_1), \nabla\Phi_1(\mathbf{x}_1), \dots; \theta) & \cdots & \mathcal{F}(\Phi_N(\mathbf{x}_1), \nabla\Phi_N(\mathbf{x}_1), \dots; \theta) \\ \mathcal{F}(\Phi_1(\mathbf{x}_2), \nabla\Phi_1(\mathbf{x}_2), \dots; \theta) & \cdots & \mathcal{F}(\Phi_N(\mathbf{x}_2), \nabla\Phi_N(\mathbf{x}_2), \dots; \theta) \\ \vdots & \ddots & \vdots \\ \mathcal{F}(\Phi_1(\mathbf{x}_M), \nabla\Phi_1(\mathbf{x}_M), \dots; \theta) & \cdots & \mathcal{F}(\Phi_N(\mathbf{x}_M), \nabla\Phi_N(\mathbf{x}_M), \dots; \theta) \end{pmatrix}}_{M \times N} \underbrace{\begin{pmatrix} u_h(\mathbf{y}_1) \\ u_h(\mathbf{y}_2) \\ \vdots \\ u_h(\mathbf{y}_N) \end{pmatrix}}_{N \times 1} = \underbrace{\begin{pmatrix} \mathbf{f}(\mathbf{x}_1; \theta) \\ \mathbf{f}(\mathbf{x}_2; \theta) \\ \vdots \\ \mathbf{f}(\mathbf{x}_M; \theta) \end{pmatrix}}_{M \times 1}$$

After calculating the interpolation point domain $u_h(Y)$, the evaluation point domain can be calculated by evaluation matrix $E_h(X, Y)$: $\mathbf{u}(X) = E_h(X, Y)u_h(Y)$.

The RBF-FD method is equally applicable for solving time-dependent dynamic PDEs. To illustrate this capability, we examine the following generalized temporal equation form:

$$\begin{cases} \frac{\partial}{\partial t} \mathbf{u}(\mathbf{x}, t; \boldsymbol{\theta}) = \mathcal{F}(\mathbf{u}, \nabla \mathbf{u}, \nabla^2 \mathbf{u}, \dots; \boldsymbol{\theta}), & \mathbf{x} \in \Omega \times (0, T], \\ \mathbf{u}(\mathbf{x}, 0; \boldsymbol{\theta}) = \mathbf{u}_0(\mathbf{x}; \boldsymbol{\theta}), \end{cases} \quad (4)$$

The construction procedures for radial basis function, evaluation and differentiation matrices remain rigorously consistent with the aforementioned methodology. The essential distinction manifests primarily in the matrix assembly strategy of the linear solving system:

$$\frac{\partial}{\partial t} E_h(X, Y) \mathbf{u}(Y, t_{n+1}; \boldsymbol{\theta}) = \mathcal{F}(E_h(X, Y) u_h(Y, t_n), D_h^\nabla(X, Y) u_h(Y, t_n), D_h^{\nabla^2}(X, Y) u_h(Y, t_n), \dots; \boldsymbol{\theta})$$

same as the matrix linear system:

$$\underbrace{\begin{pmatrix} \Phi_1(\mathbf{x}_1) & \cdots & \Phi_N(\mathbf{x}_1) \\ \Phi_1(\mathbf{x}_2) & \cdots & \Phi_N(\mathbf{x}_2) \\ \vdots & \ddots & \vdots \\ \Phi_1(\mathbf{x}_M) & \cdots & \Phi_N(\mathbf{x}_M) \end{pmatrix}}_{M \times N} \underbrace{\begin{pmatrix} (\mathbf{u}(\mathbf{y}_1, t_{n+1}) - \mathbf{u}(\mathbf{y}_1, t_n)) / \Delta t \\ (\mathbf{u}(\mathbf{y}_2, t_{n+1}) - \mathbf{u}(\mathbf{y}_2, t_n)) / \Delta t \\ \vdots \\ (\mathbf{u}(\mathbf{y}_N, t_{n+1}) - \mathbf{u}(\mathbf{y}_N, t_n)) / \Delta t \end{pmatrix}}_{N \times 1} = \underbrace{\begin{pmatrix} \mathcal{F}(\mathbf{u}(\mathbf{x}_1, t_n), \nabla \mathbf{u}(\mathbf{x}_1, t_n), \dots; \boldsymbol{\theta}) \\ \mathcal{F}(\mathbf{u}(\mathbf{x}_2, t_n), \nabla \mathbf{u}(\mathbf{x}_2, t_n), \dots; \boldsymbol{\theta}) \\ \vdots \\ \mathcal{F}(\mathbf{u}(\mathbf{x}_M, t_n), \nabla \mathbf{u}(\mathbf{x}_M, t_n), \dots; \boldsymbol{\theta}) \end{pmatrix}}_{M \times 1}.$$

where Δt is the iscretization step size in temporal domain $[0, T]$.

Through temporal discretization and the construction of the aforementioned linear system, the RBF-FD method can effectively transform complex dynamic PDE into a system of ODE in the temporal domain, which is then solved iteratively. By calculating the interpolation point domain $\mathbf{u}(Y, t_{n+1}; \boldsymbol{\theta})$, the evaluation point domain at the same time t_{n+1} can be calculated: $\mathbf{u}(X, t_{n+1}; \boldsymbol{\theta}) = E_h(X, Y) \mathbf{u}(Y, t_{n+1}; \boldsymbol{\theta})$.

In various practical applications, such as parameter estimation, one often encounters a multitude of equations $\mathbf{u}(\mathbf{x}; \boldsymbol{\theta})$ characterized by distinct unknown parameters $\boldsymbol{\theta}$. Once we have a parameter, we can calculate its solutions by $\mathcal{M}_{\text{RBF-FD}} : \boldsymbol{\theta} \mapsto \mathbf{u}(\mathbf{x}; \boldsymbol{\theta})$, but conventional methods necessitate numerous queries to costly numerical solutions of DEs, leading to significant time expenditures. Consequently, the development of a low-cost, high-accuracy surrogate model becomes essential.

2.2 Surrogate model

Surrogate models serve as computationally efficient approximations of complex, resource-intensive simulations, finding widespread applications in scientific computing, engineering optimization, and machine learning. These data-driven approximations, constructed using techniques such as Gaussian process regression, support vector regression, or neural networks, establish input-output relationships to enable rapid predictions for parameter sweeps, optimization, and uncertainty quantification.

Consider the scenario where direct numerical solutions of differential equations via $\mathcal{M}_{\text{RBF-FD}}$ in finite element analysis or computational fluid dynamics prove prohibitively expensive. A well-constructed surrogate model \mathcal{M}_{sur} can provide approximate solutions with dramatically reduced computational overhead while maintaining acceptable accuracy.

In constructing a surrogate, we assume the availability of training data $\mathcal{D} = \{(\theta_i, \mathbf{u}(\mathbf{x}; \theta_i))\}_{i=1}^N$. The goal is to learn a surrogate map:

$$\mathcal{M}_{\text{sur}} : \theta \mapsto \mathbf{u}(\mathbf{x}; \theta),$$

where $\theta = (\theta_1, \dots, \theta_q)^T \in \mathbb{R}^q$. The surrogate \mathcal{M}_{sur} proves particularly valuable when full-scale simulations become computationally intractable.

3 Law-corrected surrogate modeling with Gaussian process regression

We propose an infinite-dimensional surrogate model that maps the parameters θ to the solution function $\mathbf{u}(\mathbf{x}; \theta)$ within a Gaussian process regression framework, employing a novel law-corrected prior. Our approach primarily relies on the following key components: 1. Parameterizing the solution function $\mathbf{u}(\mathbf{x}; \theta)$ by sample features using POD; 2. Constructing finite regression models from θ to $\mathbf{u}(\mathbf{x}; \theta)$ using standard GP surrogate for each target solution function; 3. Incorporating physical laws to joint optimization of the prediction of entire physics system and learning the conditional prior mean function for each original surrogate; 4. Applying our surrogate model to parameter estimation scenarios within a Bayesian framework.

3.1 Parametric representation of the solution for PDEs

Basis function expansion is a widely used technique for representing complex functions across fields such as numerical analysis, signal processing, and differential equation solving [33]. By linearly combining basis functions, various complex functions can be approximated or reconstructed.

We construct the approximate DE solution, denoted as $\hat{\mathbf{u}}(\mathbf{x}; \theta)$, using K orthogonal basis functions:

$$\mathbf{u}(\mathbf{x}; \theta) \approx \hat{\mathbf{u}}(\mathbf{x}; \theta) = \sum_{k=1}^K \alpha_k(\theta) \phi_k(\mathbf{x}),$$

where $\phi_k(\mathbf{x})$ is the basis function and $\alpha_k(\theta)$ is the coefficient of corresponding orthogonal basis. Once the type of basis function and truncated value K is determined, the function $\mathbf{u}(\mathbf{x}; \theta)$ is completely represented by K coefficients. Different types of basis functions are suitable for different applications, and choosing the appropriate basis can significantly enhance computational efficiency and accuracy.

3.1.1 Proper orthogonal decomposition

In addressing complex physics problems, conventional basis function approaches often fail to adequately capture the essential features of DEs, particularly when employing a restricted number of basis functions to achieve dimensionality reduction objectives. To address this fundamental challenge, we employ Proper Orthogonal Decomposition (POD), a powerful technique that has found widespread application in processing high-dimensional computational data to extract low-dimensional representations of dominant physical phenomena [34]. As a numerical dimensionality reduction technique, POD significantly reduces computational complexity in intensive simulations such as computational fluid dynamics and structural analysis [35]. The method distinguishes itself from traditional basis function approaches by eliminating the need for a priori assumptions about basis function forms. Instead, POD derives optimal basis functions directly from system data through singular value decomposition of the snapshot matrix constructed from training data. This data-driven approach yields superior dimensionality reduction performance while maintaining broad applicability across diverse physical systems. The mathematical framework of POD shares conceptual similarities with principal component analysis in machine learning, further enhancing its interdisciplinary utility.

Suppose a set of training data $\mathcal{D} = \{(\boldsymbol{\theta}_n, \mathbf{u}(\mathbf{x}; \boldsymbol{\theta}_n))\}_{n=1}^N$, which contains parameters $\boldsymbol{\theta} \in \mathbb{R}^q$ and $\mathbf{x} = (x_1, \dots, x_D)^T \in \mathbb{R}^D$ represents the discretized spatial domain. By discretizing the solutions across D spatial points for each parameter $\boldsymbol{\theta}_n$, we construct an $N \times D$ snapshot matrix \mathbf{U} by:

$$\mathbf{U} = \begin{pmatrix} \mathbf{u}(\mathbf{x}; \boldsymbol{\theta}_1) \\ \vdots \\ \mathbf{u}(\mathbf{x}; \boldsymbol{\theta}_N) \end{pmatrix} = \begin{pmatrix} \mathbf{u}(x_1; \boldsymbol{\theta}_1), & \dots, & \mathbf{u}(x_D; \boldsymbol{\theta}_1) \\ \vdots & \ddots & \vdots \\ \mathbf{u}(x_1; \boldsymbol{\theta}_N), & \dots, & \mathbf{u}(x_D; \boldsymbol{\theta}_N), \end{pmatrix}$$

where each row of \mathbf{U} represents the discrete solution $\mathbf{u}(\mathbf{x}; \boldsymbol{\theta}_n)$ discretized across the spatial domain for a given parameter $\boldsymbol{\theta}_n$ and we compute the covariance matrix of snapshot matrix as:

$$\mathbf{C} = \frac{1}{N-1} \mathbf{U}^T \mathbf{U}.$$

We perform eigenvalue decomposition on the \mathbf{C} by $\mathbf{C}\phi_i = \lambda_i\phi_i$, $i = 1, \dots, D$ to obtain eigenvalues λ and eigenvectors ϕ . The eigenvectors ϕ_i represent the POD modes, and their corresponding eigenvalues λ_i quantify the relative energy contribution of each mode to the overall system. The larger the value, the greater contribution of this mode to the snapshot matrix. So we sort them in descending order according to the size of the eigenvalues and select the top-ranked K eigenvectors as POD basis function to approximate solution as:

$$\underbrace{\begin{pmatrix} \alpha_1(\boldsymbol{\theta}_1) & \dots & \alpha_K(\boldsymbol{\theta}_1) \\ \alpha_1(\boldsymbol{\theta}_2) & \dots & \alpha_K(\boldsymbol{\theta}_2) \\ \vdots & \ddots & \vdots \\ \alpha_1(\boldsymbol{\theta}_N) & \dots & \alpha_K(\boldsymbol{\theta}_N) \end{pmatrix}}_{N \times K} \underbrace{\begin{pmatrix} \phi_1(x_1) & \dots & \phi_1(x_D) \\ \phi_2(x_1) & \dots & \phi_2(x_D) \\ \vdots & \ddots & \vdots \\ \phi_K(x_1) & \dots & \phi_K(x_D) \end{pmatrix}}_{K \times D} = \underbrace{\begin{pmatrix} \mathbf{u}(x_1; \boldsymbol{\theta}_1), & \dots, & \mathbf{u}(x_D; \boldsymbol{\theta}_1) \\ \mathbf{u}(x_1; \boldsymbol{\theta}_2), & \dots, & \mathbf{u}(x_D; \boldsymbol{\theta}_2) \\ \vdots & \ddots & \vdots \\ \mathbf{u}(x_1; \boldsymbol{\theta}_N), & \dots, & \mathbf{u}(x_D; \boldsymbol{\theta}_N) \end{pmatrix}}_{N \times D}$$

We can have the compact form as: $\alpha \phi_K = \mathbf{U}$.

