SIAM J. SCI. COMPUT.
Vol. 42, No. 1, pp. A220–A249

WHEN BIFIDELITY MEETS COKRIGING: AN EFFICIENT PHYSICS-INFORMED MULTIFIDELITY METHOD*

XIU YANG†, XUEYU ZHU‡, AND JING LI§

Abstract. In this work, we propose a framework that combines approximation-theory-based multifidelity method and Gaussian-process-regression-based multifidelity method to achieve data-model convergence when stochastic simulation models and sparse accurate observation data are available. Specifically, the two types of multifidelity methods we use are the *bifidelity* method and the *cokriging* method. The new approach uses the bifidelity method to efficiently estimate the empirical mean and covariance of the stochastic simulation outputs, then it uses these statistics to construct Gaussian process (GP) representing low-fidelity in cokriging. We also combine the bifidelity method with kriging, where the approximated empirical statistics are used to construct the GP as well. We prove that the resulting posterior mean by the new physics-informed approach preserves linear physical constraints up to an error bound. We present numerical examples to demonstrate that using this method, we can obtain accurate construction of the state of interest based on partially a correct physical model and a few accurate observations.

Key words. physics-informed, Gaussian process regression, cokriging, multifidelity, error bound

AMS subject classifications. 65C60, 42C05, 41A10

DOI. 10.1137/18M1231353

1. Introduction.

Gaussian process (GP), a widely used tool in statistics and machine learning [7, 37, 40], has become popular in probabilistic scientific computing. GP regression (GPR), also known as *kriging* in geostatistics, constructs a statistical model of a partially observed process by assuming that its observations are a realization of a GP. A GP is uniquely described by its mean and covariance function (also known as *kernel*). Its variant, cokriging, was originally formulated to compute predictions of sparsely observed states of physical systems by leveraging observations of other states or parameters of the system [39, 18]. Recently, it has been employed for constructing multifidelity models [16, 22, 34] and has been applied in various fields, e.g., [21, 1, 30]. In the widely used stationary kriging/cokriging method, usually parameterized forms of mean and covariance functions are assumed, and the hyperparameters of these functions (e.g., variance and correlation length) are estimated by maximizing the log marginal likelihood.

Recently a new framework, physics-informed kriging (PhIK)/physics-informed cokriging (CoPhIK), was developed for the applications where partially correct physical models along with sparse observation data are available [49, 46]. These physical

*Submitted to the journal's Methods and Algorithms for Scientific Computing section December 7, 2018; accepted for publication (in revised form) October 1, 2019; published electronically January 14, 2020.

https://doi.org/10.1137/18M1231353

Funding: The work of the first and third authors was supported by the U.S. Department of Energy (DOE), Office of Science, Office of Advanced Scientific Computing Research (ASCR) through contract DE-AC05-76RL01830. The work of the second author was supported by the Simons Foundation.

†Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA 18015 (xiy518@lehigh.edu).

‡Corresponding author. Department of Mathematics, The University of Iowa, Iowa City, IA 52242 (xueyu-zhu@uiowa.edu).

§Advanced Computing, Mathematics and Data Division, Pacific Northwest National Laboratory, Richland, WA 99352 (jing.li@pnnl.gov).

models are constructed based on domain knowledge, and they include random variables or random fields to represent the lack of knowledge (e.g., unknown physical law, uncertain parameters, etc). The PhIK/CoPhIK framework combines the realizations of the stochastic physical model with the observation data to provide accurate reconstruction of the state of interest on the entire computational domain. These realizations are then used to approximate mean and covariance in the PhIK/CoPhIK framework. The most popular approach of obtaining model realizations is the Monte Carlo (MC) simulation. Other options to estimate mean and covariance include quasi-MC [29], probabilistic collocation [42, 45], analysis of variance [24, 47], compressive sensing [4, 48], and the moment equation method [41].

It is worthy to note that the aforementioned approaches rely on a single fidelity solver. For large scale applications, the computational cost can still be prohibitive if a large number of samples are required. In many practical problems, low-fidelity models for the underlying problem are often available, and it is more inexpensive to obtain the realizations of these models. Even though their accuracy is not high, they can still capture some important physics of the underlying models with low computational cost. Therefore, it is highly desirable to utilize the computational efficiency of low-fidelity models to reduce the overall computational cost. Many multifidelity algorithms have been developed based on different principles in different contexts. These include (a) multilevel MC [10, 2], which is already used in PhIK and CoPhIK to reduce the computational cost [49, 46]; (b) meta-models through GP, i.e., cokriging [16, 43, 33]; (c) variance reduced based approaches, i.e., control-variate based approach [32], importance sampling [31]; (d) model discrepancy based approaches [28, 5]. Another trend in the context of uncertainty quantification is to explore the parameter space by a large number of low-fidelity samples and identify a small set of important basis, then learn the "best" approximation rule of the target high-fidelity solution or its statistics based on the selected basis [27, 51, 50, 6]. Previous works demonstrated its potential to significantly reduce the computational cost for various applications by utilizing $\mathcal{O}(10)$ high-fidelity simulations, including combustion modeling [26], orbit-state uncertainty propagation [14], molecular dynamics simulations [36], and turbulence modeling [13], to name a few.

In this work, we propose to employ the multifidelity approaches presented in [27, 51] to reduce the computational cost of PhIK and CoPhIK by reducing the number of high-fidelity/high-resolution simulations. For demonstration purpose, we consider bifidelity model in this work, and the proposed framework can be implemented in multifidelity (more than two fidelity) models. Besides, we also establish an analysis framework for bifidelity-based PhIK and CoPhIK, which is applicable to not only MC based methods to compute GP moments but also other sampling-based methods.

2. Methodology.

This section begins by reviewing the general GPR framework [44], the kriging and cokriging methods with stationary kernel [7], the PhIK [49] and CoPhIK [46] methods, and the bifidelity approach [51]. Then, we introduce the bifidelity-aided PhIK and CoPhIK methods.

2.1. GPR framework.

We denote the observation locations as $\boldsymbol{X} = \{\boldsymbol{x}^{(i)}\}_{i=1}^{N}$ ($\boldsymbol{x}^{(i)}$ are d-dimensional vectors in $D \subseteq \mathbb{R}^d$) and the observed state values at these locations as $\boldsymbol{y} = (y^{(1)}, y^{(2)}, \ldots, y^{(N)})^\top$ ($y^{(i)} \in \mathbb{R}$). For simplicity, we assume that $y^{(i)}$ are scalars. We aim to predict y at any new location $\boldsymbol{x}^* \in D$. The GPR method assumes that the observation vector $\boldsymbol{y}$ is a realization of the following N-dimensional random vector that satisfies multivariate Gaussian distribution:

$$\boldsymbol{Y} = \left(Y(\boldsymbol{x}^{(1)}), Y(\boldsymbol{x}^{(2)}), \ldots, Y(\boldsymbol{x}^{(N)})\right)^\top,$$

where $Y(\boldsymbol{x}^{(i)})$ is the concise notation of $Y(\boldsymbol{x}^{(i)};\omega)$ and $Y(\boldsymbol{x}^{(i)};\omega)$ is a Gaussian random variable defined on a probability space $(\Omega,\mathcal{F},P)$ with $\omega\in\Omega$. Of note, $\boldsymbol{x}^{(i)}$ can be considered as parameters for the GP $Y(\cdot,\cdot):D\times\Omega\to\mathbb{R}$ such that $Y(\boldsymbol{x}^{(i)},\cdot):\Omega\to\mathbb{R}$ is a Gaussian random variable for any $\boldsymbol{x}^{(i)}$ in the set D. Usually, the GP $Y(\boldsymbol{x})$ is denoted as

$$Y(\boldsymbol{x})\sim\mathcal{GP}\left(\mu(\boldsymbol{x}),k(\boldsymbol{x},\boldsymbol{x}')\right), \tag{2.1}$$

where $\mu(\cdot):D\to\mathbb{R}$ and $k(\cdot,\cdot):D\times D\to\mathbb{R}$ are the mean and covariance functions:

$$\mu(\boldsymbol{x})=\mathbb{E}\left\{Y(\boldsymbol{x})\right\}, \tag{2.2}$$

$$k(\boldsymbol{x},\boldsymbol{x}')=\mathrm{Cov}\left\{Y(\boldsymbol{x}),Y(\boldsymbol{x}')\right\}=\mathbb{E}\left\{(Y(\boldsymbol{x})-\mu(\boldsymbol{x}))(Y(\boldsymbol{x}')-\mu(\boldsymbol{x}'))\right\}. \tag{2.3}$$

The variance of $Y(\boldsymbol{x})$ is $k(\boldsymbol{x},\boldsymbol{x})$, and its standard deviation is $\sigma(\boldsymbol{x})=\sqrt{k(\boldsymbol{x},\boldsymbol{x})}$. The covariance matrix of random vector $\boldsymbol{Y}$, denoted as $\mathbf{C}$, is defined as $C_{ij}=k(\boldsymbol{x}^{(i)},\boldsymbol{x}^{(j)})$. Functions $\mu(\boldsymbol{x})$ and $k(\boldsymbol{x},\boldsymbol{x}')$ are obtained by identifying their hyperparameters via maximizing the log marginal likelihood [44]:

$$\ln L=-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu})^{\top}\mathbf{C}^{-1}(\boldsymbol{y}-\boldsymbol{\mu})-\frac{1}{2}\ln|\mathbf{C}|-\frac{N}{2}\ln 2\pi, \tag{2.4}$$

where $\boldsymbol{\mu}=(\mu(\boldsymbol{x}^{(1)}),\ldots,\boldsymbol{x}^{(N)})^{\top}$. The result of GPR is a posterior distribution $y(\boldsymbol{x}^*)\sim\mathcal{N}(\hat{y}(\boldsymbol{x}^*),\hat{s}^2(\boldsymbol{x}^*))$ for any $\boldsymbol{x}^*\in D$, where

$$\hat{y}(\boldsymbol{x}^*)=\mu(\boldsymbol{x}^*)+\boldsymbol{c}(\boldsymbol{x}^*)^{\top}\mathbf{C}^{-1}(\boldsymbol{y}-\boldsymbol{\mu}), \tag{2.5}$$

$$\hat{s}^2(\boldsymbol{x}^*)=\sigma^2(\boldsymbol{x}^*)-\boldsymbol{c}(\boldsymbol{x}^*)^{\top}\mathbf{C}^{-1}\boldsymbol{c}(\boldsymbol{x}^*), \tag{2.6}$$

and $\boldsymbol{c}(\boldsymbol{x}^*)$ is a vector of covariance, i.e., $(\boldsymbol{c}(\boldsymbol{x}^*))_i=k(\boldsymbol{x}^{(i)},\boldsymbol{x}^*)$. In practice, it is common to use $\hat{y}(\boldsymbol{x}^*)$ as the prediction, and $\hat{s}^2(\boldsymbol{x}^*)$ is also called mean squared error (MSE) of the prediction because $\hat{s}^2(\boldsymbol{x}^*)=\mathbb{E}\left\{(\hat{y}(\boldsymbol{x}^*)-Y(\boldsymbol{x}^*))^2\right\}$ [7]. Consequently, $\hat{s}(\boldsymbol{x}^*)$ is the root MSE (RMSE). Moreover, to account for the observation noise, one can assume that the noise is independent and identically distributed (i.i.d.) Gaussian random variables with zero mean and variance δ^2, and replace $\mathbf{C}$ with $\mathbf{C}+\delta^2\mathbf{I}$. In this study, we assume that observations $\boldsymbol{y}$ are noiseless. If $\mathbf{C}$ is not invertible or its condition number is very large, one can add a small regularization term $\alpha\mathbf{I}$ (α is a small positive real number) to $\mathbf{C}$, which is equivalent to assuming there is an observation noise. In addition, $\hat{s}$ can be used in global optimization or in the greedy algorithm to identify locations of additional observations. Specifically, in the greedy algorithm, the new observations can be added at the maxima of $\hat{s}$; see Appendix A for details.

2.2. Kriging and cokriging with stationary kernels. In the widely used ordinary kriging method, a stationary GP is assumed [17]. Specifically, μ is set as a constant $\mu(\boldsymbol{x})\equiv\mu$, and $k(\boldsymbol{x},\boldsymbol{x}')=k(\boldsymbol{\tau})$, where $\boldsymbol{\tau}=\boldsymbol{x}-\boldsymbol{x}'$. Consequently, $\sigma^2(\boldsymbol{x})=k(\boldsymbol{x},\boldsymbol{x})=k(\mathbf{0})=\sigma^2$ is a constant. Popular forms of kernels include polynomial, exponential, Gaussian (squared-exponential), and Matérn functions. For example, the Gaussian kernel can be written as $k(\boldsymbol{\tau})=\sigma^2\exp\left(-\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{x}'\|_w^2\right)$, where the weighted norm is defined as $\|\boldsymbol{x}-\boldsymbol{x}'\|_w^2=\sum_{i=1}^{d}\left(\frac{x_i-x_i'}{l_i}\right)^2$. Here, l_i $(i=1,\ldots,d)$, the correlation

lengths of $\boldsymbol{y}$ in the i direction, are constants. In practice, the kernels are selected based on the properties of the system, such as regularity, periodicity, and boundary condition. Given a stationary covariance function, the covariance matrix $\mathbf{C}$ of $\boldsymbol{Y}$ can be written as $\mathbf{C} = \sigma^2\boldsymbol{\Psi}$, where $\Psi_{ij} = \exp(-\frac{1}{2}\|\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}\|_w^2)$. In the maximum likelihood estimation framework, the estimators of μ and σ^2, denoted as $\hat{\mu}$ and $\hat{\sigma}^2$, are

$$\hat{\mu} = \frac{\mathbf{1}^\top\boldsymbol{\Psi}^{-1}\boldsymbol{y}}{\mathbf{1}^\top\boldsymbol{\Psi}^{-1}\mathbf{1}}, \qquad \hat{\sigma}^2 = \frac{(\boldsymbol{y} - \mathbf{1}\mu)^\top\boldsymbol{\Psi}^{-1}(\boldsymbol{y} - \mathbf{1}\mu)}{N}, \tag{2.7}$$

where $\mathbf{1}$ is a constant vector and consists of 1 [7]. The hyperparameters σ and l_i are estimated by maximizing the log marginal likelihood (2.4). The $\hat{y}(\boldsymbol{x}^*)$ and $\hat{s}^2(\boldsymbol{x}^*)$ in (2.5) take the following form:

$$\hat{y}(\boldsymbol{x}^*) = \hat{\mu} + \boldsymbol{\psi}^\top\boldsymbol{\Psi}^{-1}(\boldsymbol{y} - \mathbf{1}\hat{\mu}), \tag{2.8}$$

$$\hat{s}^2(\boldsymbol{x}^*) = \hat{\sigma}^2\left(1 - \psi^\top\boldsymbol{\Psi}^{-1}\boldsymbol{\psi}\right), \tag{2.9}$$

where $\boldsymbol{\psi} = \boldsymbol{\psi}(\boldsymbol{x}^*)$ is a vector of correlations between the observed data and the prediction, i.e., $\psi_i = \frac{1}{\sigma^2}k(\boldsymbol{x}^{(i)}, \boldsymbol{x}^*)$.

Next, we briefly review the formulation of cokriging for two-level multifidelity modeling. Suppose that we have high-fidelity data (e.g., accurate measurements of states) $\boldsymbol{y}_H = (y_H^{(1)}, \ldots, y_H^{(N_H)})^\top$ at locations $\boldsymbol{X}_H = \{\boldsymbol{x}_H^{(i)}\}_{i=1}^{N_H}$ and low-fidelity data (e.g., simulation results) $\boldsymbol{y}_L = (y_L^{(1)}, \ldots, y_L^{(N_L)})^\top$ at locations $\boldsymbol{X}_L = \{\boldsymbol{x}_L^{(i)}\}_{i=1}^{N_L}$, where $y_H^{(i)}, y_L^{(i)} \in \mathbb{R}$ and $\boldsymbol{x}_H^{(i)}, \boldsymbol{x}_L^{(i)} \in D \subseteq \mathbb{R}^d$. We denote $\widetilde{\boldsymbol{X}} = \{\boldsymbol{X}_L, \boldsymbol{X}_H\}$ and $\widetilde{\boldsymbol{y}} = \left(\boldsymbol{y}_L^\top, \boldsymbol{y}_H^\top\right)^\top$.

Kennedy and O'Hagan [16] proposed a multifidelity formulation based on the autoregressive model for GP, Y_H $(\sim \mathcal{GP}(\mu_H(\cdot), k_H(\cdot,\cdot)))$:

$$Y_H(\boldsymbol{x}) = \rho Y_L(\boldsymbol{x}) + Y_d(\boldsymbol{x}), \tag{2.10}$$

where $Y_L(\cdot)$ $(\sim \mathcal{GP}(\mu_L(\cdot), k_L(\cdot,\cdot)))$ regresses the low-fidelity data, $\rho \in \mathbb{R}$ is a regression parameter, and $Y_d(\cdot)$ $(\sim \mathcal{GP}(\mu_d(\cdot), k_d(\cdot,\cdot))$ models the difference between Y_H and ρY_L. This model assumes that

$$\mathrm{Cov}\left\{Y_H(\boldsymbol{x}), Y_L(\boldsymbol{x}') \mid Y_L(\boldsymbol{x})\right\} = 0 \quad \text{for all} \quad \boldsymbol{x}' \neq \boldsymbol{x},\ \boldsymbol{x}, \boldsymbol{x}' \in D. \tag{2.11}$$

The covariance of observations, $\widetilde{\mathbf{C}}$, is then given by

$$\widetilde{\mathbf{C}} = \begin{pmatrix} \mathbf{C}_L(\boldsymbol{X}_L, \boldsymbol{X}_L) & \rho\mathbf{C}_L(\boldsymbol{X}_L, \boldsymbol{X}_H) \\ \rho\mathbf{C}_L(\boldsymbol{X}_H, \boldsymbol{X}_L) & \rho^2\mathbf{C}_L(\boldsymbol{X}_H, \boldsymbol{X}_H) + \mathbf{C}_d(\boldsymbol{X}_H, \boldsymbol{X}_H) \end{pmatrix}, \tag{2.12}$$

where $\mathbf{C}_L$ and $\mathbf{C}_d$ are the covariance matrices computed from $k_L(\cdot,\cdot)$ and $k_d(\cdot,\cdot)$, respectively. One can assume parameterized forms for these kernels (e.g., Gaussian kernel) and employ the following two-step approach [8, 7] to identify hyperparameters:

1. Use kriging to construct Y_L using $\{\boldsymbol{X}_L, \boldsymbol{y}_L\}$.
2. Denote $\boldsymbol{y}_d = \boldsymbol{y}_H - \rho\boldsymbol{y}_L(\boldsymbol{X}_H)$, where $\boldsymbol{y}_L(\boldsymbol{X}_H)$ are the values of $\boldsymbol{y}_L$ at locations common to those of $\boldsymbol{X}_H$; then construct Y_d using $\{\boldsymbol{X}_H, \boldsymbol{y}_d\}$ via kriging.

The posterior mean and variance of Y_H at $\boldsymbol{x}^* \in D$ are given by

$$\hat{y}(\boldsymbol{x}^*) = \mu_H(\boldsymbol{x}^*) + \widetilde{\boldsymbol{c}}(\boldsymbol{x}^*)^\top\widetilde{\mathbf{C}}^{-1}(\widetilde{\boldsymbol{y}} - \widetilde{\boldsymbol{\mu}}), \tag{2.13}$$

$$\hat{s}^2(\boldsymbol{x}^*) = \rho^2\sigma_L^2(\boldsymbol{x}^*) + \sigma_d^2(\boldsymbol{x}^*) - \widetilde{\boldsymbol{c}}(\boldsymbol{x}^*)^\top \widetilde{\mathbf{C}}^{-1}\widetilde{\boldsymbol{c}}(\boldsymbol{x}^*), \tag{2.14}$$

where $\mu_H(\boldsymbol{x}^*) = \rho\mu_L(\boldsymbol{x}^*) + \mu_d(\boldsymbol{x}^*)$, $\sigma_L^2(\boldsymbol{x}^*) = k_{_L}(\boldsymbol{x}^*, \boldsymbol{x}^*)$, $\sigma_d^2(\boldsymbol{x}^*) = k_d(\boldsymbol{x}^*, \boldsymbol{x}^*)$, and

$$\widetilde{\boldsymbol{\mu}} = \begin{pmatrix} \boldsymbol{\mu}_L \\ \boldsymbol{\mu}_H \end{pmatrix} = \begin{pmatrix} \left(\mu_L(\boldsymbol{x}_{_L}^{(1)}) \ldots, \mu_L(\boldsymbol{x}_{_L}^{(N_L)})\right)^\top \\ \left(\mu_H(\boldsymbol{x}_{_H}^{(1)}) \ldots, \mu_H(\boldsymbol{x}_{_H}^{(N_H)})\right)^\top \end{pmatrix}, \tag{2.15}$$

$$\widetilde{\boldsymbol{c}}(\boldsymbol{x}^*) = \begin{pmatrix} \rho\boldsymbol{c}_L(\boldsymbol{x}^*) \\ \boldsymbol{c}_H(\boldsymbol{x}^*) \end{pmatrix} = \begin{pmatrix} \left(\rho k_L(\boldsymbol{x}^*, \boldsymbol{x}_{_L}^{(1)}), \ldots, \rho k_{_L}(\boldsymbol{x}^*, \boldsymbol{x}_{_L}^{(N_L)})\right)^\top \\ \left(k_H(\boldsymbol{x}^*, \boldsymbol{x}_{_H}^{(1)}), \ldots, k_H(\boldsymbol{x}^*, \boldsymbol{x}_{_H}^{(N_H)})\right)^\top \end{pmatrix}, \tag{2.16}$$

where $k_H(\boldsymbol{x}, \boldsymbol{x}') = \rho^2 k_L(\boldsymbol{x}, \boldsymbol{x}') + k_d(\boldsymbol{x}, \boldsymbol{x}')$. Here, we have neglected a small contribution to $\hat{s}^2$ (see [7]). Alternatively, one can simultaneously identify hyperparameters in $k_L(\cdot,\cdot)$ and $k_d(\cdot,\cdot)$ along with ρ by maximizing the following log marginal likelihood:

$$\ln\widetilde{L} = -\frac{1}{2}(\widetilde{\boldsymbol{y}} - \widetilde{\boldsymbol{\mu}})^\top \widetilde{\mathbf{C}}^{-1}(\widetilde{\boldsymbol{y}} - \widetilde{\boldsymbol{\mu}}) - \frac{1}{2}\ln\left|\widetilde{\mathbf{C}}\right| - \frac{N_H + N_L}{2}\ln 2\pi. \tag{2.17}$$

2.3. PhIK and CoPhIK. The recently proposed PhIK method [49] takes advantage of the existing domain knowledge, e.g., approximate numerical or analytical physics-based models, in the form of realizations of a stochastic model of the system. Consequently, the mean and covariance of the GP model can be approximated using these realizations. As such, there is no need to assume a specific form of the covariance function and solve an optimization problem for the hyperparameters. These stochastic models typically include random parameters or random processes/fields to reflect the lack of understanding (of physical laws) or knowledge (of the coefficients, parameters, etc.) of the real system. Then, MC simulations can be conducted to generate an ensemble of state of interest, from which the statistics, e.g., mean and standard deviation, are estimated.

Specifically, assume that we have M realizations of a stochastic model $u(\boldsymbol{x};\omega)$ ($\boldsymbol{x} \in D, \omega \in \Omega$) denoted as $\{u^m(\boldsymbol{x})\}_{m=1}^M$, and we can build the following GP model:

$$Y(\boldsymbol{x}) \sim \mathcal{GP}(\mu_{\mathrm{MC}}(\boldsymbol{x}), k_{\mathrm{MC}}(\boldsymbol{x}, \boldsymbol{x}')), \tag{2.18}$$

where

$$\begin{aligned} \mu(\boldsymbol{x}) &\approx \mu_{\mathrm{MC}}(\boldsymbol{x}) = \frac{1}{M}\sum_{m=1}^{M} u^m(\boldsymbol{x}), \\ k(\boldsymbol{x}, \boldsymbol{x}') &\approx k_{\mathrm{MC}}(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{M-1}\sum_{m=1}^{M} \left(u^m(\boldsymbol{x}) - \mu_{\mathrm{MC}}(\boldsymbol{x})\right)\left(u^m(\boldsymbol{x}') - \mu_{\mathrm{MC}}(\boldsymbol{x}')\right). \end{aligned} \tag{2.19}$$

Thus, the covariance matrix of $\boldsymbol{Y}$ can be estimated as

$$\mathbf{C} \approx \mathbf{C}_{\mathrm{MC}} = \frac{1}{M-1}\sum_{m=1}^{M} \left(\boldsymbol{u}^m - \boldsymbol{\mu}_{\mathrm{MC}}\right)\left(\boldsymbol{u}^m - \boldsymbol{\mu}_{\mathrm{MC}}\right)^\top, \tag{2.20}$$

where $\boldsymbol{u}^m = \left(u^m(\boldsymbol{x}^{(1)}), \ldots, u^m(\boldsymbol{x}^{(N)})\right)^\top$, $\boldsymbol{\mu}_{\mathrm{MC}} = \left(\mu_{\mathrm{MC}}(\boldsymbol{x}^{(1)}), \ldots, \mu_{\mathrm{MC}}(\boldsymbol{x}^{(N)})\right)^\top$. The prediction and MSE at location $\boldsymbol{x}^* \in D$ are

$$\hat{y}(\boldsymbol{x}^*) = \mu_{\mathrm{MC}}(\boldsymbol{x}^*) + \boldsymbol{c}_{\mathrm{MC}}(\boldsymbol{x}^*)^\top \mathbf{C}_{\mathrm{MC}}^{-1}(\boldsymbol{y} - \boldsymbol{\mu}_{\mathrm{MC}}), \tag{2.21}$$

$$\hat{s}^2(\boldsymbol{x}^*) = \hat{\sigma}_{\mathrm{MC}}^2(\boldsymbol{x}^*) - \boldsymbol{c}_{\mathrm{MC}}(\boldsymbol{x}^*)^\top \mathbf{C}_{\mathrm{MC}}^{-1}\boldsymbol{c}_{\mathrm{MC}}(\boldsymbol{x}^*), \tag{2.22}$$

where $\hat{\sigma}^2_{\mathrm{MC}}(\boldsymbol{x}^*) = k_{\mathrm{MC}}(\boldsymbol{x}^*, \boldsymbol{x}^*)$ is the variance of data set $\{u^m(\boldsymbol{x}^*)\}_{m=1}^M$ and $\boldsymbol{c}_{\mathrm{MC}}(\boldsymbol{x}^*) = \big(k_{\mathrm{MC}}(\boldsymbol{x}^{(1)}, \boldsymbol{x}^*), \ldots, k_{\mathrm{MC}}(\boldsymbol{x}^{(N)}, \boldsymbol{x}^*)\big)^\top$.

Similarly, the CoPhIK method uses model realizations to construct Y_L in cokriging by setting $\mu_L(\boldsymbol{x}) = \mu_{\mathrm{MC}}(\boldsymbol{x})$ and $k_L(\boldsymbol{x}, \boldsymbol{x}') = k_{\mathrm{MC}}(\boldsymbol{x}, \boldsymbol{x}')$ to construct Y_L, where μ_{MC} and k_{MC} are given by (2.19). The GP model Y_d is constructed using the same approach as in the second step of the Kennedy and O'Hagan cokriging framework. We set $\boldsymbol{y}_d = \boldsymbol{y}_H - \rho\boldsymbol{\mu}_L(\boldsymbol{X}_H)$. The reason for this choice is that $\mu_L(\boldsymbol{X}_H)$ is the most probable observation of the GP Y_L. Next, we need to assume a specific form of the kernel function. Without loss of generality, we use the stationary Gaussian kernel model and constant μ_d. Once $\boldsymbol{y}_d$ is computed and the form of $\mu_d(\cdot)$ and $k_d(\cdot,\cdot)$ are decided, Y_d can be constructed as in ordinary kriging. Now all components of $\ln\widetilde{L}$ in (2.17) are specified except for the $\boldsymbol{y}_L$ in $\widetilde{\boldsymbol{y}}$. Here we set $\boldsymbol{X}_L = \boldsymbol{X}_H$ to simplify the formula and computing, and denote $N = N_H = N_L$. This assumption follows the Markov property of the multifidelity random field in Kennedy and O'Hagan's framework, i.e., $\mathrm{Cov}\{Y_H(\boldsymbol{x}), Y_L(\boldsymbol{x}')|Y_L(\boldsymbol{x})\} = 0, \boldsymbol{x} \in \boldsymbol{X}_H, \boldsymbol{x}' \in \boldsymbol{X}_L - \boldsymbol{X}_H$. Finally, we set $\boldsymbol{y}_L$ as the realization from the ensemble $\{u^m(\boldsymbol{x})\}_{m=1}^M$ that maximizes $\ln\widetilde{L}$. The algorithm is summarized in Algorithm 2.1.

It was demonstrated that PhIK prediction on the entire domain D preserves the *linear* physical constraints up to an error bound that relies on the numerical error, the discrepancy between the physical model and real system, and the smallest eigenvalue of matrix $\mathbf{C}$ [49]. For example, deterministic periodic, Dirichlet, or Neumann boundary condition can be preserved. Another type of example is the linear derivative operator, e.g., $\mathcal{L}u = \nabla^2 u$. If u satisfies $\nabla^2 u(\boldsymbol{x};\omega) = 0$ for any $\omega \in \Omega$, e.g., u is the velocity potential, $\hat{y}(\boldsymbol{x})$ from PhIK also guarantees a divergence-free flow field. CoPhIK has the potential to improve the accuracy of the prediction, namely, resulting in a smaller discrepancy between posterior mean and the exact solution because CoPhIK incorporates observations in constructing the GP model, while PhIK only uses model simulations. On the other hand, CoPhIK result may violate some physical constraints because of the choice of Y_d kernel [46].

Algorithm 2.1 CoPhIK using stochastic simulation model $u(\boldsymbol{x};\omega)$ on $D\times\Omega$ ($D \subseteq \mathbb{R}^d$) and high-fidelity observation $\boldsymbol{y}_H = (y_H^{(1)}, \ldots y_H^{(N)})^\top$ at locations $\boldsymbol{X}_H = \{\boldsymbol{x}_H^{(i)}\}_{i=1}^N$.

1: Conduct stochastic simulation, e.g., MC simulation, using $u(\boldsymbol{x};\omega)$ to generate realizations $\{u^m(\boldsymbol{x})\}_{m=1}^M$ on the domain D.
2: Use PhIK to construct GP Y_L on $D \times \Omega$, i.e., $\mu_L(\cdot) = \mu_{\mathrm{MC}}(\cdot)$ and $k_L(\cdot,\cdot) = k_{\mathrm{MC}}(\cdot,\cdot)$ in (2.19). Compute $\mu_L(\boldsymbol{X}_H) = \big(\mu_L(\boldsymbol{x}_H^{(1)}), \ldots \mu_L(\boldsymbol{x}_H^{(N)})\big)^\top$ and $\mathbf{C}_L(\boldsymbol{X}_H, \boldsymbol{X}_H)$ whose ijth element is $k_L(\boldsymbol{x}_H^{(i)}, \boldsymbol{x}_H^{(j)})$. Set $\mathbf{C}_L(\boldsymbol{X}_L, \boldsymbol{X}_H) = \mathbf{C}_L(\boldsymbol{X}_H, \boldsymbol{X}_L) = \mathbf{C}_L(\boldsymbol{X}_H, \boldsymbol{X}_H)$ (because $\boldsymbol{X}_L = \boldsymbol{X}_H$ when identifying $\boldsymbol{y}_L$ in $\widetilde{\boldsymbol{y}}$).
3: Denote $\boldsymbol{y}_d = \boldsymbol{y}_H - \rho\mu_L(\boldsymbol{X}_H)$; choose a specific kernel function $k_d(\cdot,\cdot)$ (Gaussian kernel in this work) for the GP Y_d, and identify hyperparameters via maximizing the log marginal likelihood (2.4), where $\boldsymbol{y}, \boldsymbol{\mu}, \mathbf{C}$ are specified as $\boldsymbol{y}_d, \boldsymbol{\mu}_d, \mathbf{C}_d$, respectively. Then construct $\widetilde{\boldsymbol{\mu}}$ in (2.15) and $\mathbf{C}_d$ whose ijth element is $k_d(\boldsymbol{x}_H^{(i)}, \boldsymbol{x}_H^{(j)})$.
4: Iterate over the set $\{u^m(\boldsymbol{x})\}_{m=1}^M$ to identify $u^m(\boldsymbol{x})$ that maximizes $\ln\widetilde{L}$ in (2.17), where $\boldsymbol{y}_L$ is set as $(u^m(\boldsymbol{x}_H^{(1)}), \ldots, u^m(\boldsymbol{x}_H^{(N)}))^\top$ in $\widetilde{\boldsymbol{y}}$.
5: Compute the posterior mean using (2.13) and variance using (2.14) for any $\boldsymbol{x}^* \in D$.