After computing the coefficient matrix, we can effectively approximate the original snapshot matrix through a linear combination of the POD basis modes and their corresponding coefficients by:

$$\mathbf{U} = \begin{pmatrix} \mathbf{u}(\mathbf{x}; \boldsymbol{\theta}_1) \\ \mathbf{u}(\mathbf{x}; \boldsymbol{\theta}_2) \\ \vdots \\ \mathbf{u}(\mathbf{x}; \boldsymbol{\theta}_N) \end{pmatrix} \approx \begin{pmatrix} \sum_{k=1}^K \alpha_k(\boldsymbol{\theta}_1) \phi_k(\mathbf{x}) \\ \sum_{k=1}^K \alpha_k(\boldsymbol{\theta}_2) \phi_k(\mathbf{x}) \\ \vdots \\ \sum_{k=1}^K \alpha_k(\boldsymbol{\theta}_N) \phi_k(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \sum_{k=1}^K \alpha_k(\boldsymbol{\theta}_1) \phi_k(x_1) & \cdots & \sum_{k=1}^K \alpha_k(\boldsymbol{\theta}_1) \phi_k(x_D) \\ \sum_{k=1}^K \alpha_k(\boldsymbol{\theta}_2) \phi_k(x_1) & \cdots & \sum_{k=1}^K \alpha_k(\boldsymbol{\theta}_2) \phi_k(x_D) \\ \vdots & \ddots & \vdots \\ \sum_{k=1}^K \alpha_k(\boldsymbol{\theta}_N) \phi_k(x_1) & \cdots & \sum_{k=1}^K \alpha_k(\boldsymbol{\theta}_N) \phi_k(x_D) \end{pmatrix}.$$

where α are the coefficients representing the projection onto the K -th basis function which need be computed. Through POD, we map the D -dimensional solution space to a K -dimensional space, where $K \ll D$, effectively achieving dimensionality reduction while ensuring the accuracy of the approximate solution. By fixing the modes ϕ_k as basis function, we can reconstruct the solution corresponding to any parameter sample $\boldsymbol{\theta}$ by learning surrogate model for the corresponding coefficients $\alpha(\boldsymbol{\theta})$. The limitation of the POD method lies in the need for data to construct the snapshot matrix. However, this issue can be effectively addressed within the framework of machine learning.

3.2 GP surrogate model

The GPR is a widely used technique for surrogate modeling, offering a nonparametric perspective. Suppose we have a dataset obtained by numerical method :

$$\mathcal{D}_{\text{High}} = \{(\boldsymbol{\theta}_i, \mathbf{u}(\mathbf{x}; \boldsymbol{\theta}_i))\}_{i=1}^N,$$

where $\boldsymbol{\theta}_i \in \mathbb{R}^q$, $\mathbf{u}(\mathbf{x}; \boldsymbol{\theta}_i) \in \mathbb{R}^D$ and N is the size of the dataset. The training dataset is denoted as $\mathcal{D}_{\text{High}}$, as the input $\boldsymbol{\theta}$ corresponds to high-dimensional equation solutions as outputs. To reduce dimensionality, we map the discrete space of D to a basis function coefficient space of a lower dimension using RB technology:

$$\text{POD} : \mathbf{u}(\mathbf{x}; \boldsymbol{\theta}_i) \in \mathbb{R}^D \mapsto \alpha(\boldsymbol{\theta}_i) \in \mathbb{R}^K, \quad K \ll D,$$

where $\alpha(\boldsymbol{\theta}_i) = (\alpha_1(\boldsymbol{\theta}_i), \dots, \alpha_K(\boldsymbol{\theta}_i))^T$ denoted by α_i and K is the number of basis functions. Thereby a low-dimensional training dataset is obtained by taking the basis function coefficients as the corresponding outputs, denoted as \mathcal{D}_{Low} :

$$\mathcal{D}_{\text{Low}} = \{(\boldsymbol{\theta}_i, \alpha_i)\}_{i=1}^N,$$

Our goal is to learn the function $f_k(\cdot) : \boldsymbol{\theta} \mapsto \alpha_k$ by constructing K individual GPR models, each sharing the same input $\boldsymbol{\theta}$ but with different outputs α_k :

$$f_k(\boldsymbol{\theta}) \sim \mathcal{GP}(m_k(\boldsymbol{\theta}), \mathbf{k}_k(\boldsymbol{\theta}, \boldsymbol{\theta}')),$$

where $m_k(\boldsymbol{\theta})$ is the mean function, often assumed to be a constant 0, and $\mathbf{k}_k(\boldsymbol{\theta}, \boldsymbol{\theta}')$ is the kernel function. A commonly used kernel is the Radial Basis Function (RBF) kernel, expressed as:

$$\mathbf{k}_{\text{RBF}} = \gamma^2 \exp\left(-\frac{\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j\|^2}{2\ell^2}\right).$$

According to the definition of GP, the finite projection of $f_k(\cdot)$ onto the training inputs $\boldsymbol{\theta}$, namely $\mathbf{f}_k = (f_k(\boldsymbol{\theta}_1), \dots, f_k(\boldsymbol{\theta}_N))^T$, follows a multivariate Gaussian distribution,

$$p(\mathbf{f}_k | \boldsymbol{\theta}) = \mathcal{N}(\mathbf{f}_k | \mathbf{0}, \mathbf{K}_k),$$

where \mathbf{K}_k represents the kernel matrix evaluated at $\boldsymbol{\theta}$, and each element is defined as $[\mathbf{K}_k]_{i,j} = \mathbf{k}_k(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$. Let $\boldsymbol{\eta}_k = (\gamma_k, \ell_k)$ denote the set of all hyper-parameters associated with \mathbf{K}_k . By integrating out \mathbf{f}_k , the marginal likelihood can be obtained. To learn the model, we maximize the log-likelihood to estimate the kernel parameters $\boldsymbol{\eta}_k$ for each α_k :

$$\log p(\alpha_k | \boldsymbol{\theta}, \boldsymbol{\eta}_k) = -\frac{1}{2} \alpha_k^T \mathbf{K}_k(\boldsymbol{\theta}, \boldsymbol{\theta})^{-1} \alpha_k - \frac{1}{2} \log \det \mathbf{K}_k(\boldsymbol{\theta}, \boldsymbol{\theta}) - \frac{N}{2} \log 2\pi.$$

According to the GP prior, given a new input $\boldsymbol{\theta}^*$, the posterior (or predictive) distribution of the output $f(\boldsymbol{\theta}^*)$ is a conditional Gaussian distribution:

$$p(f_k(\boldsymbol{\theta}^*) | \boldsymbol{\theta}^*, \mathcal{D}_{\text{Low}}) = \mathcal{N}(f_k(\boldsymbol{\theta}^*) | \mu_k(\boldsymbol{\theta}^*), \sigma_k^2(\boldsymbol{\theta}^*)),$$

where the posterior mean and variance are given by:

$$\begin{aligned} \mu_k(\boldsymbol{\theta}^*) &= \mathbb{E}[f_k(\boldsymbol{\theta}^*) | \boldsymbol{\theta}^*, \mathcal{D}_{\text{Low}}] = m_k(\boldsymbol{\theta}^*) + \mathbf{k}_{k*}^T \mathbf{K}_k^{-1} (\alpha_k - m_k(\boldsymbol{\theta})) \\ \sigma_k^2(\boldsymbol{\theta}^*) &= \text{Var}[f_k(\boldsymbol{\theta}^*) | \boldsymbol{\theta}^*, \mathcal{D}_{\text{Low}}] = \mathbf{k}_k(\boldsymbol{\theta}^*, \boldsymbol{\theta}^*) - \mathbf{k}_{k*}^T \mathbf{K}_k^{-1} \mathbf{k}_{k*} \end{aligned}$$

where $\mathbf{k}_{k*} = (\mathbf{k}_k(\boldsymbol{\theta}^*, \boldsymbol{\theta}_1), \dots, \mathbf{k}_k(\boldsymbol{\theta}^*, \boldsymbol{\theta}_N))^T$ represents the kernel evaluations between $\boldsymbol{\theta}^*$ and the training input.

By iterating this process, we independently learn the surrogate model $f_k(\cdot) : \mathbb{R}^q \rightarrow \mathbb{R}$ for each basis function coefficient $\alpha_k(\boldsymbol{\theta})$. The solution $\mathbf{u}(\mathbf{x}; \boldsymbol{\theta}^*)$ is then predicted as the linear combination of the K surrogate models, expressed as:

$$\hat{\mathbf{u}}(\mathbf{x}; \boldsymbol{\theta}^*) = \mathcal{M}_{\text{GP}}(\mathbf{x}; \boldsymbol{\theta}^*) = \sum_{k=1}^K \mu_k(\boldsymbol{\theta}^*) \phi_k(\mathbf{x}). \quad (5)$$

3.3 Prior correction with physical laws

Since the GP surrogate model is data-driven, it tends to have limited prediction capability within the parameter space Θ , but outside the training data points. To address this, we propose embedding physical laws from DEs directly into the surrogate model. By incorporating

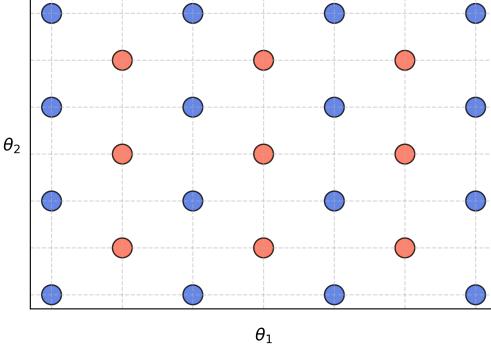


Figure 3: A basic strategy for gaining θ_{law} , the blue points are θ_{obs} for training $f_k(\cdot)$, and orange points are θ_{law} for learning physical law correction function $\omega_k(\cdot)$.

global DE constraints as prior knowledge, we enhance the GPR model performance, improving its ability to predict. This approach is independent of the data-driven nature of the GPR model and does not require additional training data from expensive numerical simulations. The new correction function $\omega(\theta|Law)$ is introduced to adjust the GP predicted coefficients, ensuring they align with the physical laws of the system, where Law is defined by the governing parametric PDE of the physical system law as Eq.(1).

The GP surrogate models for the coefficients of the DE solution is constructed with mean functions $m_k(\theta)$ of prior. A straightforward idea is that we can learn a better mean function of prior $\tilde{m}_k(\theta|Law)$ with considering the physical laws based on the original surrogate model. We denote the novel prior mean function as:

$$\tilde{m}_k(\theta|Law) = m_k(\theta) + \omega_k(\theta|Law),$$

where $\tilde{m}_k(\theta|Law)$ is physical law-corrected prior, $m_k(\theta)$ is prior mean function of $f_k(\cdot)$ which is general supposed to constant 0 and the $\omega_k(\theta|Law)$ is correction function that needs to be learned by physical law.

The reason for incorporating physical laws to refine the model is to enhance the surrogate's predictive capability for data outside the training set. Let $\theta_{\text{obs}} \subset \{\theta_i | (\theta_i, \alpha_i) \in \mathcal{D}_{\text{Low}}\}$ denote the training set used for learning the data-driven GP model. Then under this framework, the physical law-corrected prior function can be further reformulated as:

$$\tilde{m}_k(\theta|Law) = \begin{cases} 0, & \theta \in \theta_{\text{obs}}, \\ \omega_k(\theta|Law), & \theta \in \Theta \setminus \theta_{\text{obs}}, \end{cases}$$

By introducing a new prior function, it is possible to construct a novel LC-prior GP surrogate $\tilde{f}_k(\cdot)$ that simultaneously leverages data-driven and physical informed as :

$$\tilde{f}_k(\theta) \sim \mathcal{GP}(\tilde{m}_k(\theta|Law), \mathbf{k}_k(\theta, \theta'))$$

For any given parameter θ^* , we can subsequently derive the conditional posterior mean under the physical law by:

$$\begin{aligned}\tilde{\mu}_k(\theta^*) &= \mathbb{E}[f_k(\theta^*)|\theta^*, \mathcal{D}_{\text{Low}}, \text{Law}] \\ &= \tilde{m}_k(\theta^*|\text{Law}) + \mathbf{k}_{k*}^T \mathbf{K}_k^{-1} (\alpha_k - m_k(\theta)) \\ &= \omega_k(\theta^*|\text{Law}) + \mathbf{k}_{k*}^T \mathbf{K}_k^{-1} \alpha_k\end{aligned}$$

Similar to the approach in the preceding section, we can likewise reconstruct the function $\mathbf{u}(\mathbf{x}; \theta^*)$. The Eq.(5) can now be rewritten as:

$$\begin{aligned}\hat{\mathbf{u}}(\mathbf{x}; \theta^*) &= \mathcal{M}_{\text{LC}}(\mathbf{x}; \theta^*) = \sum_{k=1}^K \tilde{\mu}_k(\theta^*) \phi_k(\mathbf{x}) \\ &= \sum_{k=1}^K (\mu_k(\theta^*) + \omega_k(\theta^*)) \phi_k(\mathbf{x}) \\ &= \sum_{k=1}^K \omega_k(\theta^*) \phi_k(\mathbf{x}) + \mathcal{M}_{\text{GP}}(\mathbf{x}; \theta^*)\end{aligned}\tag{6}$$

In order to learn the mapping relationship from parameters θ to correction coefficients $\omega_k(\cdot)$, it is necessary to additionally extract a subset of parameter samples from the entire parameter space for physical law correction. So in the parameter space Θ , we extract some parameters θ_{law} outside the data-driven GP surrogate training set θ_{obs} , that used to learn the correction coefficients by physical law loss function. The Fig.3 shows a basic strategy for gaining θ_{law} .

The physical law loss function to learn the $\omega_k(\cdot)$ in Eq.(1) as:

$$\mathcal{L}oss = \|\mathcal{F}(\mathcal{M}_{\text{LC}}(\mathbf{x}; \theta_{\text{law}})) - \mathbf{f}(\mathbf{x}; \theta_{\text{law}})\|_{l_2}^2 + \lambda * \|\mathcal{G}(\mathcal{M}_{\text{LC}}(\mathbf{x}; \theta_{\text{law}})) - \mathbf{g}(\mathbf{x}; \theta_{\text{law}})\|_{l_2}^2,\tag{7}$$

where λ is a weight that controls the proportion of boundary conditions. We perform a global search within a range of $[-c \cdot \sigma_k(\theta_{\text{law}}), c \cdot \sigma_k(\theta_{\text{law}})]$ to optimize ω , where $\sigma_k(\theta_{\text{law}})$ is the standard deviation of each surrogate $f_k(\cdot)$, c is a constant that controls the size of the local interval.