2.4. Bifidelity approximation.

2.4.1. Algorithm. We briefly describe the bifidelity method [27, 51, 50, 26]. We slightly modify the notation $u(\boldsymbol{x};\omega)$ as $u(\boldsymbol{x};z(\omega))$ to denote the stochastic model used in PhIK and CoPhIK. Here, $z(\omega)$ is the finite-dimensional random variable or field included in the model, and we denote it as z for simplicity. Subsequently, $z^m = z(\omega^m)$ is a sample z, and we assume that $z^m \in I_z$ for any m. Let $u_H^m(\boldsymbol{x})$ denote the high-fidelity simulation result for $u(\boldsymbol{x};z^m)$, e.g., simulation using fine grids or high-order scheme, and $u_L^m(\boldsymbol{x})$ denote the low-fidelity simulation result for $u(\boldsymbol{x};z^m)$, e.g., simulation using coarse grids or low-order scheme. In PhIK, the mean and covariance functions of GP $Y(\boldsymbol{x})$ are approximated using $u^m(\boldsymbol{x})$ (see (2.19)), which are $u_H^m(\boldsymbol{x})$ in the bifidelity framework, and so as GP $Y_L(\boldsymbol{x})$ in the CoPhIK method. Now, we aim to approximate these mean and covariance functions using a few $u_H^m(\boldsymbol{x})$ and a large number of of $u_L^m(\boldsymbol{x})$ to reduce the computational cost of model simulations. More specifically, the bifidelity approach aims to approximate mean and covariance functions in PhIK and CoPhIK functions using $\{u_H^m(\boldsymbol{x})\}_{m=1}^{M_H}$ along with $\{u_L^m(\boldsymbol{x})\}_{m=1}^{M_L}$, where $M_H \ll M_L$, and M_L is M in the original PhIK and CoPhIK formulation.

For brevity, we denote $u_L^m(\boldsymbol{x})$ and $u_H^m(\boldsymbol{x})$ as u_L^m and u_H^m, respectively. We also assume that $u_H^m \in \mathbb{V}_H$ and $u_L^m \in \mathbb{V}_L$, where $\mathbb{V}_H$ and $\mathbb{V}_L$ are Hilbert spaces with inner product $\langle\cdot,\cdot\rangle_H$ and $\langle\cdot,\cdot\rangle_L$, respectively. Given a collection of parametric sample $\Gamma = \{z^1,\ldots,z^{M_L}\} \subset I_z$, we introduce the following notations:

$$
\begin{aligned}
u_L(\Gamma) &= \{u_L^m\}_{m=1}^{M_L}, \quad U_L(\Gamma) = \operatorname{span} u_L(\Gamma) = \operatorname{span}\left\{u_L^1,\ldots,u_L^{M_L}\right\},\\
u_H(\Gamma) &= \{u_H^m\}_{m=1}^{M_L}, \quad U_H(\Gamma) = \operatorname{span} u_H(\Gamma) = \operatorname{span}\left\{u_H^1,\ldots,u_H^{M_L}\right\}.
\end{aligned} \tag{2.23}
$$

The procedure of bifidelity algorithm for estimating mean and covariance functions is presented in Algorithm 2.2.

We detail steps 2 and 4 in Algorithm 2.2 as follows [27, 51]. Let $\mathbf{W}$ be the Gramian matrix of the low-fidelity simulations $u_L(\Gamma)$, i.e.,

$$
w_{ij} = \langle u_L^i, u_L^j\rangle_L, \qquad 1 \le i,j \le M_L. \tag{2.24}
$$

Applying the pivoted Cholesky decomposition to the matrix $\mathbf{W}$ yields

$$
\mathbf{W} = \mathbf{P}^\top \mathbf{L}\mathbf{L}^\top \mathbf{P}, \tag{2.25}
$$

where $\mathbf{L}$ is lower-triangular and $\mathbf{P}$ is a permutation matrix due to pivoting. This will produce an ordered permutation vector $\boldsymbol{p} = (i_1,\ldots,i_{M_L})$, from which we choose

Algorithm 2.2 Bifidelity method computing mean and covariance functions in PhIK and CoPhIK.

1: Conduct the low-fidelity simulations at the sample set Γ to obtain realizations $u_L(\Gamma)$.
2: Select a subset of samples $\gamma = \{z^{i_1},\ldots,z^{i_{M_H}}\} \subset \Gamma$, where $M_H \ll M_L$.
3: Conduct the high-fidelity computations at the subset γ and obtain the high-fidelity simulation samples, $u_H(\gamma) = \left\{u_H^{i_1},\ldots,u_H^{i_{M_H}}\right\}$.
4: Construct $u_B(\Gamma) = \{u_B^1,\ldots,u_B^{M_L}\}$ based on $u_H(\gamma)$.
5: Compute mean and covariance functions using $u_B(\Gamma)$.

the first M_H points to define $\gamma = \{z^{i_1}, \ldots, z^{i_{M_H}}\}$. In practice, we can use a greedy algorithm to identify γ. In each iteration, we find the next sample whose corresponding low-fidelity simulation is furthest from the space spanned by the existing low-fidelity simulation set. Specifically, starting from a trivial initial choice $\gamma_0 = \emptyset$, we let $\gamma_k = \{z_{i_1}, \ldots, z_{i_k}\} \subset \Gamma$ be the k-point existing subset in Γ. We then find the $(k+1)$th point by

$$z^{i_{k+1}} = \operatorname*{argmax}_{z \in \Gamma} \operatorname{dist}(u_L(\boldsymbol{x}; z), U_L(\gamma_k)), \qquad \gamma_{k+1} = \gamma_k \cup z^{i_{k+1}}, \tag{2.26}$$

where $U_L(\gamma_k) = \operatorname{span} u_L(\gamma_k) = \operatorname{span}\left\{u_L^{i_1}, \ldots, u_L^{i_k}\right\}$, the distance function $\operatorname{dist}(g, G)$ between the function $g \in u_L(\Gamma)$ and the space $G \subset U_L(\Gamma)$, follows the standard definition. This greedy algorithm can be readily implemented via simple operations of numerical linear algebra. More details and properties of the algorithm can be found in [27, 51].

Once steps 1–3 are accomplished in Algorithm 2.2, we have sample set Γ, low-fidelity simulations $u_L(\Gamma)$, the subset samples $\gamma \subset \Gamma$, and high-fidelity simulations $u_H(\gamma)$. The next step is a lifting procedure: we use the best approximation rule of u_L we learned on $\mathbb{V}_L$ to construct an interpolation operator, then apply it to $\mathbb{V}_H$. Therefore, $u_L(\gamma) = \left\{u_L^{i_1}, \ldots, u_L^{i_{M_H}}\right\}$ forms a linearly independent set. The convergence of the bifidelity method with respect to M_H is investigated in [27]. In this work, we set the threshold to be 10^{-12} in the greedy algorithm presented in [51] such that span $u_L(\gamma)$ is almost the same as span $u_L(\Gamma)$.

For any $v \in U_L(\Gamma) \subset \mathbb{V}_L$, the projection of v on $\operatorname{span} u_L(\gamma)$ is

$$\mathcal{P}_L v = \sum_{j=1}^{M_H} c_j u_L^{i_j}. \tag{2.27}$$

Then, we define the interpolation operator $\mathcal{I}_L^H(\gamma, v) : \mathbb{V}_L \to U_H(\gamma)$ as

$$\mathcal{I}_L^H(\gamma, v) = \sum_{j=1}^{M_H} c_j u_H^{i_j}, \quad v \in \mathbb{V}_L, \tag{2.28}$$

where $U_H(\gamma) = \operatorname{span} u_H(\gamma)$. Subsequently, we construct $u_B(\Gamma)$ as

$$u_B^m = \mathcal{I}_L^H(\gamma, u_L^m), \quad m = 1, \ldots, M_L. \tag{2.29}$$

The mean and covariance in PhIK are approximated by replacing u^m, where $u^m = u_H^m$, in (2.19) with u_B^m, and set $M = M_L$. The GP Y_L in CoPhIK is constructed in the same manner. We refer to the bifidelity-based PhIK and CoPhIK as *BiPhIK* and *CoBiPhIK*, respectively.

Here, we roughly compare the computational cost of constructing $u_H(\Gamma)$ and $u_B(\Gamma)$ for the sample set Γ of size M. We denote the cost of obtaining one realization of u_H and u_L with C_H and C_L, respectively. Therefore, the total cost of obtaining $u_H(\Gamma)$ is $M_L \mathrm{C}_H$. The computational costs of the pivot Cholesky decomposition and the lifting procedure are negligible when the simulation model is complicated. Thus, the total cost of obtaining $u_B(\Gamma)$ is approximated $M_L \mathrm{C}_L + M_H \mathrm{C}_H$. Therefore, the ratio of computational cost for obtaining $u_B(\Gamma)$ and $u_H(\Gamma)$ is $\dfrac{\mathrm{C}_L}{\mathrm{C}_H} + \dfrac{M_H}{M_L}$. The speedup of bifidelity approximation over high-fidelity simulation on the data set Γ can be significant when $\mathrm{C}_L \ll \mathrm{C}_H$ and $M_H \ll M_L$.

2.4.2. Numerical analysis. Because u_H^m in PhIK and CoPhIK are replaced by u_B^m in BiPhIK and CoBiPhIK, we present numerical analysis results to compare the new method with the original approaches. Recalling that $u^m(\boldsymbol{x})$ in (2.19) are $u_H^m(\boldsymbol{x})$ used in the bifidelity framework and $M = M_L$, we denote $\mu_{\mathrm{MC}}(\boldsymbol{x})$ and $k_{\mathrm{MC}}(\boldsymbol{x}, \boldsymbol{x}')$ in this equation as $\mu_H(\boldsymbol{x})$ and $k_H(\boldsymbol{x}, \boldsymbol{x}')$, respectively. Subsequently, we denote the resulting posterior mean and variance as $\hat{y}_H(\boldsymbol{x})$ and $\hat{s}_H^2(\boldsymbol{x})$. Similarly, when using bifidelity ensemble $u_B(\Gamma)$ to approximate mean and covariance functions in (2.19), i.e., replacing $u^m(\boldsymbol{x})$ with $u_B^m(\boldsymbol{x})$, we denote these two functions as $\mu_B(\boldsymbol{x})$ and $k_B(\boldsymbol{x}, \boldsymbol{x}')$ and the corresponding results as $\hat{y}_B(\boldsymbol{x})$ and $\hat{s}_B^2(\boldsymbol{x})$. We use $\|\cdot\|$ to denote the norm induced from inner product $\langle \cdot, \cdot \rangle_H$ in the Hilbert space $\mathbb{V}_H$ and introduce the following functions:

$$
\begin{aligned}
\sigma_H(\boldsymbol{x}) &= \left(\frac{1}{M-1} \sum_{m=1}^{M} |u_H^m(\boldsymbol{x}) - \mu_H(\boldsymbol{x})|^2 \right)^{\frac{1}{2}}, \\
\sigma_B(\boldsymbol{x}) &= \left(\frac{1}{M-1} \sum_{m=1}^{M} |u_B^m(\boldsymbol{x}) - \mu_B(\boldsymbol{x})|^2 \right)^{\frac{1}{2}}
\end{aligned} \tag{2.30}
$$

and the following constants:

$$
\begin{aligned}
\delta_1 &= \sup_{z \in I_z} \|u_H(\boldsymbol{x}; z) - u_B(\boldsymbol{x}; z)\|, \\
\delta_2 &= \sup_{z \in I_z} \|u_H(\boldsymbol{x}; z) - u_B(\boldsymbol{x}; z)\|_\infty, \\
\sigma_H(\Gamma) &= \left(\frac{1}{M-1} \sum_{m=1}^{M} \|u_H^m(\boldsymbol{x}) - \mu_H(\boldsymbol{x})\|^2 \right)^{\frac{1}{2}}, \\
\sigma_B(\Gamma) &= \left(\frac{1}{M-1} \sum_{m=1}^{M} \|u_B^m(\boldsymbol{x}) - \mu_B(\boldsymbol{x})\|^2 \right)^{\frac{1}{2}}, \\
S_H &= \left(\sum_{n=1}^{N} \sigma_H^2(\boldsymbol{x}^{(n)}) \right)^{\frac{1}{2}}, \quad S_B = \left(\sum_{n=1}^{N} \sigma_B^2(\boldsymbol{x}^{(n)}) \right)^{\frac{1}{2}}, \\
\Delta_H &= \sup_{\boldsymbol{x} \in D} \sigma_H(\boldsymbol{x}), \quad \Delta_B = \sup_{\boldsymbol{x} \in D} \sigma_B(\boldsymbol{x}).
\end{aligned} \tag{2.31}
$$

Here, functions $\sigma_H(\boldsymbol{x})$ and $\sigma_B(\boldsymbol{x})$ depict the standard deviation of $u_H(\Gamma)$ and $u_B(\Gamma)$ at each $\boldsymbol{x}$. Constants δ_1 and δ_2 measure the discrepancy between $u_H(\Gamma)$ and $u_B(\Gamma)$ using different norms. $\sigma_H(\Gamma)$ and $\sigma_B(\Gamma)$ can be understood as "averaged standard deviation" of $u_H(\Gamma)$ and $u_B(\Gamma)$, respectively. S_H and S_B can be viewed as "total standard deviation" of $u_H(\Gamma)$ and $u_B(\Gamma)$ at observation locations. Δ_H and Δ_B are the upper bounds of the standard deviation of $u_H(\Gamma)$ and $u_B(\Gamma)$ on the entire domain. In this study, we assume that $\mathbf{C}_B$ and $\mathbf{C}_H$ are invertible and present the following two theorems that describe the difference between the results of PhIK and BiPhIK.

THEOREM 2.1.

$$
\|\hat{y}_H(\boldsymbol{x}) - \hat{y}_B(\boldsymbol{x})\| \leq C_1 \delta_1 + C_2 \delta_2, \tag{2.32}
$$

where

(2.33)

$$C_1 = 1 + 2S_B\sqrt{\frac{MN}{M-1}}\left\|\mathbf{C}_B^{-1}\right\|_2\|\boldsymbol{y}-\boldsymbol{\mu}_B\|_2,$$

$$C_2 = \sqrt{N}S_H\sigma_H(\Gamma)\left\|\mathbf{C}_H^{-1}\right\|_2\left\{2\sqrt{\frac{2M}{M-1}}\left(S_H^2+S_B^2\right)^{\frac{1}{2}}\left\|\mathbf{C}_H^{-1}\right\|_2\|\boldsymbol{y}-\boldsymbol{\mu}_B\|_2+1\right\}$$

$$+ 2\sqrt{\frac{MN}{M-1}}\sigma_H(\Gamma)\left\|\mathbf{C}_B^{-1}\right\|_2\|\boldsymbol{y}-\boldsymbol{\mu}_B\|_2.$$

THEOREM 2.2.

$$\|\hat{s}_H^2(\boldsymbol{x})-\hat{s}_B^2(\boldsymbol{x})\|_\infty \le C_3\delta_2, \tag{2.34}$$

where

$$C_3 = 2\sqrt{\frac{2M}{M-1}}(\Delta_H^2+\Delta_B^2)^{\frac{1}{2}}\cdot \left\{1+N\left(\Delta_H^2\|\mathbf{C}_H^{-1}\|_2+\sqrt{N}\Delta_H^2\Delta_B^2\|\mathbf{C}_H^{-1}\|_2^2+\Delta_B^2\|\mathbf{C}_B^{-1}\|_2\right)\right\}. \tag{2.35}$$

We present the proof of these two theorems in Appendix B. Here we provide an intuitive interpretation for these theorems. Firstly, it is straightforward that the difference between the results by PhIK and BiPhIK relies on the difference between u_H and u_B. A better bifidelity approximation of the high-fidelity simulations will render smaller difference. Further, constant C_1, C_2, and C_3 depend on the variance of the simulation ensemble such as $S_B, S_H, \sigma_B, \sigma_H$ (which depend on the stochasticity of the physical model) and the discrepancy between the observation and the empirical mean (which is the most probable realization of the constructed GP) of the simulations $\|\boldsymbol{y}-\mu_B\|_2$. Therefore, a stochastic model with smaller variance in the simulations ensemble and good approximation at observation locations will result in a smaller difference. Another important factor is the minimum eigenvalue of $\mathbf{C}_H$ and $\mathbf{C}_B$. We note that (the reciprocal of) these eigenvalues may make the theoretical upper bounds too large in many cases. This is because we use the Cauchy–Schwarz inequality and matrix perturbation results in the proof, and the equalities in these formulae hold only for special cases. Moreover, as we pointed out in section 2.1, in many practical cases, these matrices are not invertible or their condition numbers are large, and a typical practice is to add to them a regularity term $\delta^2\mathbf{I}(\delta>0)$. Then, the L_2 norm of the inverse of these matrices can be bounded by δ^{-2}, and so as in the cases with noisy observations, e.g., adding $\delta^2\mathbf{I}$ to the covariance matrix to represent the i.i.d. Gaussian noise in observations. The constant $\sqrt{\frac{M}{M-1}}$ is smaller than 2, and it is approximately equal to 1 as we usually choose $M\sim\mathcal{O}(100)$. In addition, these two theorems estimate the difference between the posterior mean and variance by PhIK and BiPhIK instead of comparing the reconstruction accuracy of PhIK and BiPhIK.