In the optimization of this loss function, if uses numerical methods to recompute the differential function \mathcal{F} in every iteration, it would not only incur prohibitive computational costs but also degrade accuracy due to numerical errors, which would adversely affect the training and learning of our surrogate model. Benefiting from the RBF-FD method's characteristics for the specific problem, we can pre-store the evaluation matrix $E_h(X, Y)$ and differentiation matrices $D_h^\nabla(X, Y), D_h^{\nabla^2}(X, Y)$, etc. in sparse matrix format during the forward solving process. Since these matrices are independent of the equation parameters θ , they require no repeated solving. This advantageous feature enables that we can efficiently compute the loss function by reconstructing the RBF-FD linear system using the predicted solutions from the surrogate model. Here we denote $\mathcal{M}_{\text{LC}}(\cdot)$ compactly as $\hat{\mathbf{u}}(X; \theta)$, we can obtain:

$$\begin{cases} \|\mathcal{F}(\hat{\mathbf{u}}(X), D_h^\nabla(X, Y) E_h^\dagger(Y, X) \hat{\mathbf{u}}(X), D_h^{\nabla^2}(X, Y) E_h^\dagger(Y, X) \hat{\mathbf{u}}(X), \dots; \theta) - \mathbf{f}(X; \theta)\|_{l_2}^2 & \mathbf{x} \in \Omega \\ \|\mathcal{G}(\hat{\mathbf{u}}(X), D_h^\nabla(X, Y) E_h^\dagger(Y, X) \hat{\mathbf{u}}(X), D_h^{\nabla^2}(X, Y) E_h^\dagger(Y, X) \hat{\mathbf{u}}(X), \dots; \theta) - \mathbf{g}(X; \theta)\|_{l_2}^2 & \mathbf{x} \in \partial\Omega \end{cases}$$

Assume point sets X and Y consist of m and n discrete points located within the computational domain boundary respectively, the left terms of linear system within the Ω follow as:

$$L_f = \underbrace{\begin{pmatrix} \mathcal{F}(\Phi_1(\mathbf{x}_1), \nabla\Phi_1(\mathbf{x}_1), \dots; \boldsymbol{\theta}) & \dots & \mathcal{F}(\Phi_N(\mathbf{x}_1), \nabla\Phi_N(\mathbf{x}_1), \dots; \boldsymbol{\theta}) \\ \mathcal{F}(\Phi_1(\mathbf{x}_2), \nabla\Phi_1(\mathbf{x}_2), \dots; \boldsymbol{\theta}) & \dots & \mathcal{F}(\Phi_N(\mathbf{x}_2), \nabla\Phi_N(\mathbf{x}_2), \dots; \boldsymbol{\theta}) \\ \vdots & \ddots & \vdots \\ \mathcal{F}(\Phi_1(\mathbf{x}_m), \nabla\Phi_1(\mathbf{x}_m), \dots; \boldsymbol{\theta}) & \dots & \mathcal{F}(\Phi_N(\mathbf{x}_m), \nabla\Phi_N(\mathbf{x}_m), \dots; \boldsymbol{\theta}) \end{pmatrix}}_{m \times N} \underbrace{\begin{pmatrix} \hat{\mathbf{u}}(\mathbf{y}_1; \boldsymbol{\theta}) \\ \vdots \\ \hat{\mathbf{u}}(\mathbf{y}_n; \boldsymbol{\theta}) \\ \hline \hat{\mathbf{u}}(\mathbf{y}_{n+1}; \boldsymbol{\theta}) \\ \vdots \\ \hat{\mathbf{u}}(\mathbf{y}_N; \boldsymbol{\theta}) \end{pmatrix}}_{N \times 1}$$

the linear system on the boundary $\partial\Omega$ can likewise be expressed as:

$$L_g = \underbrace{\begin{pmatrix} \mathcal{G}(\Phi_1(\mathbf{x}_{m+1}), \nabla\Phi_1(\mathbf{x}_{m+1}), \dots; \boldsymbol{\theta}) & \dots & \mathcal{G}(\Phi_N(\mathbf{x}_{m+1}), \nabla\Phi_N(\mathbf{x}_{m+1}), \dots; \boldsymbol{\theta}) \\ \mathcal{G}(\Phi_1(\mathbf{x}_{m+2}), \nabla\Phi_1(\mathbf{x}_{m+2}), \dots; \boldsymbol{\theta}) & \dots & \mathcal{G}(\Phi_N(\mathbf{x}_{m+2}), \nabla\Phi_N(\mathbf{x}_{m+2}), \dots; \boldsymbol{\theta}) \\ \vdots & \ddots & \vdots \\ \mathcal{G}(\Phi_1(\mathbf{x}_M), \nabla\Phi_1(\mathbf{x}_M), \dots; \boldsymbol{\theta}) & \dots & \mathcal{G}(\Phi_N(\mathbf{x}_M), \nabla\Phi_N(\mathbf{x}_M), \dots; \boldsymbol{\theta}) \end{pmatrix}}_{(M-m) \times N} \underbrace{\begin{pmatrix} \hat{\mathbf{u}}(\mathbf{y}_1; \boldsymbol{\theta}) \\ \vdots \\ \hat{\mathbf{u}}(\mathbf{y}_n; \boldsymbol{\theta}) \\ \hline \hat{\mathbf{u}}(\mathbf{y}_{n+1}; \boldsymbol{\theta}) \\ \vdots \\ \hat{\mathbf{u}}(\mathbf{y}_N; \boldsymbol{\theta}) \end{pmatrix}}_{N \times 1}$$

where $\hat{\mathbf{u}}(Y; \boldsymbol{\theta}) = (\hat{\mathbf{u}}(\mathbf{y}_1; \boldsymbol{\theta}), \dots, \hat{\mathbf{u}}(\mathbf{y}_N; \boldsymbol{\theta}))^T$ can be calculated by the Moore-Penrose pseudoinverse of evaluation matrix: $\hat{\mathbf{u}}(Y; \boldsymbol{\theta}) = E_h^\dagger(Y, X)\hat{\mathbf{u}}(X; \boldsymbol{\theta})$.

Let the $M \times N$ matrix be denoted as $L = (L_f, L_g)^T$, where L_f represents discrete points within the computational domain Ω and L_g represents points on the boundary $\partial\Omega$. With the right terms $R = (R_f, R_g)^T$, where $R_f = (\mathbf{f}(\mathbf{x}_1; \boldsymbol{\theta}), \dots, \mathbf{f}(\mathbf{x}_m; \boldsymbol{\theta}))^T$, and $R_g = (\mathbf{g}(\mathbf{x}_{m+1}; \boldsymbol{\theta}), \dots, \mathbf{g}(\mathbf{x}_M; \boldsymbol{\theta}))^T$. Then the physics loss function Eq.(7) can be rewritten:

$$\text{Loss} = \|L_f E_h^\dagger(Y, X)\hat{\mathbf{u}}(X; \boldsymbol{\theta}) - R_f\|_2^2 + \lambda * \|L_g E_h^\dagger(Y, X)\hat{\mathbf{u}}(X; \boldsymbol{\theta}) - R_g\|_2^2$$

By optima the above loss function for each physics corrected point $\boldsymbol{\theta}_{\text{law}}$, we can obtain the optimization result of $\boldsymbol{\omega}$ at all discrete corrected points. By organizing the parameters and their corresponding correction coefficients, we obtain the physics corrected training set, denoted as $\mathcal{D}_{\text{law}} = \{(\boldsymbol{\theta}_{\text{law}(i)}, \boldsymbol{\omega}_{(i)})\}_{i=1}^M$, where $\boldsymbol{\omega}_{(i)} = (\omega_{1(i)}, \dots, \omega_{K(i)})^T$ corresponding the GP surrogate $f_k(\cdot)$ respectively. To further characterize the mapping relationship between the entire parameter space and the correction coefficients, we learn their connection through interpolation functions and the physics corrected training set. Similar to the GP surrogate learning in previous sections, we now need to independently learn K interpolation functions $s_k(\cdot) : \boldsymbol{\theta} \mapsto \omega_k$ based on the

Algorithm 1 LC-prior GP

Input: $\mathcal{D}_{\text{High}} = \{(\boldsymbol{\theta}_i, \mathbf{u}(\mathbf{x}; \boldsymbol{\theta}_i))\}_{i=1}^N$; the number of basis functions K ; prediction target $\boldsymbol{\theta}^*$

Output: Surrogate approximate solution $\mathcal{M}_{\text{LC}}(\mathbf{x}; \boldsymbol{\theta}^*)$.

- 1: Parameterize the output solutions in $\mathcal{D}_{\text{High}}$ by POD method and obtain the low-dimensional dataset $\mathcal{D}_{\text{low}} = \{(\boldsymbol{\theta}_i, \alpha_i)\}_{i=1}^N$.
 - 2: Construct GP surrogates for the K basis coefficients α_k : $f_k(\cdot) \sim \mathcal{GP}_k(\cdot)$ with dataset \mathcal{D}_{low} .
 - 3: Select M physical correction parameters $\{\boldsymbol{\theta}_{\text{law}(i)}\}_{i=1}^M \in \Omega$.
 - 4: **for** $i = 1$ **to** M **do**
 - 5: Predict $(\mu_k(\cdot), \sigma_k^2(\cdot))$ for each $\boldsymbol{\theta}_{\text{law}}$ using the GP surrogate $f_k(\cdot)$
 - 6: In the bounds $[-c\sigma^2(\cdot), c\sigma^2(\cdot)]$, optimize the correction coefficient ω_k using the physical law loss function in Eq. (7)
 - 7: **end for**
 - 8: Train a corrected model for each basis function coefficients $s_k(\cdot) : \boldsymbol{\theta} \mapsto \omega_k$ using the new training data $\mathcal{D}_{\text{law}} = \{(\boldsymbol{\theta}_{\text{law}(i)}, \omega_{(i)})\}_{i=1}^M$ and interpolation functions.
 - 9: Renew the prior mean using $s_k(\cdot)$ to get the LC-prior GP: $\tilde{f}_k(\cdot) \sim \mathcal{N}(\tilde{\mu}_k(\cdot), \tilde{\sigma}_k^2(\cdot))$.
 - 10: Compute the approximate solution: $\mathcal{M}_{\text{LC}}(\mathbf{x}; \boldsymbol{\theta}^*) = \sum_{k=1}^K \tilde{\mu}_k(\boldsymbol{\theta}^*) \phi_k(\mathbf{x})$, where the posterior mean $\tilde{\mu}_k(\boldsymbol{\theta}^*) = s_k(\boldsymbol{\theta}^*) + \mu_k(\boldsymbol{\theta}^*)$.
-

number of POD modes. In this training part, we are correcting the posterior mean of the learned GP surrogate, so there is no need to solve $\mathbf{u}(\mathbf{x}; \boldsymbol{\theta})$ using numerical methods to generate a new training set, all the corrections are based on the learned model $f_k(\cdot)$ and all the information comes from the physical constraints of the specific DE. The overall schematic of our LC-prior GP method is shown in Fig.1.

3.4 Prediction

The complete surrogate consists of two parts: the data-driven GPR model $f_k(\cdot) : \boldsymbol{\theta} \mapsto \alpha_k$ and the physical law corrected model $s_k(\cdot) : \boldsymbol{\theta} \mapsto \omega_k$, both models have the same input $\boldsymbol{\theta}$. We pass the prediction $s_k(\cdot)$ of the corrected model back to the GPR model as the physics constraints to renew the prior constant 0. So the LC-prior GP $\tilde{f}_k(\cdot)$ is also a Gaussian process with a law-corrected prior:

$$\tilde{f}_k(\boldsymbol{\theta}) \sim \mathcal{GP}(\tilde{m}_k(\boldsymbol{\theta}), \mathbf{k}_k(\boldsymbol{\theta}, \boldsymbol{\theta}')).$$

Given a new parameter $\boldsymbol{\theta}^*$, the conditional distribution can be written:

$$\tilde{f}_k(\boldsymbol{\theta}^*) \sim \mathcal{N}(\tilde{\mu}_k(\boldsymbol{\theta}^*), \tilde{\sigma}_k^2(\boldsymbol{\theta}^*)),$$

where $\tilde{\mu}_k(\cdot), \tilde{\sigma}_k^2(\cdot)$ are conditional mean and std:

$$\tilde{\mu}_k(\boldsymbol{\theta}^*) = \mathbb{E}[f_k(\boldsymbol{\theta}^*) | \boldsymbol{\theta}^*, \mathcal{D}_{\text{Low}}, \text{Law}] = s_k(\boldsymbol{\theta}^*) + \mathbf{k}_{k*}^T \mathbf{K}_k^{-1} \alpha_k = s_k(\boldsymbol{\theta}^*) + \mu_k(\boldsymbol{\theta}^*),$$

$$\tilde{\sigma}_k^2(\boldsymbol{\theta}^*) = \text{Var}[f_k(\boldsymbol{\theta}^*) | \boldsymbol{\theta}^*, \mathcal{D}_{\text{Low}}, \text{Law}] = \mathbf{k}_k(\boldsymbol{\theta}^*, \boldsymbol{\theta}^*) - \mathbf{k}_{k*}^T \mathbf{K}_k^{-1} \mathbf{k}_{k*}.$$

And we can reconstruct solution of the equation $\mathbf{u}(\mathbf{x}; \boldsymbol{\theta}^*)$ in domain for any given parameters by LC-prior GP:

$$\begin{aligned}\mathcal{M}_{\text{LC}}(\mathbf{x}; \boldsymbol{\theta}^*) &= \sum_{k=1}^K \tilde{\mu}_k(\boldsymbol{\theta}^*) \phi_k(\mathbf{x}) = \sum_{k=1}^K (\mu_k(\boldsymbol{\theta}^*) + s_k(\boldsymbol{\theta}^*)) \phi_k(\mathbf{x}) \\ &= \sum_{k=1}^K s_k(\boldsymbol{\theta}^*) \phi_k(\mathbf{x}) + \mathcal{M}_{\text{GP}}(\mathbf{x}; \boldsymbol{\theta}^*)\end{aligned}$$

where $\tilde{\mu}_k(\cdot)$ is the predicted posterior mean of LC-prior GP $\tilde{f}_k(\cdot)$. The overall framework of LC-prior GP can be defined in Algorithm 1.

3.5 Extension to multi-coupled PDE systems

In many real-world scenarios, the governing equations are not isolated single-physics PDEs but rather multi-coupled systems involving strongly interacting physical processes. Extending the LC-prior GP framework to such settings requires careful treatment of the correlations among multiple dependent variables. Let us denote the multi-coupled PDE system as:

$$\begin{cases} \mathcal{F}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \dots, \mathbf{u}^{(m)}; \boldsymbol{\theta}) = \mathbf{f}(\mathbf{x}; \boldsymbol{\theta}), & \mathbf{x} \in \Omega, \\ \mathcal{G}(\mathbf{u}^{(1)}, \mathbf{u}^{(2)}, \dots, \mathbf{u}^{(m)}; \boldsymbol{\theta}) = \mathbf{g}(\mathbf{x}; \boldsymbol{\theta}), & \mathbf{x} \in \partial\Omega. \end{cases} \quad (8)$$

where $\mathbf{u}^{(j)}(\mathbf{x}; \boldsymbol{\theta})$ represents the j -th physical variable in the coupled system, and $\boldsymbol{\theta}$ is the vector of parameters. Suppose we have already obtained a set of training set $\{(\boldsymbol{\theta}_i, \mathbf{u}_i^{(1)}, \dots, \mathbf{u}_i^{(m)})\}_{i=1}^N$ using the RBF-FD method and follow the same procedure as in the single-physics case, each solution field is projected onto a reduced basis constructed via POD from snapshot data:

$$\mathbf{u}^{(j)}(\mathbf{x}; \boldsymbol{\theta}) \approx \sum_{k=1}^{K_j} \alpha_k^{(j)}(\boldsymbol{\theta}) \phi_k^{(j)}(\mathbf{x}), \quad j = 1, \dots, m.$$

where K_j is the number of POD modes for the j -th physical variable, $\boldsymbol{\phi}^{(j)} = (\phi_1^{(j)}, \dots, \phi_{K_j}^{(j)})^T$ denote the modes and $\boldsymbol{\alpha}^{(j)} = (\alpha_1^{(j)}, \dots, \alpha_{K_j}^{(j)})^T$ are the corresponding reduced coefficients. For each coefficient, a Gaussian process surrogate is trained:

$$f_k^{(j)}(\boldsymbol{\theta}) \sim \mathcal{GP}(m_k^{(j)}(\boldsymbol{\theta}), \mathbf{k}_k^{(j)}(\boldsymbol{\theta}, \boldsymbol{\theta}')), \quad j = 1, \dots, m.$$

with the pure data-driven surrogate $\mathcal{M}_{\text{GP}}^{(j)}$ for each variable $\mathbf{u}^{(j)}(\mathbf{x}; \boldsymbol{\theta})$.

A naive extension would treat these surrogates independently, but this ignores the cross-variable couplings encoded in the governing equations. To address this, the LC-prior GP framework is adapted by introducing joint correction coefficient $\omega_k^{(j)}(\boldsymbol{\theta} | \text{Law})$ for each GP regression model to construct physical law corrected surrogate. These coefficients are optimized simultaneously under the constraint of the entire physical coupled system:

$$\mathcal{Loss} = \|\mathcal{F}(\mathcal{M}_{\text{LC}}^{(1)}, \dots, \mathcal{M}_{\text{LC}}^{(m)}; \boldsymbol{\theta}) - \mathbf{f}(\mathbf{x}; \boldsymbol{\theta})\|_{l_2}^2 + \lambda * \|\mathcal{G}(\mathcal{M}_{\text{LC}}^{(1)}, \dots, \mathcal{M}_{\text{LC}}^{(m)}; \boldsymbol{\theta}) - \mathbf{g}(\mathbf{x}; \boldsymbol{\theta})\|_{l_2}^2, \quad (9)$$

where $\mathcal{M}_{\text{LC}}^{(j)}$ denotes the law-corrected surrogate reconstruction of the corresponding variable. Consistent with the previous formulation, we employ a small set of parameter samples to perform optimization based on the above loss function together with the trained GP surrogates, and learn the global correction mapping $s_k^{(j)}(\cdot) : \boldsymbol{\theta} \mapsto \omega_k^{(j)}$ for each surrogate model in the parameter space through interpolation functions. The interpolation functions are then backpropagated to the original GP surrogates as the new prior mean functions, yielding the LC-prior GP $\tilde{f}_k^{(j)}(\cdot)$.

For given $\boldsymbol{\theta}^*$, any physical variable $\mathbf{u}^{(j)}(\mathbf{x}; \boldsymbol{\theta}^*)$ in multi-coupled system can be efficiently predicted through the LC-prior GP:

$$\begin{aligned}\mathcal{M}_{\text{LC}}^{(j)}(\mathbf{x}; \boldsymbol{\theta}^*) &= \sum_{k=1}^{K_j} \mathbb{E}[\tilde{f}_k^{(j)}(\boldsymbol{\theta}^*) | \boldsymbol{\theta}^*, \text{Law}] \phi_k(\mathbf{x}) = \sum_{k=1}^{K_j} (\mu_k^{(j)}(\boldsymbol{\theta}^*) + s_k^{(j)}(\boldsymbol{\theta}^*)) \phi_k(\mathbf{x}) \\ &= \sum_{k=1}^{K_j} s_k^{(j)}(\boldsymbol{\theta}^*) \phi_k(\mathbf{x}) + \mathcal{M}_{\text{GP}}^{(j)}(\mathbf{x}; \boldsymbol{\theta}^*), \quad j = 1, \dots, m.\end{aligned}$$

This joint optimization ensures that correction terms are consistent across all variables and enforce the interdependencies dictated by the PDE system. In this way, the multi-output LC-prior GP explicitly encodes both the data-driven correlations and the physical couplings, thereby yielding predictions that are physically coherent and numerically stable.

3.6 Parameter estimation by LC-prior GP

Inferring unknown parameters $\boldsymbol{\theta}$ from indirect observations \mathbf{y} is a critical application. Typically, the observed data \mathbf{y} is contaminated with noise, often modeled as $\mathbf{y} = \mathcal{M}_{\text{true}}(\mathbf{x}; \boldsymbol{\theta}) + \boldsymbol{\epsilon}$, where $\mathcal{M}_{\text{true}}(\mathbf{x}; \boldsymbol{\theta})$ represents the true model output and $\boldsymbol{\epsilon}$ is the noise term. There are two general frameworks for parameter estimation: (1) deterministic methods for point estimation, and (2) Bayesian inverse methods for posterior estimation. Here we propose to infer the unknown parameters in Bayesian framework with use of our LC-prior GP surrogate.

In Bayesian setting, the prior belief about the parameter $\boldsymbol{\theta}$ is encoded in the probability distribution $\pi_{\text{prior}}(\boldsymbol{\theta})$. Here we use a uniform distribution as prior for highlighting the action of likelihood function. Our aim is to infer the distribution of $\boldsymbol{\theta}$ conditioned on the given data \mathbf{y} , the LC-prior GP surrogate \mathcal{M}_{LC} and physical law Law , the posterior distribution $\pi(\boldsymbol{\theta} | \mathbf{y}, \mathcal{M}_{\text{LC}}, \text{Law})$. By the Bayes' rule, we have

$$\pi(\boldsymbol{\theta} | \mathbf{y}, \mathcal{M}_{\text{LC}}, \text{Law}) \propto P_\epsilon(\mathbf{y} - \mathcal{M}_{\text{LC}}(\mathbf{x}; \boldsymbol{\theta})) * \pi(\text{Law} | \mathcal{M}_{\text{LC}}, \mathbf{y}) * \pi_{\text{prior}}(\boldsymbol{\theta}), \quad (10)$$

where $P_\epsilon(\mathbf{y} - \mathcal{M}_{\text{LC}}(\mathbf{x}; \boldsymbol{\theta}))$ is the likelihood $\pi(\mathbf{y}, \mathcal{M}_{\text{LC}} | \boldsymbol{\theta})$ and $\pi(\text{Law} | \mathcal{M}_{\text{LC}}, \mathbf{y})$ is the conditional distribution of physical law defined by the loss function Eq. (7)

$$\begin{aligned}P_\epsilon(\mathbf{y} - \mathcal{M}_{\text{LC}}(\mathbf{x}; \boldsymbol{\theta})) &\propto \exp(-\beta_1 * \|\mathbf{y} - \mathcal{M}_{\text{LC}}(\mathbf{x}; \boldsymbol{\theta})\|_{l_2}^2), \\ \pi(\text{Law} | \mathcal{M}_{\text{LC}}) &\propto \exp(-\beta_2 * \|\mathcal{Loss}\|_{l_2}^2),\end{aligned}$$

fidelity to the DEs can be measured by $\pi(\text{Law}|\mathcal{M}_{\text{LC}})$, while fidelity to the data can be measured by $P_\epsilon(\mathbf{y} - \mathcal{M}_{\text{LC}}(\mathbf{x}; \boldsymbol{\theta}))$. Weights for data and equation information are assumed in $\boldsymbol{\beta} = (\beta_1, \beta_2)$.

The unnormalized posterior samples can be easily sampled using MCMC method such as Metropolis-Hastings (MH) algorithm, Gibbs sampling and DRAM et al. Here we use MH sampling algorithm to draw samples from the unnormalized posterior Eq. (10). The Metropolis-Hastings (MH) algorithm is one of the most popular MCMC methods. An MH step of invariant distribution $\pi(\boldsymbol{\theta})$ and proposal distribution $\pi_q(\boldsymbol{\theta}'|\boldsymbol{\theta}^i)$ involves sampling a candidate value $\boldsymbol{\theta}'$ given the current value $\boldsymbol{\theta}^i$ according to $\pi_q(\boldsymbol{\theta}'|\boldsymbol{\theta}^i)$. The Markov chain then moves towards $\boldsymbol{\theta}'$ with acceptance probability

$$\mathcal{A}(\boldsymbol{\theta}^i, \boldsymbol{\theta}') = \min \left\{ 1, \frac{\pi(\boldsymbol{\theta}'|y)q(\boldsymbol{\theta}^i|\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta}^i|y)q(\boldsymbol{\theta}'|\boldsymbol{\theta}^i)} \right\},$$

otherwise it remains at $\boldsymbol{\theta}^i$. In our work, we draw from $\pi(\boldsymbol{\theta}|y, \mathcal{F})$ using MH sampling:

1. initialize $\boldsymbol{\theta}^1$
2. For $i = 1$ to N
 - (a) Sample $\boldsymbol{\theta}'$ from the proposal distribution $\pi_q(\boldsymbol{\theta}'|\boldsymbol{\theta}^i)$.
 - (b) Compute $\mathcal{A}(\boldsymbol{\theta}^i, \boldsymbol{\theta}') = \min \left\{ 1, \frac{\pi(\boldsymbol{\theta}'|y)\pi_q(\boldsymbol{\theta}^i|\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta}^i|y)\pi_q(\boldsymbol{\theta}'|\boldsymbol{\theta}^i)} \right\}$ and sample u from $\mathcal{U}[0, 1]$.
 - (c) If $u < \mathcal{A}(\boldsymbol{\theta}^i, \boldsymbol{\theta}')$ then accept $\boldsymbol{\theta}'$. Otherwise, set $\boldsymbol{\theta}' = \boldsymbol{\theta}^i$.

Bayesian methods offer advantages by explicitly accounting for parameter uncertainty, making them more robust in the presence of noisy or sparse data. By providing a full posterior distribution, they enable probabilistic interpretation and decision-making under uncertainty. Both deterministic and Bayesian methods can be computationally intensive, especially for high-dimensional models or when the likelihood function is complex. The surrogate model we construct mitigates the challenge of high computational cost by replacing the map $\mathcal{M}_{\text{RBF-FD}}(\cdot)$ with $\mathcal{M}_{\text{LC}}(\cdot)$.

4 Numerical examples

To validate the performance of LC-prior GP method, we present five numerical examples based on our proposed method. In the numerical examples of the Reaction-diffusion equation, miscible flooding model and the Navier-Stokes model, we primarily demonstrate the surrogate performance of the LC-prior GP method on the irregular computational domain. To illustrate the accuracy of this approach, we compare it with the standard GP method and the DMD-wiNN method proposed in reference [29], further highlighting the performance of the LC-prior GP method under small-sample settings. Additionally, within the framework of our surrogate model, we achieve highly efficient parameter estimation applications in miscible flooding model and the Korteweg-de Vries equation using given noisy observations.

Table 1: The relative errors of GP method, LC-prior GP method and DMD-wiNN.

	GP	LC-prior GP	DMD-wiNN
t=0.5	0.0763	0.0211	0.0275
t=1.0	0.1127	0.0307	0.0408
All discrete time	0.0839	0.0233	0.0304

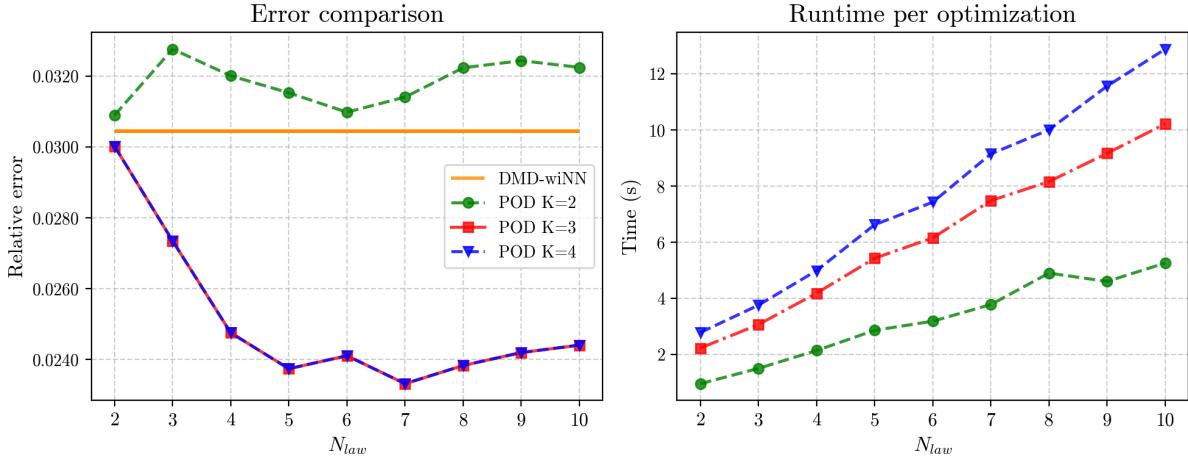


Figure 4: Relative errors between the mean solutions obtained by the RBF-FD and the LC-prior GP with different N_{law} and POD modes K (left), and the optimization time with physical law correction under the corresponding conditions (right).