We also note that Theorem 2.2 uses the L_∞ norm because the greedy algorithm we use to add new observations is based on the maximum of $\hat{s}$. The error estimate in L_2 norm can also be derived using the similar procedure in the proofs of Theorems 2.1 and 2.2. Moreover, although we present upper bounds in terms of δ_1 and δ_2 (which are adapted from [27]), the analysis is dependent on the mean and covariance functions constructed by different methods. It is possible that in some cases, the pathwise difference is large in different methods (i.e., δ_1 and δ_2 are large), but the

difference between the mean and covariance functions is small. Quantitative analysis for CoBiPhIK also depends on the property of kernel function for Y_d and the convexity of the optimization, and it is not available at this time. Empirically, CoPhIK is more sensitive to the difference between $u_H(\Gamma)$ and $u_B(\Gamma)$.

The next two theorems describe how well a linear physical constraint is preserved in BiPhIK and CoBiPhIK posterior mean.

THEOREM 2.3. *Assume that* $\|\mathcal{L}u_H(\boldsymbol{x}; z(\omega^m)) - g(\boldsymbol{x})\| \leq \epsilon$ *for any* $\omega^m \in \Omega$, *where* $\mathcal{L}$ *is a deterministic bounded linear operator,* $g(\boldsymbol{x})$ *is a well-defined deterministic function on* $\mathbb{R}^d$, *and* $\|\cdot\|$ *is the norm in* $\mathbb{V}_H$. *Then, the posterior mean* $\hat{y}_B(\boldsymbol{x})$ *from BiPhIK satisfies*

$$\begin{aligned}\|\mathcal{L}\hat{y}_B(\boldsymbol{x}) - g(\boldsymbol{x})\| \leq & \epsilon \left\{ 1 + 2S_H \sqrt{\frac{M}{M-1}} \left\|\mathbf{C}_H^{-1}\right\|_2 \|\boldsymbol{y} - \boldsymbol{\mu}_H\|_2 \right\} \\ & + M_{\mathcal{L}}(C_1\delta_1 + C_2\delta_2),\end{aligned} \tag{2.36}$$

where $M_{\mathcal{L}}$ *is the bound of* $\mathcal{L}$ *and* C_1 *and* C_2 *are defined in Theorem* 2.1.

Proof.

$$\begin{aligned}\|\mathcal{L}\hat{y}_B(\boldsymbol{x}) - g(\boldsymbol{x})\| &= \|\mathcal{L}[\hat{y}_B(\boldsymbol{x}) - \hat{y}_H(\boldsymbol{x}) + \hat{y}_H(\boldsymbol{x})] - g(\boldsymbol{x})\| \\ &\leq \|\mathcal{L}\hat{y}_H(\boldsymbol{x}) - g(\boldsymbol{x})\| + \|\mathcal{L}(\hat{y}_B(\boldsymbol{x}) - \hat{y}_H(\boldsymbol{x}))\| \\ &= \|\mathcal{L}\hat{y}_H(\boldsymbol{x}) - g(\boldsymbol{x})\| + M_{\mathcal{L}}\|\hat{y}_B(\boldsymbol{x}) - \hat{y}_H(\boldsymbol{x})\|.\end{aligned}$$

Theorem 2.1 presents the upper bound of $\|\hat{y}_B(\boldsymbol{x}) - \hat{y}_H(\boldsymbol{x})\|$. The upper bound for $\|\mathcal{L}\hat{y}_H(\boldsymbol{x}) - g(\boldsymbol{x})\|$ needs slightly modifying the proof of Theorem 2.1 in [49]. Specifically, by setting $g(\boldsymbol{x}; \omega) = g(\boldsymbol{x})$ the last line in that proof will be

$$\|\mathcal{L}\hat{y}_H(\boldsymbol{x}) - g(\boldsymbol{x})\| \leq \epsilon + 2\epsilon\sqrt{\frac{M}{M-1}} \sum_{n=1}^{N} |a_n| \sigma_H(\boldsymbol{x}^{(n)}), \tag{2.37}$$

where

$$\begin{aligned}\sum_{n=1}^{N} |a_n| \sigma_H(\boldsymbol{x}^{(n)}) &\leq \left(\sum_{n=1}^{N} a_n^2\right)^{\frac{1}{2}} \left(\sum_{n=1}^{N} \sigma_H^2(\boldsymbol{x}^{(n)})\right)^{\frac{1}{2}} \\ &= \|\mathbf{C}_H^{-1}(\boldsymbol{y} - \boldsymbol{\mu}_H)\|_2 S_H \leq \|\mathbf{C}_H^{-1}\|_2 \|\boldsymbol{y} - \boldsymbol{\mu}_H\|_2 S_H.\end{aligned}$$

∎

THEOREM 2.4. *Assume that* $\|\mathcal{L}u_H(\boldsymbol{x}; z(\omega^m)) - g(\boldsymbol{x})\| \leq \epsilon$ *for any* $\omega^m \in \Omega$, *where* $\mathcal{L}$ *is a deterministic bounded linear operator,* $g(\boldsymbol{x})$ *is a well-defined deterministic function on* $\mathbb{R}^d$, *and* $\|\cdot\|$ *is the norm in* $\mathbb{V}_H$. *Then, the posterior mean* $\hat{y}_B(\boldsymbol{x})$ *from CoBiPhIK, in which* Y_L *is constructed by* $u_B(\Gamma)$, *satisfies*

$$\begin{aligned}\|\mathcal{L}\hat{y}_B(\boldsymbol{x}) - g(\boldsymbol{x})\| \leq & \rho\epsilon + (1-\rho)\|g(\boldsymbol{x})\| + 2\epsilon\rho S_H \sqrt{\frac{M}{M-1}} \|\mathbf{C}_B^{-1}\|_2 \|\boldsymbol{y}_L - \boldsymbol{\mu}_B\|_2 \\ & + \|\mathcal{L}\mu_d\| + \|\mathbf{C}_2^{-1}\|_2 \|\boldsymbol{y}_H - \rho\boldsymbol{y}_B - \mathbf{1}\mu_d\|_2 \sum_{n=1}^{N} \left\| \mathcal{L}k_d(\boldsymbol{x}, \boldsymbol{x}^{(n)}) \right\| \\ & + M_{\mathcal{L}}(C_1\delta_1 + C_2\delta_2),\end{aligned} \tag{2.38}$$

where $\mathbf{C}_1$ *and* $\mathbf{C}_2$ *are* $\mathbf{C}_L(\boldsymbol{X}_L, \boldsymbol{X}_L)$ *and* $\mathbf{C}_d(\boldsymbol{X}_H, \boldsymbol{X}_H)$ *in Algorithm* 2.1, *respectively, and* $M_{\mathcal{L}}$ *is the bound of* $\mathcal{L}$.

Proof.

$$\|\mathcal{L}\hat{y}_B(\boldsymbol{x}) - g(\boldsymbol{x})\| \le \|\mathcal{L}\hat{y}_H(\boldsymbol{x}) - g(\boldsymbol{x})\| + M_{\mathcal{L}}\|\hat{y}_B(\boldsymbol{x}) - \hat{y}_H(\boldsymbol{x})\|,$$

where $\hat{y}_H(\boldsymbol{x})$ is the posterior mean by the original CoPhIK method, i.e., $\{u^m(\boldsymbol{x})\}_{m=1}^M$ is taken as $u_H(\Gamma)$ in Algorithm 2.1. Then similar to the proof in Theorem 2.4, we set $g(\boldsymbol{x};\omega) = g(\boldsymbol{x})$ and use the Cauchy–Schwarz inequality to slightly modify the upper bound estimate of $\sum_{n=1}^N |a_n|\sigma_H(\boldsymbol{x}^{(n)})$ in Theorem 2.2 in [46] to finish the proof. ∎

Remark. In Theorems 2.3 and 2.4 we use constants defined in (2.31) and Theorem 2.1 to keep the notations consistent. An alternative form of the bound (not presented in this work) can be derived based on the interpolation operator $\mathcal{I}_L^H(\gamma, v)$ in (2.28) following the proof of Theorem 2.1 in [49]. Moreover, in general, even if $\mathcal{L}u_H(\boldsymbol{x}; z(\omega^m)) = g(\boldsymbol{x})$, there is no guarantee $\mathcal{L}u_B(\boldsymbol{x}; z(\omega^m)) = g(\boldsymbol{x})$, because the summation of c_j in $\mathcal{I}_L^H(\gamma, v)$ may not be equal to 1. A special case is $\mathcal{L}u_H(\boldsymbol{x}; z(\omega^m)) = 0$ for any $\omega^m \in \Omega$. In this scenario, $\mathcal{L}u_B(\boldsymbol{x}; z(\omega^m)) = 0$ for any $\omega^m \in \Omega$; thus $\mathcal{L}\hat{y}_B(\boldsymbol{x}) = 0$ because of the linearity of $\mathcal{L}$.

In summary, the upper bounds provided in this section rely on the variance of the physical model simulations, the difference between observations and the GP mean, and the minimum eigenvalue of covariance matrices (which can be bounded by the magnitude of the additive regularity matrix for small condition number case or by the magnitude of the observation noise in practical problems; see, e.g., [19]). Although in many cases these bounds are conservative and sharper bounds require further investigation, they still provide an intuition of understanding the proposed algorithms. Moreover, these analysis results provide a framework of analyzing physics-informed GP with mean and covariance generated by different methods (e.g., the methods we mentioned in the introduction), and the conclusions in this section can be easily adapted to those cases.

3. Numerical examples. We present three numerical examples to demonstrate the performance of the proposed methods. The first two examples are drawn from the previous examples in [46] to compare different methods, and they are two-dimensional problems in physical space. We denote reference solution, a discretized two-dimensional field, as matrix $\mathbf{F}$, the reconstructed field (posterior mean) as $\mathbf{F}_r$. We present the RMSE $\hat{s}$, difference $\mathbf{F}_r - \mathbf{F}$, and relative error $\|\mathbf{F}_r - \mathbf{F}\|_F / \|\mathbf{F}\|_F$ ($\|\cdot\|_F$ is the Frobenius norm) to compare different methods. Moreover, we adaptively add new observations at the maxima of $\hat{s}$ (see Appendix A) to numerically study the convergence with respect to the number of observations. In the first two numerical examples, we use Gaussian kernel in kriging, CoPhIK, and CoBiPhIK because it is a special case in the widely used Matérn kernel family suitable for the relatively smooth field. In the third numerical example, we use a periodic kernel in aligned with the periodic boundary condition in the problem. All simulations were performed by MATLAB 2016b on a Mac desktop with 4GHz Intel i7 CPU.

3.1. Branin function. We consider the following modified Branin function [7]:

$$f(\boldsymbol{x}) = a(\bar{y} - b\bar{x}^2 + c\bar{x} - r)^2 + g(1-p)\cos(\bar{x}) + g + qx, \tag{3.1}$$

where $\boldsymbol{x} = (x, y)$,

$$\bar{x} = 15x - 5,\ \bar{y} = 15y,\ (x, y) \in D = [0, 1] \times [0, 1],$$

and

$$a = 1,\ b = \frac{5.1}{4\pi^2},\ c = \frac{5}{\pi},\ r = 6,\ g = 10,\ p = \frac{1}{8\pi},\ q = 5.$$

The contour of f and eight randomly chosen observation locations $\{(0.1, 0.225), (0.475, 0.2), (0.625, 0.5), (0.675, 0.55), (0.7, 0), (0.775, 0.1), (0.8, 0.9), (0.925, 0.9)\}$ are presented in Figure 3.1. The function f is evaluated on a 41×41 uniform grid.

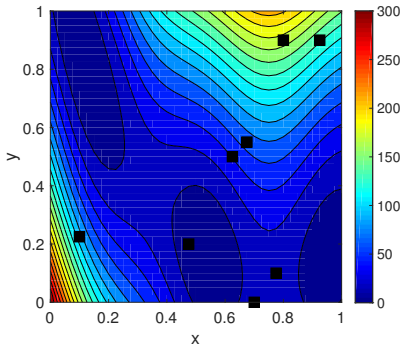


FIG. 3.1. *Contours of modified Branin function (on* 41×41 *uniform grids) and locations of eight observations (black squares).*

We assume that based on "domain knowledge," $f(\boldsymbol{x})$ is partially known. Specifically, its form is known, but the coefficients b and q are unknown. Then, we treat these coefficients as random fields $\hat{b}$ and $\hat{q}$, and we also modify the second g as $\hat{g}$, which indicates that the field f is described by a random function $\hat{f} : D \times \Omega \to \mathbb{R}$:

$$\hat{f}(\boldsymbol{x};\omega) = a(\bar{y} - \hat{b}(\boldsymbol{x};\omega)\bar{x}^2 + c\bar{x} - r)^2 + g(1-p)\cos(\bar{x}) + \hat{g} + \hat{q}(\boldsymbol{x};\omega)x, \tag{3.2}$$

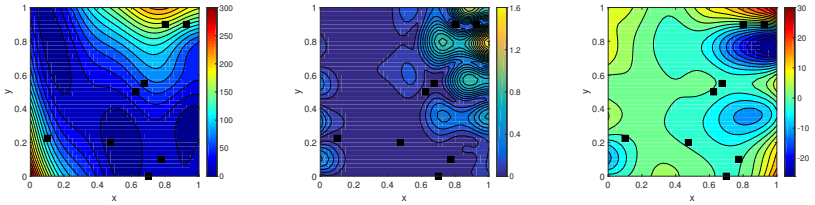
where $\hat{g} = 20$,

$$\hat{b}(\boldsymbol{x};\omega) = b\Bigg\{0.9 + \frac{0.2}{\pi}\sum_{i=1}^{3}\Bigg[\frac{1}{4i-1}\sin((2i-0.5)\pi x)\xi_{2i-1}(\omega) + \frac{1}{4i+1}\sin((2i+0.5)\pi y)\xi_{2i}(\omega)\Bigg]\Bigg\}, \tag{3.3}$$

$$\hat{q}(\boldsymbol{x};\omega) = q\Bigg\{1.0 + \frac{0.6}{\pi}\sum_{i=1}^{3}\Bigg[\frac{1}{4i-3}\cos((2i-1.5)\pi x)\xi_{2i+5}(\omega) + \frac{1}{4i-1}\cos((2i-0.5)\pi y)\xi_{2i+6}(\omega)\Bigg]\Bigg\}, \tag{3.4}$$

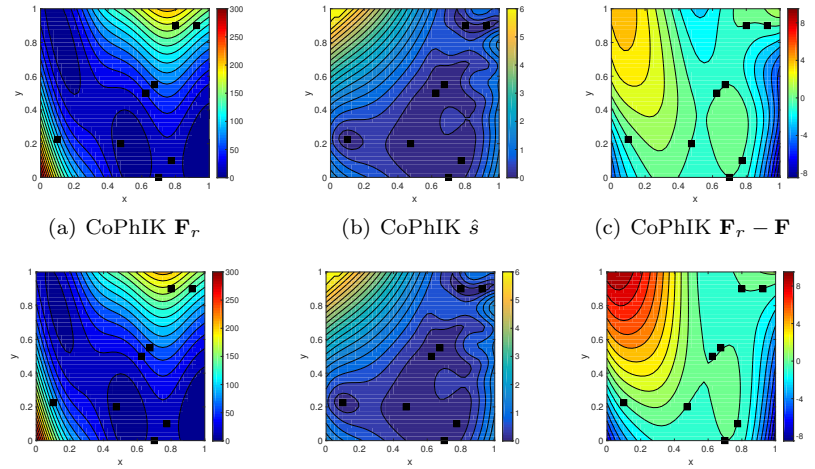
and $\{\xi_i(\omega)\}_{i=1}^{12}$ are i.i.d. Gaussian random variables with zero mean and unit variance. We use this "physical knowledge" to compute the mean and covariance function of $\hat{f}$ by generating $M = 300$ samples of $\xi_i(\omega)$ and evaluating $\hat{f}$ on the 21×21 uniform grid for each sample of $\xi_i(\omega)$ to obtain realization ensemble $u_L(\Gamma)$. M_H is chosen to be 21 to construct γ and subsequently evaluate $\hat{f}$ on 41×41 uniform grid to obtain $u_H(\gamma)$. Finally, we construct $u_B(\Gamma)$ based on $u_H(\gamma)$ and $u_L(\Gamma)$ (Algorithm 2.2) and use it in BiPhIK and CoBiPhIK.

It is shown in [49, 46] that kriging results in inaccurate reconstruction of $\mathbf{F}$ by using the eight observation data. The results of PhIK and BiPhIK are very similar in this case, and we present the latter in Figure 3.2. These results are much better than the kriging (see also the quantitative comparison in Figure 3.6). There are slight difference between the results by CoPhIK and CoBiPhIK as shown in Figure 3.3.