4.1 Reaction-diffusion model

In this subsection, we consider a reaction-diffusion equation, which involves a parameter ϵ and with a homogeneous Dirichlet boundary condition:

$$\begin{cases} u_t = \epsilon^2 \Delta u - F'(u) + f_0(x, y, t), & \text{in } (0, T) \times \Omega, \\ F(u) = \frac{1}{4}(u^2 - 1)^2, & \text{in } (0, T) \times \Omega, \\ u(0, \cdot) = u_0, & \text{in } \Omega, \end{cases} \quad (11)$$

For the example in this subsection, Eq.(11) with $f_0(x, y, t) = 0$ is the classical Allen-Cahn equation, which involves a parameter ϵ related to the interface thickness. We set $T = 1$ and computational domain is $\Omega = [-1, 1] \times [-1, 1] \in \mathbb{R}^2$. The initial value is given by

$$u_0(x, y) = \begin{cases} 1, & \text{if } (x^2 + y^2)^{\frac{1}{2}} \leq \frac{1}{8}(3 + 3 \sin(5\gamma)), \\ 0, & \text{elsewhere,} \end{cases}$$

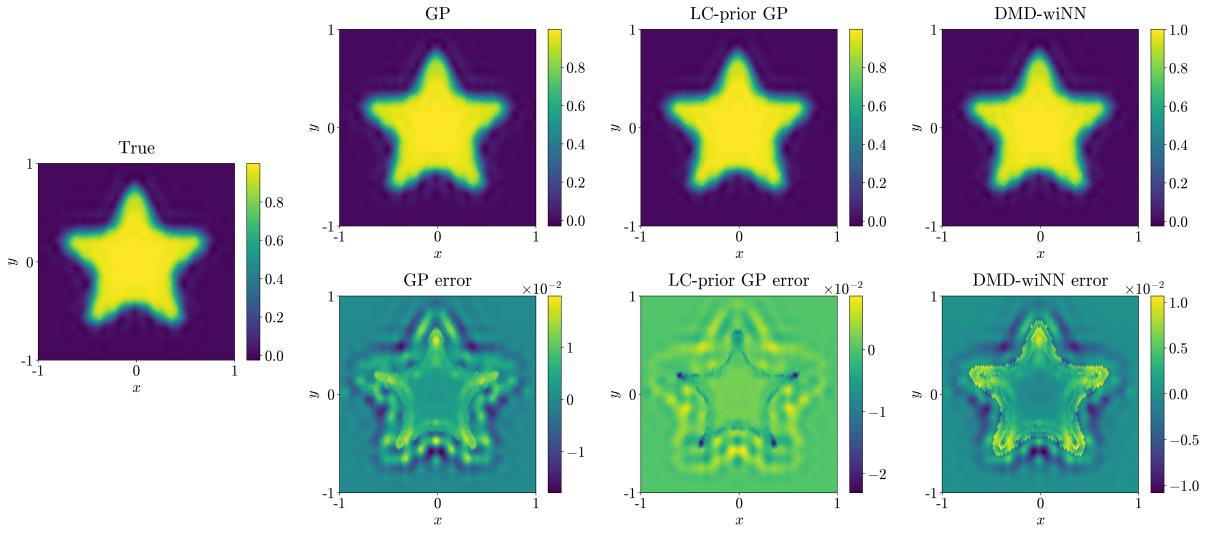


Figure 5: The first column presents the mean solutions of the RBF-FD method at $t=0.5$, while columns two to four display the results at $t=0.5$ for GP, LC-prior GP and DMD-wiNN methods, respectively. The first row shows the mean solutions, and the second row provides the corresponding errors.

with

$$\gamma = \begin{cases} \arccos\left(\frac{x}{\sqrt{x^2+y^2}}\right), & y \geq 0, \\ 2\pi - \arccos\left(\frac{x}{\sqrt{x^2+y^2}}\right), & y < 0. \end{cases}$$

The discrete scheme we used in the temporal direction is

$$u^{n+1} - \tau\epsilon^2\Delta u^{n+1} = u^n - \tau f(u^n).$$

Here $\boldsymbol{\theta} = \epsilon$. The training set with 3 sample parameters is uniformly generated from the interval $\pi_{\text{prior}}(\epsilon) \sim \mathcal{U}[0, 1]$. For the testing set, we randomly generate 200 samples from the uniform distribution $\pi_{\text{prior}}(\epsilon)$. In this numerical example, only the surrogate model for $u(x, y, t; \boldsymbol{\theta})$ needs to be constructed.

In this section, to illustrate the impact of the number of parameter points used for physical law correction and the number of POD modes on the final modeling accuracy, we compare the relative errors of the LC-prior GP and the RBF-FD method on the test set under different conditions, as shown in Fig.4(left). Here, N_{law} denotes the number of samples used to learn the physical-law prior function, obtained by uniformly sampling within the interval $\pi_{\text{prior}}(\epsilon) \sim \mathcal{U}[0, 1]$, and K represents the first K POD modes extracted from the eigen-decomposition of the snapshot matrix. When $K = 2$, the number of selected modes is too small to provide a good approximation of the solution space of the original function; under this condition, even with additional physical law corrections, the surrogate performance cannot be

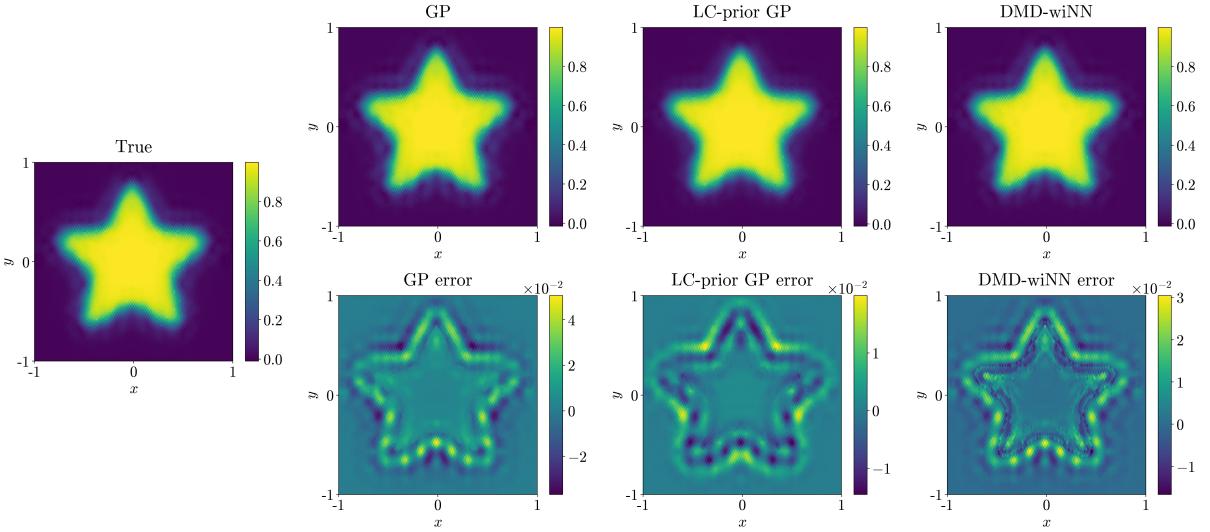


Figure 6: The first column presents the mean solutions of the RBF-FD method at $t=1$, while columns two to four display the results at $t=1$ for GP, LC-prior GP and DMD-wiNN methods, respectively. The first row shows the mean solutions, and the second row provides the corresponding errors.

significantly improved. In contrast, when the number of modes is increased to $K = 3$, the overall system is well captured, and appropriately increasing the number of correction samples N_{law} further reduces the prediction error of the final model. It is worth noting that when the number of modes is further increased to $K = 4$, the relative prediction errors are nearly the same as those with $K = 3$. This indicates that the surrogate performance of the LC-prior GP is closely tied to the approximation capability of the POD method: once the approximation is sufficiently accurate, adding more modes does not further enhance performance. Fig.4(right) presents the corresponding optimization time, whose computational cost grows with both N_{law} and K . Based on these results, one can set an error threshold such that when the POD approximation error falls below this threshold, there is no need to keep increasing K and incurring additional computational costs, while still maintaining a high-quality surrogate model.

In order to present more concrete results, we choose $N_{\text{law}} = 7$ and $K = 3$ for the surrogate model construction. The Table.1 presents the relative errors between three methods and the RBF-FD full-order solution at $t = 0.5$, $t = 1$ and all discrete time instants. Although the conventional GP method performs poorly on the test set, the physical law-corrected method demonstrates relative errors smaller than those of the DMD-wiNN method with equivalent sample size, confirming the effectiveness of our approach. Fig.5 and Fig.6 show the means and errors of the test set solutions at $t = 0.5$ and $t = 1$ respectively. The first column shows the full-order solution obtained by RBF-FD, followed by the GP and LC-prior GP surrogate model results in the second and third columns, and the DMD-wiNN method's result in the fourth column. The first row presents the means of solutions for testing set, while the second row shows their corresponding errors between the full-order solution.

Table 2: The relative errors with different number of physical law corrected samples

	GP	LC-prior GP
$N_{\text{law}} = 4$	0.3105	0.1732
$N_{\text{law}} = 9$	0.3105	0.0952

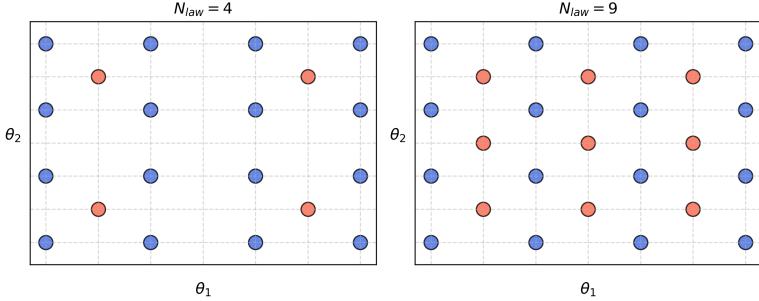


Figure 7: Two different strategies to chose the physical law correction points. The blue points are GP surrogate training data θ_{obs} and orange points are correction points θ_{law}

4.2 The KdV Equation

The Korteweg–de Vries equation is a nonlinear partial differential equation arising in the study of a number of different physical systems, e.g., water waves, plasma physics, anharmonic lattices, and elastic rods [36]. It describes the long time evolution of small-but-finite amplitude dispersive waves:

$$\begin{cases} u_t + \theta_1 u \frac{\partial u}{\partial x} + \theta_2 \frac{\partial^3 u}{\partial x^3} = 0, & \text{in } [0, T] \times \Omega \\ u(x, 0) = \frac{c_1}{2} \operatorname{sech}^2\left(\frac{\sqrt{c_1}}{2}(x - L_1)\right) + \frac{c_2}{2} \operatorname{sech}^2\left(\frac{\sqrt{c_2}}{2}(x - L_2)\right), & \text{in } [0, T] \times \Omega \end{cases} \quad (12)$$

where $\theta = (\theta_1, \theta_2)$ and $\Omega \in \mathbb{R}^1$ we construct surrogate models for the nonlinear term θ_1 and the dispersion term θ_2 to $u(x, t; \theta)$, with $\pi_{\text{prior}}(\theta_1) \sim \mathcal{U}(2, 8)$ and $\pi_{\text{prior}}(\theta_2) \sim \mathcal{U}(0.2, 2)$. This is a highly nonlinear PDE, so 16 training set parameter θ_{obs} are uniformly selected in prior distribution.

In order to illustrate the accuracy of parameter estimation, here we supposed a true parameters $\theta^* = (6, 1)$. And given noise observation $\mathbf{y} = \mathcal{M}_{\text{RBF-FD}}(\theta^*) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ to get posterior samples under our proposed framework.