(a) BiPhIK $\mathbf{F}_r$ (b) BiPhIK $\hat{s}$ (c) BiPhIK $\mathbf{F}_r - \mathbf{F}$

FIG. 3.2. *Reconstruction of the modified Branin function by BiPhIK with eight original observations (squares).*



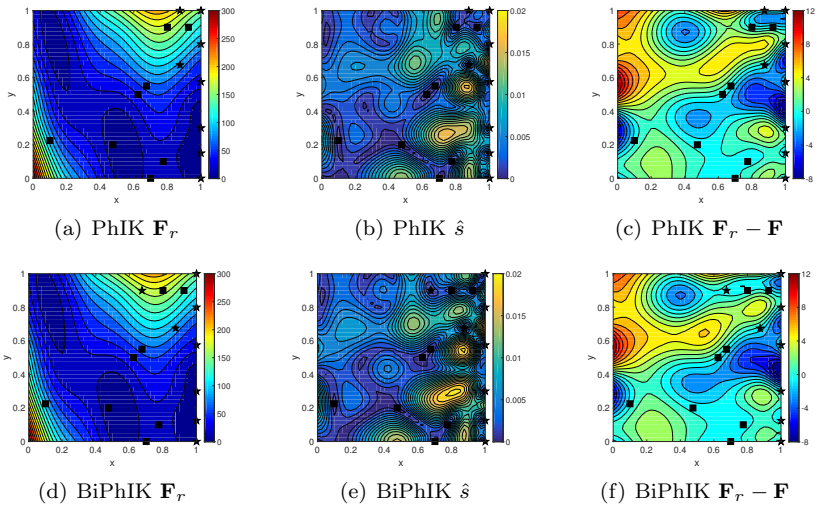
(a) CoPhIK $\mathbf{F}_r$ (b) CoPhIK $\hat{s}$ (c) CoPhIK $\mathbf{F}_r - \mathbf{F}$

(d) CoBiPhIK $\mathbf{F}_r$ (e) CoBiPhIK $\hat{s}$ (f) CoBiPhIK $\mathbf{F}_r - \mathbf{F}$

FIG. 3.3. *Reconstruction of the modified Branin function by CoPhIK and Co-BiPhIK with eight original observations (squares).*

We then use a greedy algorithm (in the appendix) that acquires additional observations of the exact field one by one. Figure 3.4 presents the comparison of PhIK and BiPhIK when eight additional observations (marked as black stars) are added, which shows a slight difference in the location of additional observations and small discrepancy between the results. Also, Figure 3.5 illustrates the difference between CoPhIK and CoBiPhIK, and we see more significant differences in the pattern of $\hat{s}$ and $\boldsymbol{F}_r - \boldsymbol{F}$ than in the comparison between PhIK and BiPhIK.

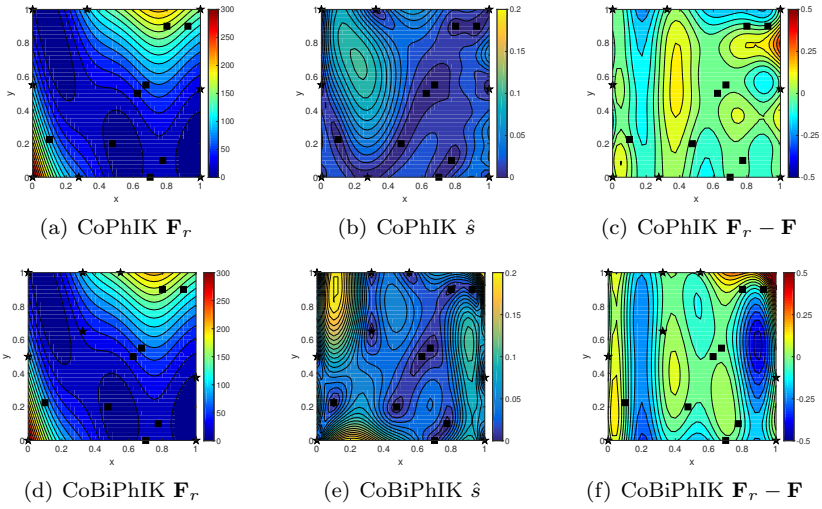
Figure 3.6 presents a quantitative study of the difference between the posterior mean and the reference solution with respect to the total number of observation data. The results are consistent with Figures 3.2–3.5 in that the difference between PhIK and BiPhIK is very small while the difference between CoPhIK and CoBiPhIK is larger. We note that the latter is still very small ranging from $\mathcal{O}(10^{-3})$ to $\mathcal{O}(10^{-2})$ depending on the number of observations. This is because $u_B(\Gamma)$ approximates $u_H(\Gamma)$ very well in this case. Specifically, $\delta_1 = 0.0279$ and $\delta_2 = 0.0012$, and the actual averaged difference between bifidelity reconstruction (denoted as $\hat{\mathbf{F}}_B$) and high-fidelity reconstruction (denoted as $\hat{\mathbf{F}}_H$) is $\frac{\|\hat{\mathbf{F}}_B - \hat{\mathbf{F}}_H\|_F}{\sqrt{1681}} = 0.0257$.



(a) PhIK $\mathbf{F}_r$ (b) PhIK $\hat{s}$ (c) PhIK $\mathbf{F}_r - \mathbf{F}$

(d) BiPhIK $\mathbf{F}_r$ (e) BiPhIK $\hat{s}$ (f) BiPhIK $\mathbf{F}_r - \mathbf{F}$

FIG. 3.4. *Reconstruction of the modified Branin function by PhIK and BiPhIK with eight original observations (squares) and eight additional observations (stars).*



(a) CoPhIK $\mathbf{F}_r$ (b) CoPhIK $\hat{s}$ (c) CoPhIK $\mathbf{F}_r - \mathbf{F}$

(d) CoBiPhIK $\mathbf{F}_r$ (e) CoBiPhIK $\hat{s}$ (f) CoBiPhIK $\mathbf{F}_r - \mathbf{F}$

FIG. 3.5. *Reconstruction of the modified Branin function by CoPhIK and CoBiPhIK with eight original observations (squares) and eight additional observations (stars).*

We note that the errors by PhIK/BiPhIK do not decrease from 12 observations to 24 observations. This is the drawback of this method as discussed in [49]. The reason is that the GP in PhIK is only dependent on physical model simulations, and the observations are used to project the exact solution to the space spanned by these simulations. CoPhIK/CoBiPhIK use the observations to modify the GP structure; hence they have the potential to obtain better approximation. Moreover, the comparison of CoPhIK and CoBiPhIK is not consistent in terms of the relative error. This is because we use the greedy algorithm to select additional observation locations, and these two methods provide different selections (see Figure 3.5).

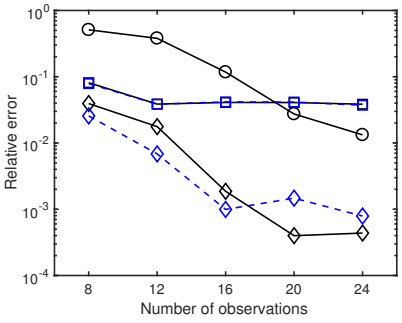


FIG. 3.6. *Relative error of reconstructed modified Branin function* $\|\boldsymbol{F}_r - \boldsymbol{F}\|_F/\|\boldsymbol{F}\|_F$ *using kriging ("∘"), PhIK (blue "□"), BiPhIK (black "□"), CoPhIK (blue "⋄"), and CoBiPhIK (black "⋄") with different numbers of total observations via active learning.*

3.2. Heat transfer. In the second example, we consider the steady state of a heat transfer problem. The nondimesionalized heat equation is given as

$$\frac{\partial T}{\partial t} - \nabla \cdot (\kappa(T)\nabla T) = 0, \quad \boldsymbol{x} \in D, \tag{3.5}$$

where $T(\boldsymbol{x}, t)$ is the temperature, and the heat conductivity κ is set as a function of T. The computational domain D is a rectangle $[-0.5, 0.5] \times [-0.2, 0.2]$ with two circular cavities $R_1(O_1, r_1)$ and $R_2(O_2, r_2)$, where $O_1 = (-0.3, 0), O_2 = (0.2, 0), r_1 = 0.1, r_2 = 0.15$ (see Figure 3.7). The boundary conditions are given as follows:

$$\begin{cases} T = -30\cos(2\pi x) + 40, & x \in \Gamma_1; \\ \kappa(T)\dfrac{\partial T}{\partial \boldsymbol{n}} = -20, & x \in \Gamma_2; \\ T = 30\cos(2\pi(x + 0.1)) + 40, & x \in \Gamma_3; \\ \kappa(T)\dfrac{\partial T}{\partial \boldsymbol{n}} = 20, & x \in \Gamma_4; \\ \kappa(T)\dfrac{\partial T}{\partial \boldsymbol{n}} = 0, & x \in \Gamma_5. \end{cases} \tag{3.6}$$

The "real" conductivity is set as

$$\kappa(T) = 1.0 + \exp(0.02T), \tag{3.7}$$

and the profile of the steady state temperature is presented in Figure 3.7. This solution is obtained by finite element method with unstructured triangular mesh using the MATLAB PDE toolbox, and the degree of freedom (DOF) is 1319 (maximum grid size is 0.02). The observations of this exact profile (denoted as $\mathbf{F}$) are collected at six locations $\{(-0.4, \pm 0.1), (-0.05, \pm 0.1), (0.4, \pm 0.1)\}$ (black squares in Figure 3.7).

Now we assume that due to the lack of knowledge, the conductivity is modeled as

$$\kappa(T; \omega) = 0.1 + \xi T, \tag{3.8}$$

where $\xi(\omega)$ is a uniform random variable $\mathcal{U}[0.0012, 0.0108]$. Apparently, this physical model significantly *underestimates* the heat conductivity, and its form is incorrect. We generate $M = 400$ samples of $\xi(\omega)$ and solve (3.5) on coarser grid (maximum

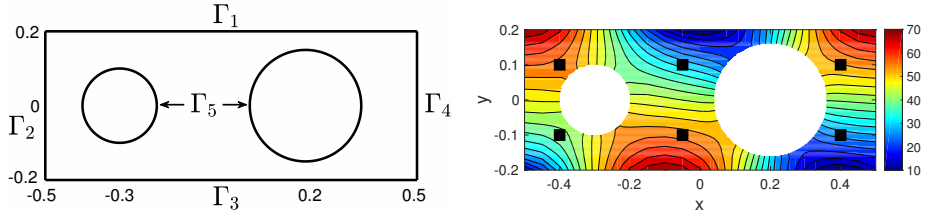
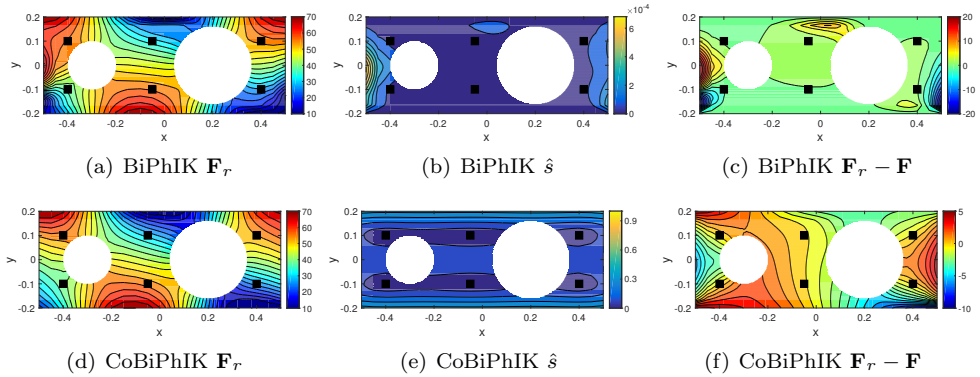


FIG. 3.7. *Heat transfer problem. Left: computational domain; right: contours of steady state solution and locations of six observations (black squares).*



(a) BiPhIK $\mathbf{F}_r$ (b) BiPhIK $\hat{s}$ (c) BiPhIK $\mathbf{F}_r - \mathbf{F}$

(d) CoBiPhIK $\mathbf{F}_r$ (e) CoBiPhIK $\hat{s}$ (f) CoBiPhIK $\mathbf{F}_r - \mathbf{F}$

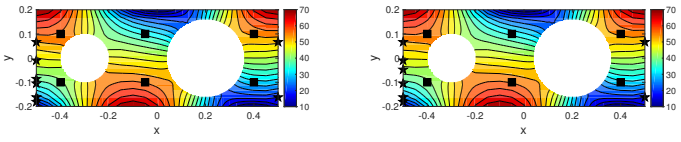
FIG. 3.8. *Reconstruction of the steady state solution of heat transfer problem by BiPhIK (first row) and CoBiPhIK (second row) with eight original observations (squares).*

grid size is 0.1) with DOF = 96 to obtain corresponding temperature solutions which form $u_L(\Gamma)$. M_H is chosen to be 19 in this example. The walltime of generating samples for bifidelity method (400 low-resolution simulations and 19 high-resolution simulations) is 167.5s. The walltime of generating 400 high-resolution simulations is 434.5s. Therefore, the computational cost saving is roughly 2.5 times.

It is shown in [46] that the kriging reconstruction is not accurate because of the selection of observations and the property of the exact solution. Figure 3.8 presents the results by BiPhIK and CoBiPhIK, and they are very similar to the results by PhIK and CoPhIK in [46] (not shown here), respectively. These results are better than kriging (see Figure 3.11 for quantitative comparison).

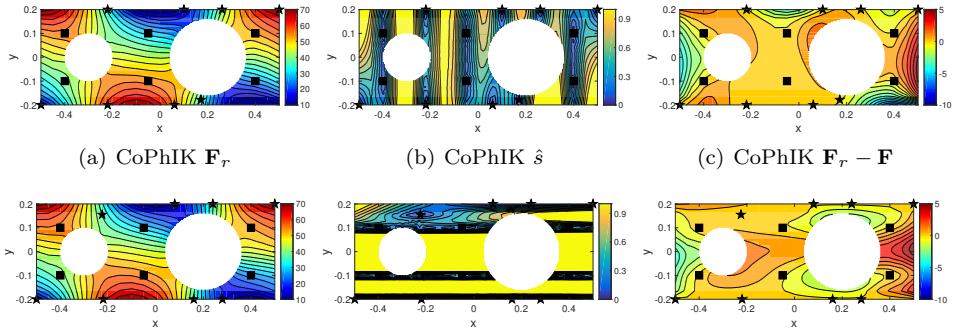
Next, we adaptively add more observation data one by one. The results by PhIK and BiPhIK are very similar. We only present $\mathbf{F}_r$ by these two methods in Figure 3.9 for comparison. We can see that the discrepancy in $\mathbf{F}_r$ is very insignificant, and there is only a slight difference in the locations of new observations (marked as stars) on the boundary Γ_2. Figure 3.10 compares the results by CoPhIK and CoBiPhIK. Although $\mathbf{F}_r$ and $\mathbf{F}_r - \mathbf{F}$ are similar, there is significant discrepancy in $\hat{s}$, mainly because the correlation length l_i in the Y_d's kernel function is different.

Figure 3.11 presents the quantitative comparison of the relative error by different methods. In this case, the PhIK and BiPhIK results are almost the same, and the CoPhIK and CoBiPhIK results are also very similar. Kriging performs poorly when the number of observations is smaller than 22; however, it outperforms PhIK (and BiPhIK) when 22 observations are available. CoPhIK (and CoBiPhIK) is always better than kriging and PhIK (and BiPhIK) as shown in [46]. Again, in this case,



(a) PhIK $\mathbf{F}_r$ (b) BiPhIK $\mathbf{F}_r$

FIG. 3.9. *Reconstruction of the steady state solution of heat transfer problem by PhIK and BiPhIK with eight original observations (squares) and eight additional observations (stars).*



(a) CoPhIK $\mathbf{F}_r$ (b) CoPhIK $\hat{s}$ (c) CoPhIK $\mathbf{F}_r - \mathbf{F}$

(d) CoBiPhIK $\mathbf{F}_r$ (e) CoBiPhIK $\hat{s}$ (f) CoBiPhIK $\mathbf{F}_r - \mathbf{F}$

FIG. 3.10. *Reconstruction of the steady state solution of heat transfer problem by CoPhIK and CoBiPhIK with eight original observations (squares) and eight additional observations (stars).*

$u_B(\Gamma)$ approximates $u_H(\Gamma)$ very well ($\delta_1 = 0.0039$ and $\delta_2 = 0.0015$), which yields small difference between PhIK and BiPhIK (the actual averaged difference between bifidelity reconstruction $\hat{\mathbf{F}}_B$ and high-fidelity reconstruction $\hat{\mathbf{F}}_H$ is $\frac{\|\hat{\mathbf{F}}_B - \hat{\mathbf{F}}_H\|_F}{\sqrt{1396}} = 0.181$) and between CoPhIK and CoBiPhIK. The high-fidelity simulations $u_H(\Gamma)$ satisfy the Dirichlet boundary condition on Γ_1 and Γ_3, and so do the results by PhIK. BiPhIK slightly violates this boundary condition; the L_2 error is about 0.014 on both Γ_1 and Γ_2. The L_2 error for CoBiPhIK is 17.7 on Γ_1 and 33.8 on Γ_2 because the covariance function of Y_d does not satisfy the boundary condition.