In this study, we investigate how the number of correction points selected based on physical laws affects the performance of the surrogate model, and further extend the analysis to a two-dimensional parameter space. Specifically, the number of θ_{law} used for correction is recorded as N_{law} , we compare two point-selection strategies: $N_{\text{law}} = 4$ and $N_{\text{law}} = 9$, as shown in Fig.7,

Table 3: MCMC results for the KdV Equation with $\theta^* = (6, 1)$

Method		$N_{\text{law}} = 9$	$N_{\text{law}} = 4$
$\sigma_{\text{obs}}^2 = 0.1$	LC-prior GP	(6.08 ± 1.16, 1.16 ± 0.67)	(4.61 ± 4.43, 1.67 ± 3.14)
	GP	(5.32 ± 2.40, 1.22 ± 1.69)	—
$\sigma_{\text{obs}}^2 = 0.2$	LC-prior GP	(5.95 ± 1.11, 1.09 ± 0.62)	(4.37 ± 4.12, 1.24 ± 3.02)
	GP	(6.43 ± 2.52, 1.93 ± 1.71)	—

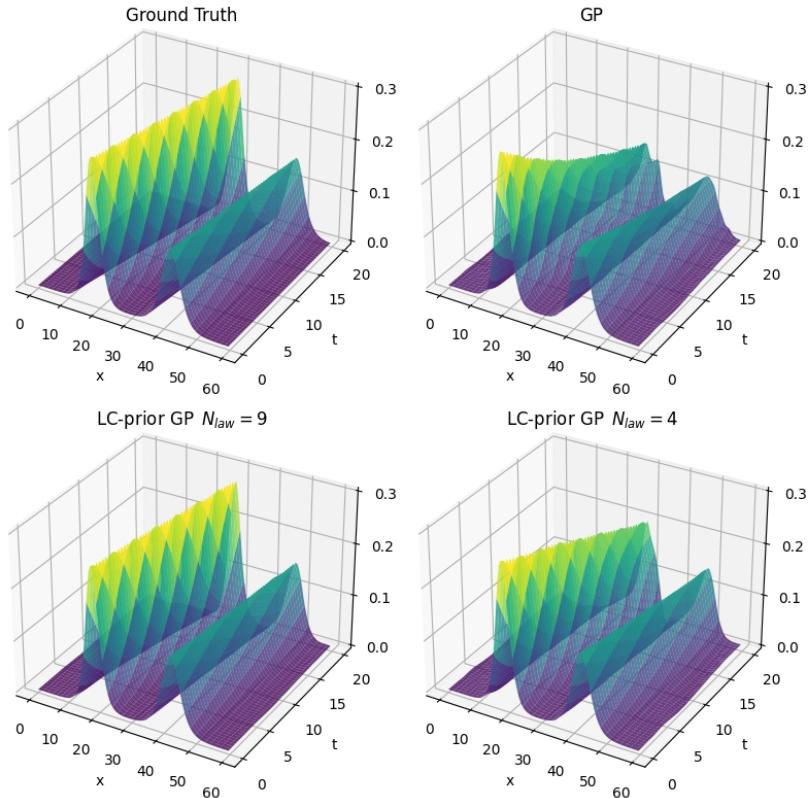


Figure 8: The prediction of $u(x, t; \theta^*)$ for KdV equation by different methods. The top-left figure shows the ground truth, the top-right figure represents the standard GP surrogate, and the second row displays LC-prior GP with different number of the physical law corrected points

and evaluate their impact on the accuracy of the surrogate model and the MCMC posterior distribution. Fig.8 shows that, compared with the GP surrogate model driven solely by data, our method greatly reduces the error in the surrogate model. As the number of the θ_{law} increases, the performance of the LC-prior GP improves, but the optimization cost for ω also increases. The specific relative errors are presented in Table 2, since the standard GP method does not require θ_{law} modification, the errors for standard GP only need calculation once. The experimental

results indicate that appropriately increasing the number of correction points can substantially enhance the robustness of LC-prior GP predictions.

For the parameters estimation in the KdV equation, results are reflected in the posterior samples of the MCMC, as shown in Table 3. The posterior samples obtained using LC-prior GP with $N_{\text{law}} = 9$ exhibit a more accurate mean and smaller standard deviation compared to both the standard GP and $N_{\text{law}} = 4$. All the experimental results above demonstrate that our method provides a more competitive performance, even when dealing with nonlinear PDEs that describe complex physical phenomena, as reflected in both the surrogate model errors and the posterior distribution of samples.

4.3 Incompressible miscible flooding model

In this subsection, we test a model with multiple parameters. The incompressible miscible flooding in the porous media is widely used in the engineering fields, such as the reservoir simulation and the exploration of the underground water and oil. The classical equations are given as follows:

$$\begin{cases} \nabla \cdot \mathbf{u} = q, & \text{in } (0, T) \times \Omega, \\ \mathbf{u} = -\frac{\kappa}{\mu(c)} \nabla p, & \text{in } (0, T) \times \Omega, \\ \phi c_t + \mathbf{u} \cdot \nabla c = \nabla \cdot (\mathbf{D}(\mathbf{u}) \nabla c), & \text{in } (0, T) \times \Omega, \end{cases} \quad (13)$$

where $\Omega \in \mathbb{R}^d$ with $d = 2$. The parameter κ represents the permeability, μ represents the viscosity, and ϕ is the porosity. The unknown functions \mathbf{u} , p and c are the velocity, pressure, and concentration, respectively.

By replacing the velocity \mathbf{u} with the pressure p , Eq.(13) is equivalent to:

$$\begin{cases} -\frac{\kappa}{\mu(c)} \Delta p = q, & \text{in } (0, T) \times \Omega, \\ \phi c_t - \frac{\kappa}{\mu(c)} \nabla p \cdot \nabla c = \nabla \cdot (\mathbf{D}(\mathbf{u}) \nabla c), & \text{in } (0, T) \times \Omega, \end{cases} \quad (14)$$

with

$$\mathbf{D}(\mathbf{u}) = d_m I + |\mathbf{u}|(d_l E(\mathbf{u}) + d_t(I - E(\mathbf{u}))),$$

where I is the identity matrix, d_m is the effective diffusion coefficient, d_l is the longitudinal dispersion coefficient, d_t is the transverse dispersion coefficient, and $E(\mathbf{u})$ is the tensor that projects onto the velocity $\mathbf{u} = (u_1, \dots, u_d)$ direction with each component defined as follows:

$$(E(\mathbf{u}))_{i,j} = \frac{u_i u_j}{|\mathbf{u}|^2}, \quad 1 \leq i, j \leq d.$$

Table 4: The relative errors of GP method, LC-prior GP method and DMD-wiNN.

	GP	LC-prior GP	DMD-wiNN
Two parameters example	0.3269	0.1744	0.2112
Three parameters example	0.2056	0.1229	0.1367

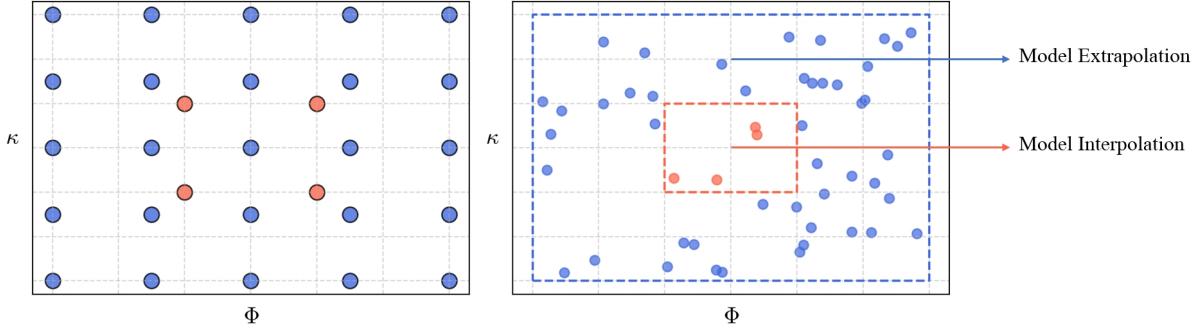


Figure 9: The strategy to chose training set and physics correction set (left) and some testing set for a brief illustration (right).

For this example, we consider $\mathbf{D}(\mathbf{u}) = d_m I$ and set $T = 0.1$. We select an irregular region in the two-dimensional space Ω . The radius of this region satisfies the following requirement:

$$r_a = 1 + \frac{\sin(7\gamma) + \sin(\gamma)}{10}, \quad \gamma \in [0, 2\pi].$$

The uniform division of the DistMesh is used for the spatial division. The computation domain is $[-1.5, 1.5] \times [-1.5, 1.5]$, and the distance between the longest two points after the division is $h = 0.04$. The discrete numerical scheme of the temporal direction is as follows:

$$\begin{cases} -\frac{\kappa}{\mu(c)} \Delta p^{n+1} = q^{n+1}, \\ \phi \frac{c^{n+1} - c^n}{\tau} - \frac{\kappa}{\mu(c)} \nabla p^n \cdot \nabla c^{n+1} - d_m \Delta c^{n+1} = 0. \end{cases}$$

4.3.1 Two parameters example

In this experiment, we aim to examine whether applying physics law corrections beyond the training set can further improve the LC-prior GP's extrapolation performance on the test set, so the parameters κ and ϕ in Eq.(14) are uncertain. Here $\theta = (\kappa, \phi)$. The training set is constructed by selecting the central points from the prior distributions, i.e., $\kappa = \{-1, 1\}$ from $\pi_{\text{prior}}(\kappa) \sim \mathcal{U}[-3, 3]$ and $\phi = \{-2, 2\}$ from $\pi_{\text{prior}}(\phi) \sim \mathcal{U}[-6, 6]$, yielding 4 training samples from their Cartesian product, while the testing set contains 20 points randomly scattered in the

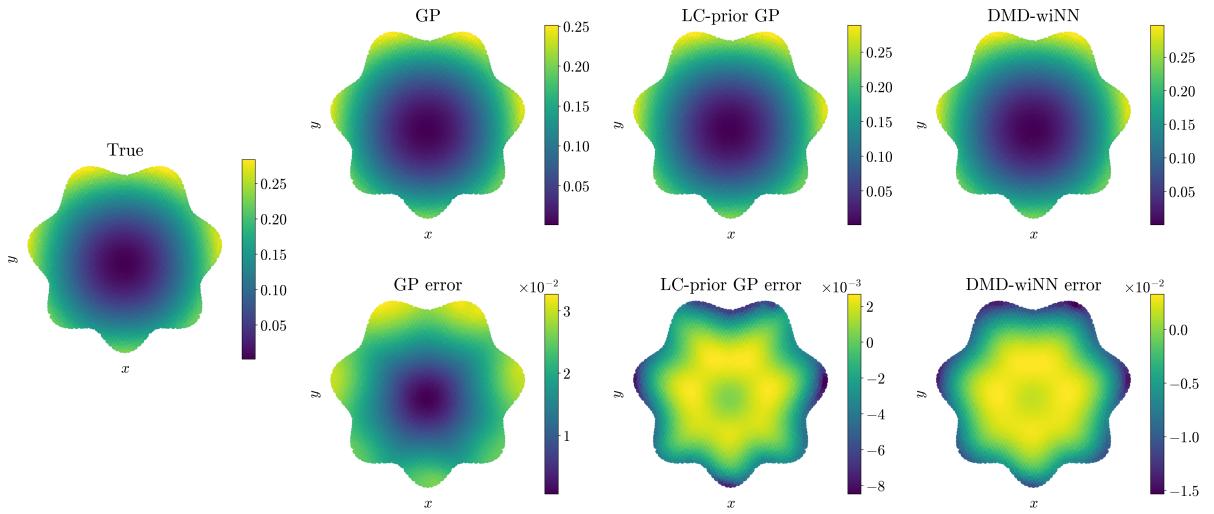


Figure 10: The first column presents the mean solutions of the RBF-FD method at $T=0.1$, while columns two to four display the results at $T=0.1$ for GP, LC-prior GP and DMD-wiNN methods, respectively. The first row shows the mean solutions, and the second row provides the corresponding errors.

Table 5: MCMC results for two parameters example with $\theta^* = (0.64, 4.97)$

Method		(Mean \pm Std) of posterior
$\sigma_{\text{obs}}^2 = 0.1$	LC-prior GP	(0.68 \pm 0.31, 5.12 \pm 0.63)
	GP	(2.21 \pm 0.54, 5.30 \pm 0.55)
$\sigma_{\text{obs}}^2 = 0.2$	LC-prior GP	(0.98 \pm 0.33, 4.72 \pm 0.48)
	GP	(1.55 \pm 0.51, 3.90 \pm 0.35)

same intervals. Thus, the size of the training set is 4 and the size of the testing set is 400. For the physics corrected set, 5 equally spaced points were sampled from each of the uniform prior distribution, yielding a total of 25 data points. Fig.9 shows a concise schematic representation.

For this numerical example, Eq.(14) involves two unknown functions: pressure p and concentration c . Therefore, it is necessary to construct surrogate models for both $p(x, y, t; \theta)$ and $c(x, y, t; \theta)$ simultaneously to enable subsequent correction through the physics corrected loss function. During the parametric representation stage, the first three POD modes are selected as the basis functions for both variables.

The Fig.10 visualizes the results of a two-parameter miscible flooding model at $t = 0.1$, comparing the performance of three methods across 400 test samples under the specified problem setup. The first row displays the mean solutions of each method on the test set, while the second row presents their corresponding errors against the RBF-FD full-order mean solutions. The proposed LC-prior GP method achieves relative errors on the order of 10^{-3} . Additional details are provided in the accompanying Table.4, which compares the aggregated errors of

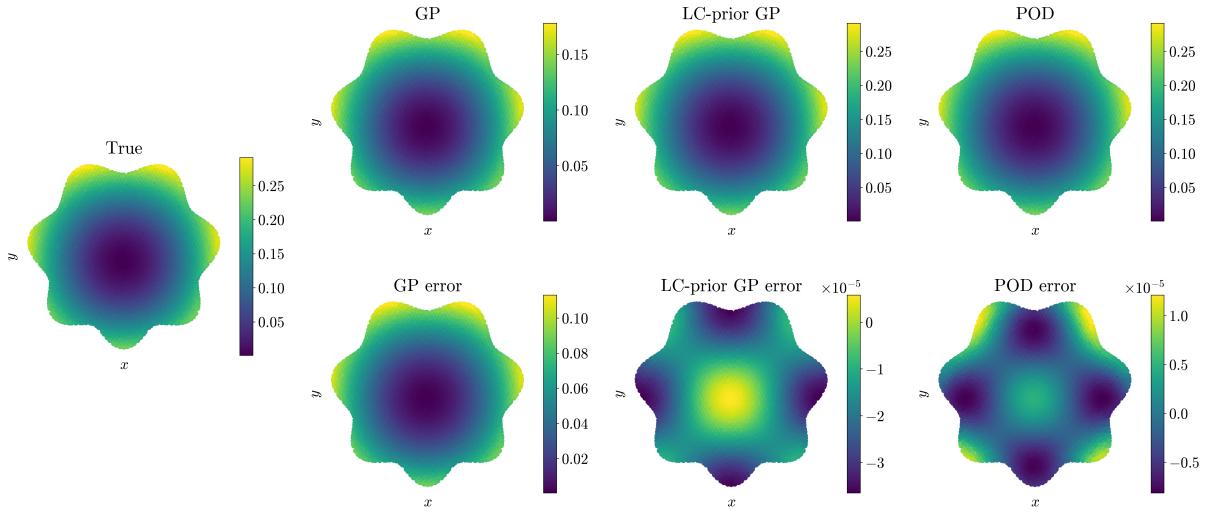


Figure 11: The first column presents one physical law-corrected sample’s solutions of the RBF-FD method at $T=0.1$, while columns two to four display the results at $T=0.1$ for GP, LC-prior GP and POD, respectively. The first row shows the mean solutions, and the second row provides the corresponding errors.

Table 6: MCMC results for three parameters example with $\theta^* = (1.45, -2.21, -4.68)$

Method		(Mean \pm Std) of posterior
$\sigma_{\text{obs}}^2 = 0.1$	LC-prior GP	(1.59 \pm 0.46, -2.41 \pm 0.70, -4.47 \pm 1.15)
	GP	(1.31 \pm 0.25, -1.08 \pm 0.65, -1.57 \pm 1.30)
$\sigma_{\text{obs}}^2 = 0.2$	LC-prior GP	(1.78 \pm 0.26, -2.01 \pm 0.27, -4.72 \pm 0.43)
	GP	(0.91 \pm 0.28, 2.83 \pm 0.49, -5.58 \pm 0.62)

surrogate models at ten discrete time points within the interval $[0, T]$. Notably, the LC-prior GP method demonstrates superior prediction accuracy over both standard GP method and DMD-wiNN method.

In parameter estimation applications, to further evaluate the model’s robustness, we randomly selected a sample of testing set with $\theta^* = (0.64, 4.97)$ and added white Gaussian noise at varying intensity levels to simulate observed measurements for MCMC sampling. The specific results of the posterior distribution’s mean and standard deviation are presented in the Table.5. The posterior samples based on the LC-prior GP surrogate demonstrate more accurate mean estimates compared to those from the standard GP surrogate under both noise conditions.

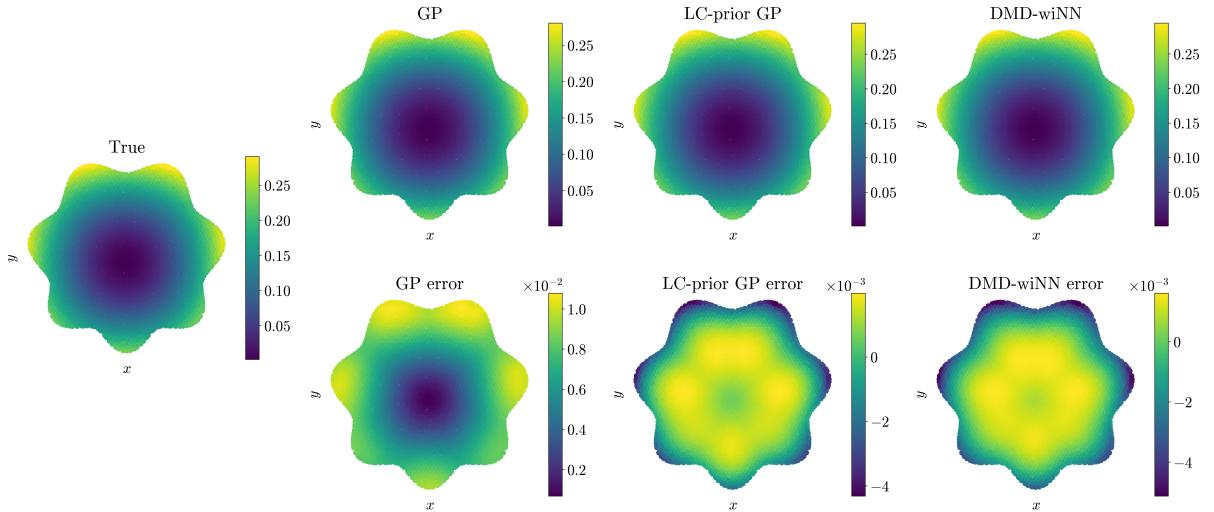


Figure 12: The first column presents the mean solutions of the RBF-FD method at $T=0.1$, while columns two to four display the results at $T=0.1$ for GP, LC-prior GP and DMD-wiNN methods, respectively. The first row shows the mean solutions, and the second row provides the corresponding errors.

4.3.2 Three parameters example

In this subsection, we further test the parameterization problem based on Eq.(14), where the three parameters κ , μ , and ϕ are all uncertain. Thus, $\theta = (\kappa, \mu, \phi)$. For this example, the size of the training set is 8 with evenly scattered 2 points in intervals $\pi_{\text{prior}}(\kappa) \sim \mathcal{U}[-3, 3]$ for κ , $\pi_{\text{prior}}(\phi) \sim \mathcal{U}[-6, 6]$ for ϕ and $\pi_{\text{prior}}(\mu) \sim \mathcal{U}[-10, 10]$ for μ . The size of the testing set is 8000 with evenly scattered 20 points and size of the physics corrected set is 64 with evenly scattered 4 points in the same intervals. Here, the temporal and spatial discretization schemes and the number of POD modes remain the same as in two parameters example, except for the additional parameter μ .

To demonstrate the effectiveness of the physics-informed correction, we selected the sample with the largest error in the physics-constrained set and visualized its correction performance. Figure 11 presents the detailed results: column 1 shows the full-order RBF-FD solution as reference; column 2 displays the standard GP method's prediction, which exhibits poor performance (error $\sim 10^{-2}$) under purely data-driven conditions; column 3 presents our proposed LC-prior GP method and column 4 provides the POD approximation obtained from the full-order solution and basis functions, representing the theoretical accuracy upper bound in our framework. The results clearly show that after optimization with the physics-informed loss function, the prediction accuracy improves significantly, achieving a error of 10^{-5} that approaches the parameterization-determined accuracy limit.

The mean solutions for the numerical reference solution, GP surrogate, LC-prior GP surrogate and DMD-wiNN method at $t = 0.1$ are shown in Fig.12. The LC-prior GP method achieves a error of 10^{-3} , representing an order-of-magnitude improvement over the standard

Table 7: The relative errors of GP method, LC-prior GP method and DMD-wiNN.

	GP	LC-prior GP	DMD-wiNN
\mathbf{u} in the x-direction	0.0309	0.0202	0.0320
\mathbf{u} in the y-direction	0.0265	0.0163	0.0264

GP approach. However, using partial datasets to characterize correction terms across the entire parameter space introduces inherent prediction errors. This explains why individual corrected samples may outperform the aggregated test set results. As quantified in Table.4 through relative error computations at all temporal discretization points, our proposed method maintains best performance even for complex multi-parameter systems with irregular computational domains. The parameter estimation in this example, we follow the same workflow as described in the previous section, with detailed numerical results presented in the accompanying Table.6. Under our proposed framework, the posterior samples for systems with multi-physics coupling and multiple parameters demonstrate more competitive performance compared to purely data-driven surrogate models.

4.4 Incompressible Navier-Stokes model

In practice, many fluid phenomena can be described using the incompressible Navier-Stokes model. Thus, in this subsection, we mainly consider the incompressible Navier-Stokes model, where the parameters are viscosity and density. The incompressible Navier-Stokes model with the Dirichlet boundary condition is as follows:

$$\begin{cases} \mathbf{u}_t + (\mathbf{u} \cdot \nabla) \mathbf{u} - \mu \Delta \mathbf{u} + \frac{\nabla p}{\rho} = 0, & \text{in } (0, T) \times \Omega, \\ \nabla \cdot \mathbf{u} = 0, & \text{in } (0, T) \times \Omega, \\ \mathbf{u}(0, \cdot) = \mathbf{u}_0, \end{cases} \quad (15)$$

where $\Omega \in \mathbb{R}^2$, $\mathbf{u} = [u, v]$ is the velocity, μ is the viscosity, p is the pressure, and ρ is the density. u and v represent the velocity components in the x- and y-directions, respectively. So the Eq.(15) can be rewritten by:

$$\begin{cases} \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} - \mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + \frac{1}{\rho} \frac{\partial p}{\partial x} = 0, \\ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} - \mu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) + \frac{1}{\rho} \frac{\partial p}{\partial y} = 0, \\ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \\ u(x, y, 0) = \mathbf{u}_0(x, y), \\ v(x, y, 0) = \mathbf{u}_0(x, y), \end{cases} \quad \text{in } (0, T) \times \Omega. \quad (16)$$

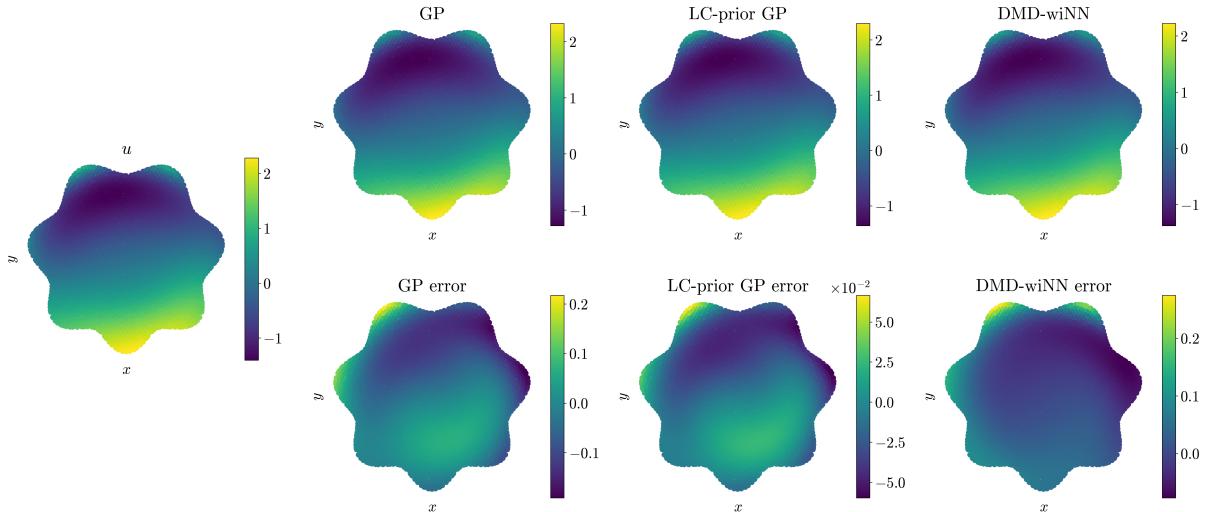


Figure 13: The first column presents the mean solutions of the RBF-FD method at \mathbf{u} in the x-direction and $T=0.1$, while columns two to four display the results at $T=0.1$ for GP, LC-prior GP and DMD-wiNN methods, respectively. The first row shows the mean solutions, and the second row provides the corresponding errors.

For this example, we set $T = 0.1$ and select an irregular region in two-dimensional space. The radius of the region satisfies the same requirement in Eq.(13) of the last Subsection. The uniform division of the DistMesh is used for the spatial division, the computation domain is $[-1.5, 1.5] \times [-1.5, 1.5]$, and the distance between the longest two points after the division is $h = 0.04$. The initial value is given by:

$$\mathbf{u}_0(x, y) = \left[-\pi y \sin\left(\frac{\pi}{2}(x^2 + y^2)\right), \pi x \sin\left(\frac{\pi}{2}(x^2 + y^2)\right) \right]', \quad x, y \in \Omega.$$

The discrete numerical scheme of the temporal direction is as follows:

$$\begin{cases} \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\tau} + (\mathbf{u}^n \cdot \nabla) \mathbf{u}^n - \nu \Delta \mathbf{u}^{n+1} + \frac{\nabla p^{n+1}}{\rho} = 0, \\ \frac{p^{n+1} - p^n}{\tau} + \nabla \cdot \mathbf{u}^{n+1} = 0, \end{cases}$$

Here parameters μ and ρ are uncertainty, $\theta = (\mu, \rho)$. The training set is evenly dispersed by 3 points in intervals $\pi_{\text{prior}}(\mu) \sim \mathcal{U}[0, 1]$ and $\pi_{\text{prior}}(\rho) \sim \mathcal{U}[0.1, 1]$, the physical law corrected set is randomly dispersed by 5 points and the testing set is randomly dispersed by 20 points in the intervals. Thus, the size of the training set is 9, the corrected set contains 16 points (excluding the overlapping 9 points between the corrected set and the training set), and the testing set contains 400 points.

The Navier-Stokes model, as a classical system of partial differential equations governing viscous fluid motion, involve complex coupling among multiple physical quantities. Therefore, in this section, we must simultaneously construct surrogate models for all three solution

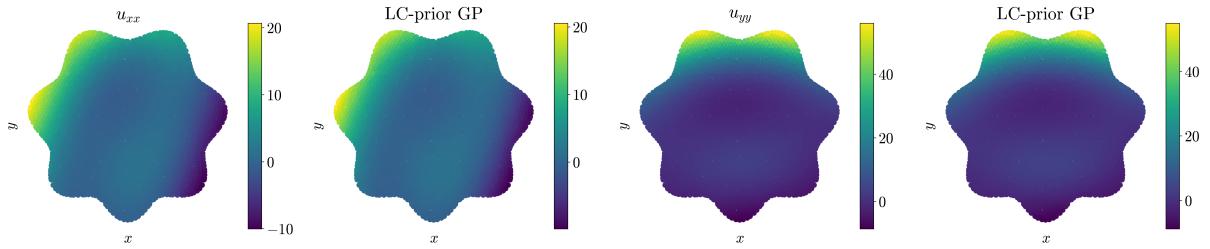


Figure 14: The second derivatives of \mathbf{u} in the x -direction and $T=0.1$ obtained by the RBF-FD(the first and third). The LC-prior GP results are the second one and the fourth one.