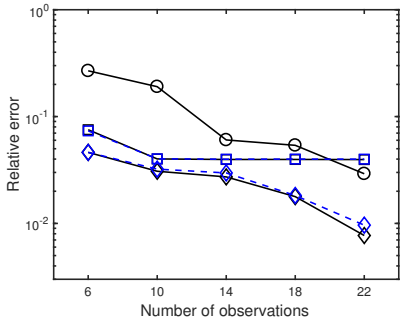


FIG. 3.11. *Relative error of reconstructed steady state solution of heat transfer problem $\|\boldsymbol{F}_r - \boldsymbol{F}\|_F / \|\boldsymbol{F}\|_F$ using kriging ("○"), PhIK (blue "□"), BiPhIK (black "□"), CoPhIK (blue "◇"), and CoBiPhIK (black "◇") with different numbers of total observations via active learning.*

3.3. Kuramoto–Sivashinsky equation. We consider the one-dimensional Kuramoto-Sivashinsy (KS) equation [20, 38]:

$$\begin{aligned} &u_t + 4u_{xxxx} + \alpha\left[u_{xx} + \frac{1}{2}(u_x)^2\right] = 0, \quad 0 \le x \le 2\pi, \\ &u(x+2\pi, t) = u(x,t), \\ &u(x,0) = u_0(x), \end{aligned} \tag{3.9}$$

where

$$u_0(x) = 2.9420\cos(2x) + 0.4642\cos(4x) + 0.0410\cos(6x) + 0.0034\cos(8x). \tag{3.10}$$

It is well known that this equation can be used to depict a chaotic system, and it is very sensitive to the parameter α when it is large. More importantly, in numerical simulation, high precision is necessary because of the extreme sensitivity of the simulations with respect to numerical accuracy [11, 12]. We use the spectral method for spatial derivatives as in [23]. Specifically, we use Fourier expansion with 256 terms to obtain the reference solution and u_H and use expansion with 128 terms to compute u_L. For the time integration, we use the fourth-order Runge–Kutta method with time step 10^{-3}. We investigate the solution of KS equation at $T = 5$ and the "exact" $\alpha = 37.545$. Accurate observations are available at

$$x = \frac{12\pi}{256} + \frac{56\pi}{256} \cdot j, \quad j = 0, 1, \ldots, 8.$$

We assume that we don't know the exact α and use the biased "domain knowledge" to set α as a uniform random variable $\mathcal{U}[30, 36]$. Apparently, this range is below the exact α. We generate 400 samples of α and compute corresponding u_H^m and u_L^m to compare the performance of different methods. Specifically, the M_H is chosen to be 17 to construct $u_B(\Gamma)$. The walltime of generating simulations in bifidelity method (400 low-resolution simulations and 17 high-resolution simulations) is $69.6 + 5.9 = 75.5$s. The walltime of generating 400 high-resolution simulation is 136.8s. The speedup is roughly 1.8 times. For the kriging, we use the following periodic kernel [25]:

$$k(x, x') = \sigma^2 \exp\left(-\frac{2\sin^2(\frac{x-x'}{2})}{l^2}\right).$$

We also use this kernel for Y_d in CoPhIK and CoBiPhIK.

Figure 3.12 illustrates the exact solution and the locations of accurate observations. The mean of the $\{u_H^m\}_{m=1}^{400}$ is illustrated as the dashed line, which deviates from the exact solution significantly with the relative L_2 error more than 140%. We also compute the standard deviation of u_H at each x, and because we will use statistics of u_H to construct a GP in PhIK, we present the "95% confidence interval," i.e., mean plus and minus two standard deviations, in Figure 3.12. We note that this is not the exact confidence interval of the ensemble $\{u_H^m\}_{m=1}^{400}$ itself. Figure 3.12 shows clearly that the exact solution is not bounded by the confidence interval. This is because in the stochastic model, the α is below its exact value, and the KS equation is very sensitive to α.

Figure 3.13 illustrates the kriging results by using the nine accurate observations. The uncertainty is larger (i.e., the confidence interval is wider) near the right end

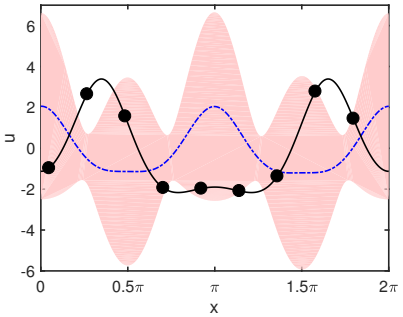


FIG. 3.12. *Exact solution of the KS equation (solid line), observations (circles), mean of the realizations (dashed line), and the "95% confidence interval" (shaded area).*

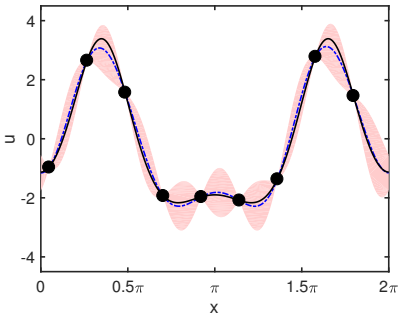
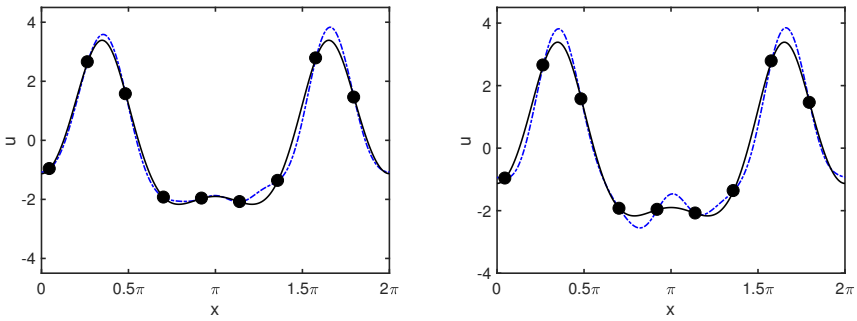


FIG. 3.13. *Kriging reconstruction of KS equation: posterior mean (dashed line), exact solution (solid line), observations (circles), and 95% confidence interval (shaded area).*

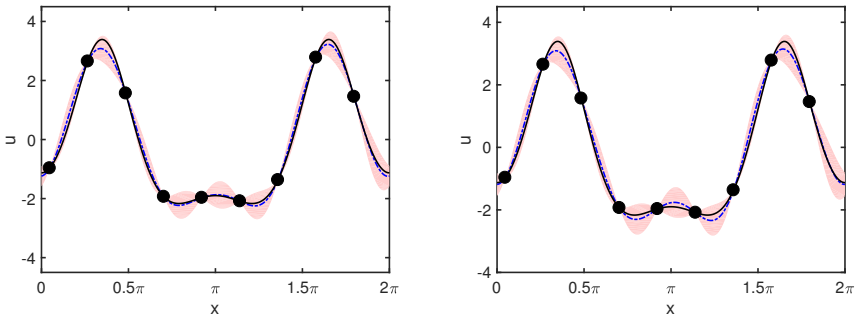
even though we use a periodic kernel function. This is because there is no observation near the right end compared with other regions. Figure 3.14 shows the results by PhIK (using $\{u_H^m\}_{m=1}^{400}$) and BiPhIK (using $\{u_B^m\}_{m=1}^{400}$). The PhIK performs better than BiPhIK. We note that in this figure, the confidence intervals in both methods are very narrow ($\sim \mathcal{O}(10^{-2})$). Similar to the examples in [49, 46] and the other two examples in this session, PhIK usually yields a less uncertain result, but this estimate of the uncertainty may not be very accurate because it relies on its prior covariance, which is totally dependent on the stochasticity of the physical model. Figure 3.15 demonstrates the results by CoPhIK and CoBiPhIK. The CoPhIK is the most accurate of all the methods (smallest discrepancy between posterior mean and the exact accuracy) with confidence intervals that cover the exact solution well. The CoBiPhIK result is closer to that of CoPhIK in both posterior mean and confidence interval than the difference between PhIK and BiPhIK.

Table 3.1 presents the relative L_2 errors of all methods. Different from the previous two examples, in this example, the structures of space $U_L(\Gamma)$ and $U_H(\Gamma)$ are different because of the sensitivity of the system to the numerical solution, so $u_B(\Gamma)$ can't approximate $u_H(\Gamma)$ well as in the previous examples. Specifically, in this case $\delta_1 = 89.8$ and $\delta_2 = 7.7$. Consequently, the difference between PhIK and BiPhIK is larger than the same comparison in the other two examples (the actual averaged difference between bifidelity reconstruction $\hat{\boldsymbol{y}}_B$ and high-fidelity reconstruction $\hat{\boldsymbol{y}}_H$ is $\frac{\|\hat{\boldsymbol{y}}_B - \hat{\boldsymbol{y}}_H\|_F}{\sqrt{256}} = 0.193$), and the error of CoBiPhIK is larger than CoPhIK. Moreover,



(a) PhIK (b) BiPhIK

FIG. 3.14. *PhIK (left) and BiPhIK (right) reconstruction of the KS equation: posterior mean (dashed line), exact solution (solid line), observations (circles), and 95% confidence interval (very narrow in this case).*



(a) CoPhIK (b) CoBiPhIK

FIG. 3.15. *CoPhIK (left) and BiCoPhIK (right) reconstruction of the KS equation: posterior mean (dashed line), exact solution (solid line), observations (circles), and 95% confidence interval (shaded area).*

TABLE 3.1
Relative L_2 error (posterior mean vs. reference solution) of different methods for reconstructing KS equation solution.

Kriging	PhIK	CoPhIK	BiPhIK	CoBiPhIK
0.0786	0.1145	0.0674	0.1439	0.0791

because of the high sensitivity of the systems and the underestimate of the critical parameter α in the stochastic model, PhIK is less accurate than kriging with the specific settings in this example. The CoPhIK method has the potential to improve the accuracy of PhIK because it incorporates observation information in constructing the GP. In all methods, the periodic boundary condition is preserved, i.e., $\hat{y}(0) = \hat{y}(2\pi)$ for the reconstructed field $\hat{y}(\boldsymbol{x})$.

In summary, the numerical examples demonstrate the efficiency of bifidelity-based GP method in terms of cost of running physical model simulations. As we mentioned in section 2, the theoretical bounds are loose in many cases, and they are much larger than the actual values in the above examples (theoretical bounds are not

presented here) mainly because the condition number of covariance matrices is large. Besides, we also observe that in some cases, CoBiPhIK performs (slightly) better than CoPhIK when additional observations are included. This may be caused by the different selection of additional observation locations by the greedy algorithm.

4. Conclusion. In this work, we extend the PhIK/CoPhIK approach by combining two types of multifidelity methods in the BiPhIK/CoBiPhIK framework to reduce the computational cost of physical models simulations. Specifically, the approximation-theory-based bifidelity method is used to generate approximated high-fidelity realizations of the physical model, which are used to construct GP in PhIK and CoPhIK. The cokriging approach utilizes an auxiliary GP to describe the discrepancy between the model outputs and the sparse accurate observation data. We present the analysis of the difference between the posterior mean and variance in the resulting GPs by using BiPhIK/CoBiPhIK and PhIK/CoPhIK. We also analyze the accuracy of preserving linear physical constraints in the posterior mean of BiPhIK/CoBiPhIK.

The presented methods are nonintrusive and can utilize existing domain codes to compute the necessary realizations. Therefore, these methods are suitable for large-scale complex applications for which physical models and codes are available. Especially, when the parametric dependence of the low-fidelity model can well inform the structure of the high-fidelity model, the computational cost can be reduced dramatically.

Our future work would include two directions. One is to use advanced sampling strategies. For example, instead of MC, the probabilistic collocation method is used for the bifidelity method in [27, 51], which can further reduce the computational cost. The other direction is to directly approximate high-fidelity mean and covariance without generating approximated high-fidelity realizations as in [50]. This approach will save the cost of the lifting procedure because it only needs solving a much simpler linear system.

Appendix A. Adaptive sampling. In this work, we use a greedy algorithm to add additional observations, i.e., to add new observations at the maxima of $s(\boldsymbol{x})$, e.g., [7, 35]. Then, we can make a new prediction $\hat{y}(\boldsymbol{x})$ for $\boldsymbol{x} \in D$ and compute a new $\hat{s}^2(\boldsymbol{x})$ to select the next location for additional observation (see Algorithm A.1). This selection criterion is based on the statistical interpretation of the interpolation. More sophisticated sensor placement algorithms can be found in the literature, e.g., [15, 19, 9], and PhIK/BiPhIK or CoPhIK/CoBiPhIK are complementary to these methods.

Algorithm A.1 Active learning based on GPR.

1: Specify the locations $\boldsymbol{X}$, corresponding observations $\boldsymbol{y}$, and the maximum number of observations $N_{\max}$ affordable. The number of available observations is denoted as N.
2: **while** $N_{\max} > N$ **do**
3: Compute the MSE $\hat{s}^2(\boldsymbol{x})$ of prediction $\hat{y}(\boldsymbol{x})$ for $\boldsymbol{x} \in D$.
4: Locate the location $\boldsymbol{x}_m$ for the maximum of $\hat{s}^2(\boldsymbol{x})$ for $\boldsymbol{x} \in D$.
5: Obtain observation y_m at $\boldsymbol{x}_m$, and set $\boldsymbol{X} = \{\boldsymbol{X}, \boldsymbol{x}_m\}, \boldsymbol{y} = (\boldsymbol{y}^\top, y_m)^\top, N = N + 1$.
6: **end while**
7: Construct the prediction of $\hat{y}(\boldsymbol{x})$ on D using $\boldsymbol{X}$ and $\boldsymbol{y}$.

Appendix B. Proofs of Theorems 2.1 and 2.2. We present the following three lemmas.

LEMMA B.1. *For* $1 \le n \le N$,

$$\begin{aligned} \|k_H(\boldsymbol{x}, \boldsymbol{x}^{(n)})\| &\le \sigma_H(\Gamma)\sigma_H(\boldsymbol{x}^{(n)}), \\ \|k_B(\boldsymbol{x}, \boldsymbol{x}^{(n)})\| &\le \sigma_B(\Gamma)\sigma_B(\boldsymbol{x}^{(n)}). \end{aligned} \tag{B.1}$$

Proof. According to (2.19), we have

$$\begin{aligned} &\|k_H(\boldsymbol{x}, \boldsymbol{x}^{(n)})\| \\ &= \left\| \frac{1}{M-1} \sum_{m=1}^{M} \left(u_H^m(\boldsymbol{x}) - \mu_H(\boldsymbol{x})\right) \left(u_H^m(\boldsymbol{x}^{(n)}) - \mu_H(\boldsymbol{x}^{(n)})\right) \right\| \\ &\le \frac{1}{M-1} \sum_{m=1}^{M} \|u_H^m(\boldsymbol{x}) - \mu_H(\boldsymbol{x})\| \left| u_H^m(\boldsymbol{x}^{(n)}) - \mu_H(\boldsymbol{x}^{(n)}) \right| \\ &\le \frac{1}{M-1} \left(\sum_{m=1}^{M} \|u_H^m(\boldsymbol{x}) - \mu_H(\boldsymbol{x})\|^2 \right)^{\frac{1}{2}} \left(\sum_{m=1}^{M} \left| u_H^m(\boldsymbol{x}^{(n)}) - \mu_H(\boldsymbol{x}^{(n)}) \right|^2 \right)^{\frac{1}{2}} \\ &= \left(\frac{1}{M-1} \sum_{m=1}^{M} \|u_H^m(\boldsymbol{x}) - \mu_H(\boldsymbol{x})\|^2 \right)^{\frac{1}{2}} \left(\frac{1}{M-1} \sum_{m=1}^{M} \left| u_H^m(\boldsymbol{x}^{(n)}) - \mu_H(\boldsymbol{x}^{(n)}) \right|^2 \right)^{\frac{1}{2}} \\ &= \sigma_H(\Gamma)\sigma_H(\boldsymbol{x}^{(n)}). \end{aligned}$$

Similarly, $\|k_B(\boldsymbol{x}, \boldsymbol{x}^{(n)})\| \le \sigma_B(\Gamma)\sigma_B(\boldsymbol{x}^{(n)})$. ∎

LEMMA B.2. *For* $1 \le n \le N$,

$$\begin{aligned} &\left\| k_H(\boldsymbol{x}, \boldsymbol{x}^{(n)}) - k_B(\boldsymbol{x}, \boldsymbol{x}^{(n)}) \right\| \le \\ &\qquad \frac{2}{M-1} \sum_{m=1}^{M} \left(\delta_1 \left| u_B^m(\boldsymbol{x}^{(n)}) - \mu_B^m(\boldsymbol{x}^{(n)}) \right| + \delta_2 \|u_H^m(\boldsymbol{x}) - \mu_H(\boldsymbol{x})\| \right). \end{aligned} \tag{B.2}$$

Proof. For any $x \in D$,

$$\begin{aligned} \|\mu_H(\boldsymbol{x}) - \mu_B(\boldsymbol{x})\| &= \left\| \frac{1}{M} \sum_{m=1}^{M} u_H^m(\boldsymbol{x}) - \frac{1}{M} \sum_{m=1}^{M} u_B^m(\boldsymbol{x}) \right\| \\ &\le \frac{1}{M} \sum_{m=1}^{M} \|u_H^m(\boldsymbol{x}) - u_B^m(\boldsymbol{x})\| \le \delta_1, \end{aligned} \tag{B.3}$$

and

$$\begin{aligned} |\mu_H(\boldsymbol{x}) - \mu_B(\boldsymbol{x})| &= \left| \frac{1}{M} \sum_{m=1}^{M} u_H^m(\boldsymbol{x}) - \frac{1}{M} \sum_{m=1}^{M} u_B^m(\boldsymbol{x}) \right| \\ &\le \frac{1}{M} \sum_{m=1}^{M} |u_H^m(\boldsymbol{x}) - u_B^m(\boldsymbol{x})| \le \delta_2. \end{aligned} \tag{B.4}$$

Therefore,

$$
\begin{aligned}
&\left\|k_H(\boldsymbol{x},\boldsymbol{x}^{(n)})-k_B(\boldsymbol{x},\boldsymbol{x}^{(n)})\right\| \\
&=\frac{1}{M-1}\Bigg\|\sum_{m=1}^{M}\left(u_H^m(\boldsymbol{x})-\mu_H(\boldsymbol{x})\right)\left(u_H^m(\boldsymbol{x}^{(n)})-\mu_H(\boldsymbol{x}^{(n)})\right) \\
&\qquad -\sum_{m=1}^{M}\left(u_B^m(\boldsymbol{x})-\mu_B(\boldsymbol{x})\right)\left(u_B^m(\boldsymbol{x}^{(n)})-\mu_B(\boldsymbol{x}^{(n)})\right)\Bigg\| \\
&\le\frac{1}{M-1}\sum_{m=1}^{M}\Bigg\{\Bigg\|\left(u_H^m(\boldsymbol{x})-\mu_H(\boldsymbol{x})\right)\left(u_H^m(\boldsymbol{x}^{(n)})-\mu_H(\boldsymbol{x}^{(n)})\right) \\
&\qquad\qquad -\left(u_H^m(\boldsymbol{x})-\mu_H(\boldsymbol{x})\right)\left(u_B^m(\boldsymbol{x}^{(n)})-\mu_B(\boldsymbol{x}^{(n)})\right)\Bigg\| \\
&\qquad\quad +\Bigg\|\left(u_H^m(\boldsymbol{x})-\mu_H(\boldsymbol{x})\right)\left(u_B^m(\boldsymbol{x}^{(n)})-\mu_B(\boldsymbol{x}^{(n)})\right) \\
&\qquad\qquad -\left(u_B^m(\boldsymbol{x})-\mu_B(\boldsymbol{x})\right)\left(u_B^m(\boldsymbol{x}^{(n)})-\mu_B(\boldsymbol{x}^{(n)})\right)\Bigg\|\Bigg\} \\
&\le\frac{1}{M-1}\sum_{m=1}^{M}\Bigg\{\left\|u_H^m(\boldsymbol{x})-\mu_H(\boldsymbol{x})\right\|\left|u_H^m(\boldsymbol{x}^{(n)})-\mu_H(\boldsymbol{x}^{(n)})-u_B^m(\boldsymbol{x}^{(n)})+\mu_B(\boldsymbol{x}^{(n)})\right| \\
&\qquad\quad +\left|u_B^m(\boldsymbol{x}^{(n)})-\mu_B(\boldsymbol{x}^{(n)})\right|\left\|u_H^m(\boldsymbol{x})-\mu_H(\boldsymbol{x})-u_B^m(\boldsymbol{x})+\mu_B(\boldsymbol{x})\right\|\Bigg\} \\
&\le\frac{1}{M-1}\sum_{m=1}^{M}\left\{2\delta_2\left\|u_H^m(\boldsymbol{x})-\mu_H(\boldsymbol{x})\right\|+2\delta_1\left|u_B^m(\boldsymbol{x}^{(n)})-\mu_B(\boldsymbol{x}^{(n)})\right|\right\}. \qquad \square
\end{aligned}
$$

LEMMA B.3. *For* $1\le i,j\le N$,

$$
\left|k_H(\boldsymbol{x}^{(i)},\boldsymbol{x}^{(j)})-k_B(\boldsymbol{x}^{(i)},\boldsymbol{x}^{(j)})\right|\le 2\delta_2\sqrt{\frac{M}{M-1}}\left[\sigma_H(\boldsymbol{x}^{(i)})+\sigma_B(\boldsymbol{x}^{(j)})\right]. \tag{B.5}
$$

Proof.

$$
\begin{aligned}
&\left|k_H(\boldsymbol{x}^{(i)},\boldsymbol{x}^{(j)})-k_B(\boldsymbol{x}^{(i)},\boldsymbol{x}^{(j)})\right| \\
&=\frac{1}{M-1}\Bigg|\sum_{m=1}^{M}\left(u_H^m(\boldsymbol{x}^{(i)})-\mu_H(\boldsymbol{x}^{(i)})\right)\left(u_H^m(\boldsymbol{x}^{(j)})-\mu_H(\boldsymbol{x}^{(j)})\right) \\
&\qquad -\sum_{m=1}^{M}\left(u_B^m(\boldsymbol{x}^{(i)})-\mu_B(\boldsymbol{x}^{(i)})\right)\left(u_B^m(\boldsymbol{x}^{(j)})-\mu_B(\boldsymbol{x}^{(j)})\right)\Bigg| \\
&\le\frac{1}{M-1}\sum_{m=1}^{M}\Bigg\{\Bigg|\left(u_H^m(\boldsymbol{x}^{(i)})-\mu_H(\boldsymbol{x}^{(i)})\right)\left(u_H^m(\boldsymbol{x}^{(j)})-\mu_H(\boldsymbol{x}^{(j)})\right) \\
&\qquad\qquad -\left(u_H^m(\boldsymbol{x}^{(i)})-\mu_H(\boldsymbol{x}^{(i)})\right)\left(u_B^m(\boldsymbol{x}^{(j)})-\mu_B(\boldsymbol{x}^{(j)})\right)\Bigg| \\
&\qquad\quad +\Bigg|\left(u_H^m(\boldsymbol{x}^{(i)})-\mu_H(\boldsymbol{x}^{(i)})\right)\left(u_B^m(\boldsymbol{x}^{(j)})-\mu_B(\boldsymbol{x}^{(j)})\right) \\
&\qquad\qquad -\left(u_B^m(\boldsymbol{x}^{(i)})-\mu_B(\boldsymbol{x}^{(i)})\right)\left(u_B^m(\boldsymbol{x}^{(j)})-\mu_B(\boldsymbol{x}^{(j)})\right)\Bigg|\Bigg\}
\end{aligned}
$$

$$
\begin{aligned}
&\le \frac{1}{M-1}\sum_{m=1}^{M}\left\{\left|u_H^m(\boldsymbol{x}^{(i)})-\mu_H(\boldsymbol{x}^{(i)})\right|\left|u_H^m(\boldsymbol{x}^{(j)})-\mu_H(\boldsymbol{x}^{(j)})-u_B^m(\boldsymbol{x}^{(j)})+\mu_B(\boldsymbol{x}^{(j)})\right|\right.\\
&\qquad\left.+\left|u_B^m(\boldsymbol{x}^{(j)})-\mu_B(\boldsymbol{x}^{(j)})\right|\left|u_H^m(\boldsymbol{x}^{(i)})-\mu_H(\boldsymbol{x}^{(i)})-u_B^m(\boldsymbol{x}^{(i)})+\mu_B(\boldsymbol{x}^{(i)})\right|\right\}\\
&\le \frac{2\delta_2}{M-1}\sum_{m=1}^{M}\left\{\left|u_H^m(\boldsymbol{x}^{(i)})-\mu_H(\boldsymbol{x}^{(i)})\right|+\left|u_B^m(\boldsymbol{x}^{(j)})-\mu_B(\boldsymbol{x}^{(j)})\right|\right\}\\
&\le \frac{2\delta_2\sqrt{M}}{M-1}\left\{\left(\sum_{m=1}^{M}\left|u_H^m(\boldsymbol{x}^{(i)})-\mu_H(\boldsymbol{x}^{(i)})\right|^2\right)^{\frac{1}{2}}+\left(\sum_{m=1}^{M}\left|u_B^m(\boldsymbol{x}^{(j)})-\mu_B(\boldsymbol{x}^{(j)})\right|^2\right)^{\frac{1}{2}}\right\}\\
&=2\delta_2\sqrt{\frac{M}{M-1}}\left[\sigma_H(\boldsymbol{x}^{(i)})+\sigma_B(\boldsymbol{x}^{(j)})\right].
\end{aligned}
$$

∎

Now we prove Theorem 2.1 as follows.

Proof. Rewriting (2.21) in the functional form, we have

$$\hat{y}_H(\boldsymbol{x})=\mu_H(\boldsymbol{x})+\sum_{n=1}^{N}a_n^H k_H(\boldsymbol{x},\boldsymbol{x}^{(n)}), \tag{B.6}$$

where a_n^H is the nth entry of the vector $\mathbf{C}_H^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_H)$, $(\mathbf{C}_H)_{ij}=k_H(\boldsymbol{x}^{(i)},\boldsymbol{x}^{(j)})$, and $(\boldsymbol{\mu}_H)_i=\mu_H(\boldsymbol{x}^{(i)})$. Similarly, we have

$$\hat{y}_B(\boldsymbol{x})=\mu_B(\boldsymbol{x})+\sum_{n=1}^{N}a_n^B k_B(\boldsymbol{x},\boldsymbol{x}^{(n)}), \tag{B.7}$$

where a_n^B is the nth entry of the vector $\mathbf{C}_B^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_B)$, $(\mathbf{C}_B)_{ij}=k_B(\boldsymbol{x}^{(i)},\boldsymbol{x}^{(j)})$, and $(\boldsymbol{\mu}_B)_i=\mu_B(\boldsymbol{x}^{(i)})$. In (B.3), we show that

$$\|\mu_H(\boldsymbol{x})-\mu_B(\boldsymbol{x})\|\le\delta_1.$$

Next,

$$
\begin{aligned}
&\left\|a_n^H k_H(\boldsymbol{x},\boldsymbol{x}^{(n)})-a_n^B k_B(\boldsymbol{x},\boldsymbol{x}^{(n)})\right\|\\
&\le\left\|a_n^H k_H(\boldsymbol{x},\boldsymbol{x}^{(n)})-a_n^B k_H(\boldsymbol{x},\boldsymbol{x}^{(n)})\right\|+\left\|a_n^B k_H(\boldsymbol{x},\boldsymbol{x}^{(n)})-a_n^B k_B(\boldsymbol{x},\boldsymbol{x}^{(n)})\right\|\\
&\le\left|a_n^H-a_n^B\right|\left\|k_H(\boldsymbol{x},\boldsymbol{x}^{(n)})\right\|+\left|a_n^B\right|\left\|k_H(\boldsymbol{x},\boldsymbol{x}^{(n)})-k_B(\boldsymbol{x},\boldsymbol{x}^{(n)})\right\|\\
&\le\left|a_n^H-a_n^B\right|\sigma_H(\Gamma)\sigma_H(\boldsymbol{x}^{(n)})\\
&\quad+\frac{2\left|a_n^B\right|}{M-1}\sum_{m=1}^{M}\left(\delta_1\left|u_B^m(\boldsymbol{x}^{(n)})-\mu_B^m(\boldsymbol{x}^{(n)})\right|+\delta_2\left\|u_H^m(\boldsymbol{x})-\mu_H(\boldsymbol{x})\right\|\right),
\end{aligned}
$$

where the last inequality utilizes Lemmas B.1 and B.2. Therefore,

$$\begin{aligned}
&\left\|\sum_{n=1}^{N} a_n^H k_H(\boldsymbol{x},\boldsymbol{x}^{(n)}) - \sum_{n=1}^{N} a_n^B k_B(\boldsymbol{x},\boldsymbol{x}^{(n)})\right\| \\
&\le \underbrace{\sum_{n=1}^{N} \left|a_n^H - a_n^B\right| \sigma_H(\Gamma)\sigma_H(\boldsymbol{x}^{(n)})}_{J_1} + \underbrace{\frac{2\delta_1}{M-1}\sum_{n=1}^{N}\left|a_n^B\right|\sum_{m=1}^{M}\left|u_B^m(\boldsymbol{x}^{(n)}) - \mu_B^m(\boldsymbol{x}^{(n)})\right|}_{J_2} \\
&\quad + \underbrace{\frac{2\delta_2}{M-1}\sum_{n=1}^{N}\left|a_n^B\right|\sum_{m=1}^{M}\left\|u_H^m(\boldsymbol{x}) - \mu_H^m(\boldsymbol{x})\right\|}_{J_3}.
\end{aligned}$$

Here,

$$\begin{aligned}
J_1 &= \sigma_H(\Gamma)\sum_{n=1}^{N}\left|a_n^H - a_n^B\right|\sigma_H(\boldsymbol{x}^{(n)}) \\
&\le \sigma_H(\Gamma)\left(\sum_{n=1}^{N}\sigma_H^2(\boldsymbol{x}^{(n)})\right)^{\frac{1}{2}}\left(\sum_{n=1}^{N}\left|a_n^H - a_n^B\right|^2\right)^{\frac{1}{2}} \\
&= S_H\sigma_H(\Gamma)\left\|\mathbf{C}_H^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_H) - \mathbf{C}_B^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_B)\right\|_2 \\
&= S_H\sigma_H(\Gamma)\left\|\mathbf{C}_H^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_H) - \mathbf{C}_H^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_B) + \mathbf{C}_H^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_B) - \mathbf{C}_B^{-1}(\boldsymbol{y}-\boldsymbol{\mu}_B)\right\|_2 \\
&\le S_H\sigma_H(\Gamma)\left(\left\|\mathbf{C}_H^{-1}\right\|_2\left\|\boldsymbol{\mu}_H - \boldsymbol{\mu}_B\right\|_2 + \left\|\mathbf{C}_H^{-1} - \mathbf{C}_B^{-1}\right\|_2\left\|\boldsymbol{y}-\boldsymbol{\mu}_B\right\|_2\right) \\
&\le S_H\sigma_H(\Gamma)\left\{\left\|\mathbf{C}_H^{-1}\right\|_2\sqrt{N}\delta_2 + \left\|\mathbf{C}_H^{-1}\right\|_2^2\left\|\mathbf{C}_H - \mathbf{C}_B\right\|_2\left\|\boldsymbol{y}-\boldsymbol{\mu}_B\right\|_2\right\} \\
&\le S_H\sigma_H(\Gamma)\left\{\left\|\mathbf{C}_H^{-1}\right\|_2\sqrt{N}\delta_2 + \left\|\mathbf{C}_H^{-1}\right\|_2^2\left\|\mathbf{C}_H - \mathbf{C}_B\right\|_F\left\|\boldsymbol{y}-\boldsymbol{\mu}_B\right\|_2\right\},
\end{aligned}$$

where we use the well-known matrix perturbation conclusion (e.g., [3]):

$$\left\|(\mathbf{A}+\Delta\mathbf{A})^{-1} - \mathbf{A}^{-1}\right\| \lesssim \left\|\mathbf{A}^{-1}\right\|^2\left\|\Delta\mathbf{A}\right\| \tag{B.8}$$

for invertible matrices $\mathbf{A}$ and $\mathbf{A}+\Delta\mathbf{A}$, and a well-defined matrix norm $\|\cdot\|$ (assuming $\|\Delta A\|$ is much smaller than $\|\mathbf{A}^{-1}\|$). Further, using Lemma B.3, we have

$$\begin{aligned}
\left\|\mathbf{C}_H - \mathbf{C}_B\right\|_F^2 &= \sum_{i=1}^{N}\sum_{j=1}^{N}\left|k_H(\boldsymbol{x}^{(i)},\boldsymbol{x}^{(j)}) - k_B(\boldsymbol{x}^{(i)},\boldsymbol{x}^{(j)})\right|^2 \\
&\le \sum_{i=1}^{N}\sum_{j=1}^{N}\frac{4\delta_2^2 M}{M-1}\left[\sigma(u_H^m(\boldsymbol{x}^{(i)})) + \sigma(u_B^m(\boldsymbol{x}^{(j)}))\right]^2 \\
&\le \frac{8\delta_2^2 M}{M-1}\sum_{i=1}^{N}\sum_{j=1}^{N}\left[\sigma^2(u_H^m(\boldsymbol{x}^{(i)})) + \sigma^2(u_B^m(\boldsymbol{x}^{(j)}))\right] \\
&= \frac{8\delta_2^2 MN}{M-1}\sum_{n=1}^{N}\left[\sigma^2(u_H^m(\boldsymbol{x}^{(n)})) + \sigma^2(u_B^m(\boldsymbol{x}^{(n)}))\right] \\
&= \frac{8\delta_2^2 MN}{M-1}(S_H^2 + S_B^2)
\end{aligned} \tag{B.9}$$

which yields

$$J_1 \leq \sqrt{N} S_H \delta_2 \sigma_H(\Gamma) \left\|\mathbf{C}_H^{-1}\right\|_2 \left\{ 2\sqrt{\frac{2M}{M-1}} \left(S_H^2 + S_B^2\right)^{\frac{1}{2}} \left\|\mathbf{C}_H^{-1}\right\|_2 \|\boldsymbol{y} - \boldsymbol{\mu}_B\|_2 + 1 \right\}.$$

Also,

$$\begin{aligned} J_2 \leq & 2\delta_1 \sqrt{\frac{M}{M-1}} \sum_{n=1}^{N} \left|a_n^B\right| \sigma(u_B^m(\boldsymbol{x}^{(n)})) \\ \leq & 2\delta_1 \sqrt{\frac{MN}{M-1}} \left\|\mathbf{C}_B^{-1}(\boldsymbol{y} - \boldsymbol{\mu}_B)\right\|_2 \left(\sum_{n=1}^{N} \sigma^2(u_B^m(\boldsymbol{x}^{(n)})) \right)^{\frac{1}{2}} \\ \leq & 2\delta_1 \sqrt{\frac{MN}{M-1}} \left\|\mathbf{C}_B^{-1}\right\|_2 \|\boldsymbol{y} - \boldsymbol{\mu}_B\|_2 S_B. \end{aligned}$$

Similarly,

$$\begin{aligned} J_3 \leq & 2\delta_2 \sqrt{\frac{M}{M-1}} \sum_{n=1}^{N} \left|a_n^B\right| \sigma_H(\Gamma) \\ \leq & 2\delta_2 \sqrt{\frac{MN}{M-1}} \sigma_H(\Gamma) \left\|\mathbf{C}_B^{-1}(\boldsymbol{y} - \boldsymbol{\mu}_B)\right\|_2 \\ \leq & 2\delta_2 \sqrt{\frac{MN}{M-1}} \sigma_H(\Gamma) \left\|\mathbf{C}_B^{-1}\right\|_2 \|\boldsymbol{y} - \boldsymbol{\mu}_B\|_2 . \end{aligned}$$

Therefore, the conclusion holds. ∎

Next, we present the proof of Theorem 2.2.