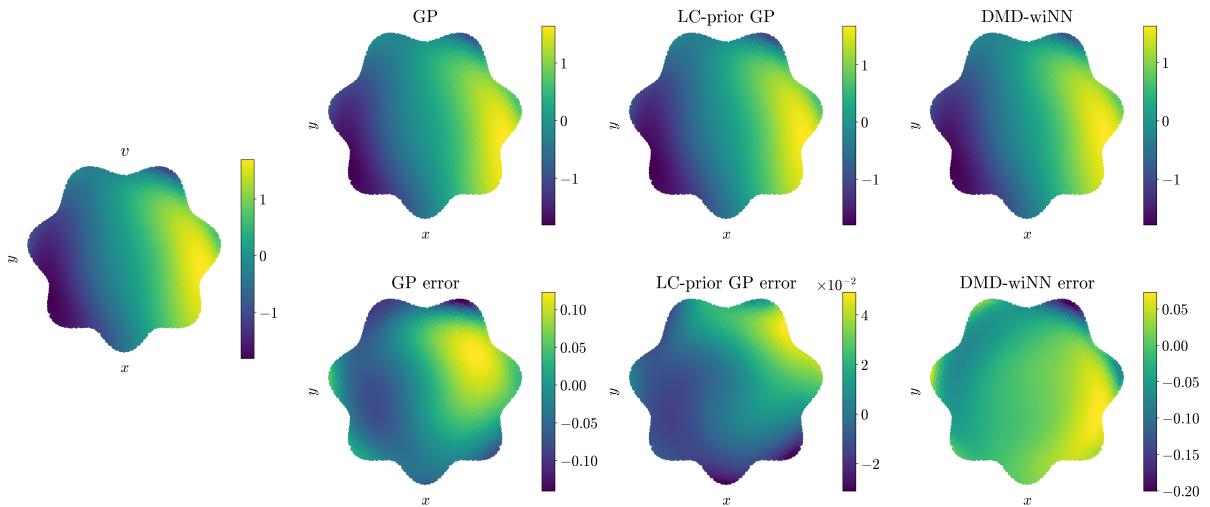


Figure 15: The first column presents the mean solutions of the RBF-FD method at \mathbf{u} in the y -direction and $T=0.1$, while columns two to four display the results at $T=0.1$ for GP, LC-prior GP and DMD-wiNN methods, respectively. The first row shows the mean solutions, and the second row provides the corresponding errors.

fields: the velocity components $u(x, y, t; \theta)$ and $v(x, y, t; \theta)$ in both coordinate directions and the pressure field $p(x, y, t; \theta)$. For each physical quantity, we parameterize the solutions using basis functions constructed from the first 4 POD modes.

In this section, we compute both the mean solution of the velocity field \mathbf{u} on the test set and evaluate its second-order derivative characteristics using the evaluation and differentiation matrices derived from the RBF-FD method. Fig.13 and Fig.15 present the computational results for the x -direction and y -direction at $t = 0.1$ respectively. The first column displays the full-order reference solution obtained through numerical methods, while columns 2 through 4 show the mean solutions generated by the GP, LC-prior GP and DMD-wiNN methods along with their corresponding errors relative to the reference solution. Within this complex system, while the DMD-wiNN method demonstrates significantly increased errors compared to previous cases, our proposed method maintains exceptionally small error magnitudes, conclusively

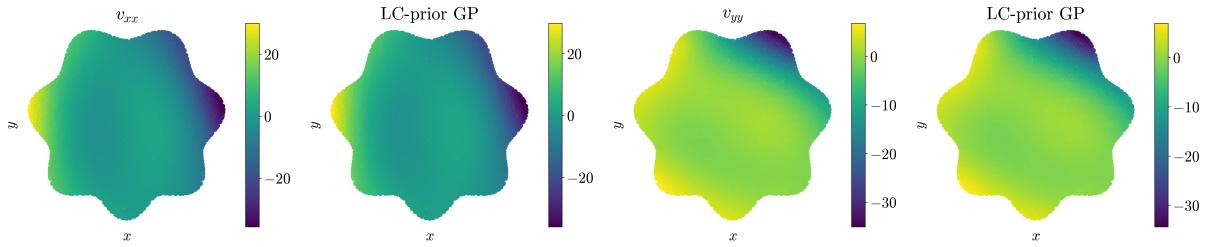


Figure 16: The second derivatives of \mathbf{u} in the y -direction and $T=0.1$ obtained by the RBF-FD(the first and third). The LC-prior GP results are the second one and the fourth one.

demonstrating its generalization capability. Table.7 provides a comprehensive error analysis, quantifying the aggregate relative errors of the surrogate model across all temporal discretization points within $[0, T]$.

Moreover, the precomputed evaluation matrix $E_h(X, Y)$ and differentiation matrices such as $D_h^{\mathcal{L}_x}(X, Y)$ and $D_h^{\mathcal{L}_{xx}}(X, Y)$ obtained during training set generation enable efficient and accurate computation of higher-order derivatives for the target functions. Let us denote the surrogate model predictions by \hat{u} and \hat{v} , so the second-order derivative can be calculated by:

$$\frac{\partial^2 \hat{u}}{\partial x^2} = D_h^{\mathcal{L}_{xx}}(X, Y)E_h^\dagger(Y, X)\hat{u}, \quad \frac{\partial^2 \hat{v}}{\partial x^2} = D_h^{\mathcal{L}_{xx}}(X, Y)E_h^\dagger(Y, X)\hat{v}.$$

Similarly, we can compute $\partial^2 \hat{u}/\partial y^2$ and $\partial^2 \hat{v}/\partial y^2$ in the same manner. Fig.14 and Fig.16 present the second-order derivative results of \mathbf{u} along different spatial directions, where the first and third columns display the full-order reference solutions, while the second and fourth columns show the predictions obtained using the LC-prior GP method. These results demonstrate that the proposed method maintains high accuracy in approximating multiple target quantities, even for complex computational domains with coupled multi-physics interactions.

5 Conclusion

We present a method for improving computation efficiency and accuracy of numerical estimations for both parametric DEs solutions and unknown parameters using a GP surrogate with a physical law-corrected prior. This approach is applicable to both PDEs defined on complex geometries or multi-physics interaction system. The method employs the POD method to map the high-dimensional solution space of the DEs into a low-dimensional basis coefficient space. The GP surrogate is then trained to predict the basis coefficients from the DE parameters. To enhance the model further, we introduce a correction mechanism that incorporates physical law constraints, refining the GP surrogate's prior. This physics-informed machine learning approach leverages both solution training data and the underlying physical model, offering improved performance.

Additionally, we use the learned forward model to infer the posterior distribution of unknown parameters from noisy observations. By combining data-driven modeling with physical law constraints, our method effectively handles complex equations and demonstrates robust performance even in the presence of noisy observations with varying magnitudes.

Despite its strengths, there are limitations and opportunities for future work. First, the current approach samples correction model points uniformly from the coefficient model's samples, whereas a more effective strategy may involve selecting points with high GP surrogate prediction variance in the parameter space. Second, when using POD to parameterize the DE, the number of discrete grids in the predicted DE solution is constrained to the number of grids in the sample data. This limitation is absent when using conventional orthogonal bases. Future work will explore methods to reconstruct more sophisticated discrete solutions within the POD framework.

Acknowledgment

Heng Yong acknowledges the support of National Safety Academic Fund (NSAF) under Grant No.U2230208 and the National Natural Science Foundation of China (NSFC) under Grant Number 12331010. Hongqiao Wang acknowledges the support of National Natural Science Foundation of China (NSFC) under Grant Number 12271562. The work was supported by the Major Scientific and Technological Innovation Platform Project of Human Province (2024JC1003). This work was carried out in part using computing resources at the High Performance Computing Center of Central South University.

References

- [1] WE Schiesser. Time Delay ODE/PDE Models: Applications in Biomedical Science and Engineering. CRC Press, 2019.
- [2] Xinhai Chen, Jie Liu, Shengguo Li, Peizhen Xie, Lihua Chi, and Qinglin Wang. Tamm: A new topology-aware mapping method for parallel applications on the tianhe-2a supercomputer. In Algorithms and Architectures for Parallel Processing: 18th International Conference, ICA3PP 2018, Guangzhou, China, November 15–17, 2018, Proceedings, Part I 18, pages 242–256. Springer, 2018.
- [3] Ameya D Jagtap, Ehsan Kharazmi, and George Em Karniadakis. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. Computer Methods in Applied Mechanics and Engineering, 365:113028, 2020.

- [4] Guofei Pang and George Em Karniadakis. Physics-informed learning machines for partial differential equations: Gaussian processes versus neural networks. *Emerging frontiers in nonlinear science*, pages 323–343, 2020.
- [5] John David Anderson and John Wendt. *Computational fluid dynamics*, volume 206. Springer, 1995.
- [6] Siddhartha Mishra. A machine learning framework for data driven acceleration of computations of differential equations. *arXiv preprint arXiv:1807.09519*, 2018.
- [7] Pau Batlle, Matthieu Darcy, Bamdad Hosseini, and Houman Owhadi. Kernel methods are competitive for operator learning. *Journal of Computational Physics*, 496:112549, 2024.
- [8] Carlos Mora, Amin Yousefpour, Shirin Hosseini, Houman Owhadi, and Ramin Bostanabad. Operator learning with gaussian processes. *Computer Methods in Applied Mechanics and Engineering*, 434:117581, 2025.
- [9] Samuel Rudy, Alessandro Alla, Steven L Brunton, and J Nathan Kutz. Data-driven identification of parametric partial differential equations. *SIAM Journal on Applied Dynamical Systems*, 18(2):643–660, 2019.
- [10] Tapas Tripura and Souvik Chakraborty. Wavelet neural operator for solving parametric partial differential equations in computational mechanics problems. *Computer Methods in Applied Mechanics and Engineering*, 404:115783, 2023.
- [11] Sifan Wang, Hanwen Wang, and Paris Perdikaris. Learning the solution operator of parametric partial differential equations with physics-informed deeponets. *Science advances*, 7(40):eabi8605, 2021.
- [12] Marc C Kennedy and Anthony O’Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001.
- [13] Xinhai Chen, Rongliang Chen, Qian Wan, Rui Xu, and Jie Liu. An improved data-free surrogate model for solving partial differential equations using deep neural networks. *Scientific reports*, 11(1):19507, 2021.
- [14] Majdi I Radaideh and Tomasz Kozlowski. Surrogate modeling of advanced computer simulations using deep gaussian processes. *Reliability Engineering & System Safety*, 195:106731, 2020.
- [15] Raphaël Pestourie, Youssef Mroueh, Chris Rackauckas, Payel Das, and Steven G Johnson. Physics-enhanced deep surrogates for partial differential equations. *Nature Machine Intelligence*, 5(12):1458–1465, 2023.

- [16] Yongchao Li, Yanyan Wang, and Liang Yan. Surrogate modeling for bayesian inverse problems based on physics-informed neural networks. *Journal of Computational Physics*, 475:111841, 2023.
- [17] David J Lucia, Philip S Beran, and Walter A Silva. Reduced-order modeling: new approaches for computational physics. *Progress in aerospace sciences*, 40(1-2):51–117, 2004.
- [18] Olivier Ezvan, Anas Batou, Christian Soize, and Laurent Gagliardini. Multilevel model reduction for uncertainty quantification in computational structural dynamics. *Computational Mechanics*, 59(2):219–246, 2017.
- [19] Jan Hesthaven, Cecilia Pagliantini, and Nicolò Ripamonti. Adaptive symplectic model order reduction of parametric particle-based vlasov–poisson equation. *Mathematics of Computation*, 93(347):1153–1202, 2024.
- [20] Federico Pichi, Beatriz Moya, and Jan S Hesthaven. A graph convolutional autoencoder approach to model order reduction for parametrized pdes. *Journal of Computational Physics*, 501:112762, 2024.
- [21] Akhil Nekkanti and Oliver T Schmidt. Gappy spectral proper orthogonal decomposition. *Journal of Computational Physics*, 478:111950, 2023.
- [22] Stefania Fresca and Andrea Manzoni. Pod-dl-rom: Enhancing deep learning-based reduced order models for nonlinear parametrized pdes by proper orthogonal decomposition. *Computer Methods in Applied Mechanics and Engineering*, 388:114181, 2022.
- [23] JP Moitinho de Almeida. A basis for bounding the errors of proper generalised decomposition solutions in solid mechanics. *International Journal for Numerical Methods in Engineering*, 94(10):961–984, 2013.
- [24] Francisco Chinesta, Amine Ammar, Adrien Leygue, and Roland Keunings. An overview of the proper generalized decomposition with applications in computational rheology. *Journal of Non-Newtonian Fluid Mechanics*, 166(11):578–592, 2011.
- [25] Bruce Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE transactions on automatic control*, 26(1):17–32, 1981.
- [26] Ulrike Baur, Christopher Beattie, Peter Benner, and Serkan Gugercin. Interpolatory projection methods for parameterized model reduction. *SIAM Journal on Scientific Computing*, 33(5):2489–2518, 2011.
- [27] Jan S Hesthaven and Stefano Ubbiali. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *Journal of Computational Physics*, 363:55–78, 2018.

- [28] Kyung Hyun Park, Sang Ook Jun, Sung Min Baek, Maeng Hyo Cho, Kwan Jung Yee, and Dong Ho Lee. Reduced-order model with an artificial neural network for aerostructural design optimization. *Journal of Aircraft*, 50(4):1106–1116, 2013.
- [29] Huailing Song, Yuming Ba, Dongqin Chen, and Qiuqi Li. A model reduction method for parametric dynamical systems defined on complex geometries. *Journal of Computational Physics*, 506:112923, 2024.
- [30] Victor Bayona, Miguel Moscoso, Manuel Carretero, and Manuel Kindelan. Rbf-fd formulas and convergence properties. *Journal of Computational Physics*, 229(22):8281–8295, 2010.
- [31] Varun Shankar. The overlapped radial basis function-finite difference (rbf-fd) method: A generalization of rbf-fd. *Journal of Computational Physics*, 342:211–228, 2017.
- [32] Victor Bayona, Natasha Flyer, Bengt Fornberg, and Gregory A Barnett. On the role of polynomials in rbf-fd approximations: II. numerical solution of elliptic pdes. *Journal of Computational Physics*, 332:257–273, 2017.
- [33] Robert Schaback and Holger Wendland. Using compactly supported radial basis functions to solve partial differential equations. *WIT Transactions on Modelling and Simulation*, 23, 2024.
- [34] Gal Berkooz, Philip Holmes, and John L Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual review of fluid mechanics*, 25(1):539–575, 1993.
- [35] Ngoc Cuong Nguyen and Andrew Rohskopf. Proper orthogonal descriptors for efficient and accurate interatomic potentials. *Journal of Computational Physics*, 480:112030, 2023.
- [36] Robert M Miura. The korteweg–devries equation: a survey of results. *SIAM review*, 18(3):412–459, 1976.