Proof. For any $\boldsymbol{x}^* \in D$, we use the following concise notations:

$$\begin{aligned} \boldsymbol{c}_H &= \left(k_H(\boldsymbol{x}^{(1)}, \boldsymbol{x}^*), \ldots, k_H(\boldsymbol{x}^{(N)}, \boldsymbol{x}^*)\right)^\top, \\ \boldsymbol{c}_B &= \left(k_B(\boldsymbol{x}^{(1)}, \boldsymbol{x}^*), \ldots, k_B(\boldsymbol{x}^{(N)}, \boldsymbol{x}^*)\right)^\top. \end{aligned} \tag{B.10}$$

Following the same procedure in the proof of Lemma B.3, we have

$$\begin{aligned} |k_H(\boldsymbol{x}^*, \boldsymbol{x}^*) - k_B(\boldsymbol{x}^*, \boldsymbol{x}^*)| &\leq 2\delta_2 \sqrt{\frac{M}{M-1}} \left[\sigma(u_H^m(\boldsymbol{x}^*)) + \sigma(u_B^m(\boldsymbol{x}^*))\right] \\ &\leq 2\delta_2 \sqrt{\frac{2M}{M-1}} (\Delta_H^2 + \Delta_B^2)^{\frac{1}{2}}. \end{aligned}$$

According to Lemma B.3,

$$\begin{aligned} \|\boldsymbol{c}_H - \boldsymbol{c}_B\|_2 &= \left(\sum_{n=1}^{N} \left| k_H(\boldsymbol{x}^{(n)}, \boldsymbol{x}^*) - k_B(\boldsymbol{x}^{(n)}, \boldsymbol{x}^*) \right|^2 \right)^{\frac{1}{2}} \\ &\leq 2\delta_2 \sqrt{\frac{M}{M-1}} \left(\sum_{n=1}^{N} \left[\sigma(u_H^m(\boldsymbol{x}^{(n)})) + \sigma(u_B^m(\boldsymbol{x}^*)) \right]^2 \right)^{\frac{1}{2}} \\ &\leq 2\delta_2 \sqrt{\frac{2M}{M-1}} \left(\sum_{n=1}^{N} \left[\sigma^2(u_H^m(\boldsymbol{x}^{(n)})) + \sigma^2(u_B^m(\boldsymbol{x}^*)) \right] \right)^{\frac{1}{2}} \\ &= 2\delta_2 \sqrt{\frac{2M}{M-1}} \left(S_H^2 + S_B^2\right)^{\frac{1}{2}}. \end{aligned}$$

Also,

(B.11)
$$\begin{aligned}\|\boldsymbol{c}_H\|_2 =& \left\{\sum_{n=1}^{N}\left|k_H(\boldsymbol{x}^{(n)},\boldsymbol{x}^*)\right|^2\right\}^{\frac{1}{2}}\\ \le& \left\{\sum_{n=1}^{N}\left|\frac{1}{M-1}\sum_{m=1}^{M}\left(u_H^m(\boldsymbol{x}^{(n)})-\mu_H(\boldsymbol{x}^{(n)})\right)\left(u_H^m(\boldsymbol{x}^*)-\mu_H(\boldsymbol{x}^*)\right)\right|^2\right\}^{\frac{1}{2}}\\ \le& \left\{\sum_{n=1}^{N}\left(\frac{1}{M-1}\sum_{m=1}^{M}\left(u_H^m(\boldsymbol{x}^{(n)})-\mu_H(\boldsymbol{x}^{(n)})\right)^2\right)\cdot\right.\\ &\left.\left(\frac{1}{M-1}\sum_{m=1}^{M}\left(u_H^m(\boldsymbol{x}^*)-\mu_H(\boldsymbol{x}^*)\right)^2\right)\right\}^{\frac{1}{2}}\\ =&\sigma_H(\boldsymbol{x}^*)S_H.\end{aligned}$$

Similarly,

(B.12)
$$\|\boldsymbol{c}_B\|_2 \le \sigma_B(\boldsymbol{x}^*)S_B.$$

Thus,

(B.13)
$$\begin{aligned}&\left|\boldsymbol{c}_H^\top\mathbf{C}_H^{-1}\boldsymbol{c}_H-\boldsymbol{c}_B^\top\mathbf{C}_B^{-1}\boldsymbol{c}_B\right|\\ \le&\left|\boldsymbol{c}_H^\top\mathbf{C}_H^{-1}\boldsymbol{c}_H-\boldsymbol{c}_H^\top\mathbf{C}_H^{-1}\boldsymbol{c}_B\right|+\left|\boldsymbol{c}_H^\top\mathbf{C}_H^{-1}\boldsymbol{c}_B-\boldsymbol{c}_H^\top\mathbf{C}_B^{-1}\boldsymbol{c}_B\right|+\left|\boldsymbol{c}_H^\top\mathbf{C}_B^{-1}\boldsymbol{c}_B-\boldsymbol{c}_B^\top\mathbf{C}_B^{-1}\boldsymbol{c}_B\right|\\ \le&\underbrace{\|\boldsymbol{c}_H\|_2\|\mathbf{C}_H^{-1}\|_2\|\boldsymbol{c}_H-\boldsymbol{c}_B\|_2}_{J_1}+\underbrace{\|\boldsymbol{c}_H\|_2\|\mathbf{C}_H^{-1}-\mathbf{C}_B^{-1}\|_F\|\boldsymbol{c}_B\|_2}_{J_2}\\ &+\underbrace{\|\boldsymbol{c}_B\|_2\|\mathbf{C}_B^{-1}\|_2\|\boldsymbol{c}_H-\boldsymbol{c}_B\|_2}_{J_3},\end{aligned}$$

and

$$\begin{aligned}J_1 &\le 2\delta_2\sqrt{\frac{2M}{M-1}}\sigma_H(\boldsymbol{x}^*)S_H(S_B^2+S_H^2)^{\frac{1}{2}}\|\mathbf{C}_H^{-1}\|_2,\\ J_2 &\le 2\delta_2\sqrt{\frac{2MN}{M-1}}\sigma_H(\boldsymbol{x}^*)\sigma_B(\boldsymbol{x}^*)S_HS_B(S_H^2+S_B^2)^{\frac{1}{2}}\|\mathbf{C}_H^{-1}\|_2^2,\\ J_3 &\le 2\delta_2\sqrt{\frac{2M}{M-1}}\sigma_B(\boldsymbol{x}^*)S_B(S_B^2+S_H^2)^{\frac{1}{2}}\|\mathbf{C}_B^{-1}\|_2,\end{aligned}$$

where in the inequality of J_2 we use (B.8) and (B.9). Finally, using the fact $\sigma_H(\boldsymbol{x}^*) \le \Delta_H$ and $\sigma_B(\boldsymbol{x}^*) \le \Delta_B$, we have

(B.14)
$$S_H \le \sqrt{N}\Delta_H, \quad S_B \le \sqrt{N}\Delta_B,$$

which indicates

$$\begin{aligned}J_1 &\le 2N\delta_2\sqrt{\frac{2M}{M-1}}\Delta_H^2(\Delta_B^2+\Delta_H^2)^{\frac{1}{2}}\|\mathbf{C}_H^{-1}\|_2,\\ J_2 &\le 2N\delta_2\sqrt{\frac{2MN}{M-1}}\Delta_H^2\Delta_B^2(\Delta_H^2+\Delta_B^2)^{\frac{1}{2}}\|\mathbf{C}_H^{-1}\|_2^2,\\ J_3 &\le 2N\delta_2\sqrt{\frac{2M}{M-1}}\Delta_B^2(\Delta_B^2+\Delta_H^2)^{\frac{1}{2}}\|\mathbf{C}_B^{-1}\|_2.\end{aligned}$$

Therefore, the conclusion holds. ∎

REFERENCES

[1] C. J. Brooks, A. Forrester, A. Keane, and S. Shahpar, *Multi-fidelity design optimisation of a transonic compressor rotor*, in Proceedings of the 9th European Conference on Turbomachinery, 2011.

[2] K. A. Cliffe, M. B. Giles, R. Scheichl, and A. L. Teckentrup, *Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients*, Comput. Vis. Sci., 14 (2011), 3.

[3] J. Demmel, *The componentwise distance to the nearest singular matrix*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 10–19.

[4] A. Doostan and H. Owhadi, *A non-adapted sparse approximation of PDEs with stochastic inputs*, J. Comput. Phys., 230 (2011), pp. 3015–3034.

[5] M. S. Eldred, L. W. Ng, M. F. Barone, and S. P. Domino, *Multifidelity uncertainty quantification using spectral stochastic discrepancy models*, in Handbook of Uncertainty Quantification, R. Ghanem, D. Higon, and H. Owhadi, eds., Springer, Cham, 2017, pp. 991–1036.

[6] H. R. Fairbanks, A. Doostan, C. Ketelsen, and G. Iaccarino, *A low-rank control variate for multilevel Monte Carlo simulation of high-dimensional uncertain systems*, J. Comput. Phys., 341 (2017), pp. 121–139.

[7] A. Forrester, A. Sóbester, and A. Keane, *Engineering Design via Surrogate Modelling: A Practical Guide*, John Wiley & Sons, Hoboken, NJ, 2008.

[8] A. I. Forrester, A. Sóbester, and A. J. Keane, *Multi-fidelity optimization via surrogate modelling*, in Proc. A, 463 (2007), pp. 3251–3269.

[9] R. Garnett, M. A. Osborne, and S. J. Roberts, *Bayesian optimization for sensor set selection*, in Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, ACM, 2010, pp. 209–219.

[10] M. B. Giles, *Multilevel Monte Carlo path simulation*, Oper. Res., 56 (2008), pp. 607–617.

[11] J. M. Hyman and B. Nicolaenko, *The Kuramoto-Sivashinsky equation: A bridge between PDE's and dynamical systems*, Phys. D, 18 (1986), pp. 113–126.

[12] J. M. Hyman, B. Nicolaenko, and S. Zaleski, *Order and complexity in the Kuramoto-Sivashinsky model of weakly turbulent interfaces*, Phys. D, 23 (1986), pp. 265–292.

[13] L. Jofre, G. Geraci, H. Fairbanks, A. Doostan, and G. Iaccarino, *Multi-Fidelity Uncertainty Quantification of Irradiated Particle-Laden Turbulence*, https://arxiv.org/abs/1801.06062, 2018.

[14] B. A. Jones and R. Weisman, *Multi-fidelity orbit uncertainty propagation*, Acta Astronautica, 155 (2018), pp. 406–417.

[15] D. R. Jones, M. Schonlau, and W. J. Welch, *Efficient global optimization of expensive black-box functions*, J. Glob. Optim., 13 (1998), pp. 455–492.

[16] M. C. Kennedy and A. O'Hagan, *Predicting the output from a complex computer code when fast approximations are available*, Biometrika, 87 (2000), pp. 1–13.

[17] P. K. Kitanidis, *Introduction to Geostatistics: Applications in Hydrogeology*, Cambridge University Press, Cambridge, UK, 1997.

[18] M. Knotters, D. Brus, and J. O. Voshaar, *A comparison of kriging, co-kriging and kriging combined with regression for spatial interpolation of horizon depth with censored observations*, Geoderma, 67 (1995), pp. 227–246.

[19] A. Krause, A. Singh, and C. Guestrin, *Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies*, J. Mach. Learn. Res., 9 (2008), pp. 235–284.

[20] Y. Kuramoto, *Diffusion-induced chaos in reaction systems*, Progress Theoret. Phys. Suppl., 64 (1978), pp. 346–367.

[21] J. Laurenceau and P. Sagaut, *Building efficient response surfaces of aerodynamic functions with kriging and cokriging*, AIAA J., 46 (2008), pp. 498–507.

[22] L. Le Gratiet and J. Garnier, *Recursive co-kriging model for design of computer experiments with multiple levels of fidelity*, Int. J. Uncertain. Quant., 4 (2014).

[23] J. Li and P. Stinis, *A unified framework for mesh refinement in random and physical space*, J. Comput. Phys., 323 (2016), pp. 243–264.

[24] X. Ma and N. Zabaras, *An adaptive high-dimensional stochastic model representation technique for the solution of stochastic partial differential equations*, J. Comput. Phys., 229 (2010), pp. 3884–3915.

[25] D. J. MacKay, *Introduction to Gaussian processes*, NATO Adv. Study Inst. Ser. F: Comput. Systems Sci., 168 (1998), pp. 133–166.

[26] R. Munipalli, J. Hamilton, and X. Zhu, *A multifidelity approach to parameter dependent modeling of combustion instability*, in Proceedings of the 2018 Joint Propulsion Conference, 2018, 4557.

[27] A. Narayan, C. Gittelson, and D. Xiu, *A stochastic collocation algorithm with multifidelity models*, SIAM J. Sci. Comput., 36 (2014), pp. A495–A521.
[28] L. W.-T. Ng and M. Eldred, *Multifidelity uncertainty quantification using non-intrusive polynomial chaos and stochastic collocation*, in Proceedings of the 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA, 2012, 1852.
[29] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM, Philadelphia, PA, 1992.
[30] W. Pan, X. Yang, J. Bao, and M. Wang, *Optimizing discharge capacity of Li-O_2 batteries by design of air-electrode porous structure: Multifidelity modeling and optimization*, J. Electrochem. Soc., 164 (2017), pp. E3499–E3511.
[31] B. Peherstorfer, T. Cui, Y. Marzouk, and K. Willcox, *Multifidelity importance sampling*, Comput. Methods Appl. Mech. Engrg., 300 (2016), pp. 490–509.
[32] B. Peherstorfer, K. Willcox, and M. Gunzburger, *Optimal model management for multifidelity Monte Carlo estimation*, SIAM J. Sci. Comput., 38 (2016), pp. A3163–A3194.
[33] P. Perdikaris, M. Raissi, A. Damianou, N. Lawrence, and G. E. Karniadakis, *Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling*, Proc. A, 473 (2017), 20160751.
[34] P. Perdikaris, D. Venturi, J. Royset, and G. Karniadakis, *Multi-fidelity modelling via recursive co-kriging and Gaussian–Markov random fields*, Proc. A, 471 (2015), 20150018.
[35] M. Raissi, P. Perdikaris, and G. E. Karniadakis, *Machine learning of linear differential equations using Gaussian processes*, J. Comput. Phys., 348 (2017), pp. 683–693.
[36] M. Razi, A. Narayan, R. Kirby, and D. Bedrov, *Fast predictive models based on multi-fidelity sampling of properties in molecular dynamics simulations*, Comput. Materials Sci., 152 (2018), pp. 125–133.
[37] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn, *Design and analysis of computer experiments*, Statist. Sci., 4 (1989), pp. 409–423.
[38] G. Sivashinsky, *On flame propagation under conditions of stoichiometry*, SIAM J. Appl. Math., 39 (1980), pp. 67–82.
[39] A. Stein and L. Corsten, *Universal kriging and cokriging as a regression procedure*, Biometrics, 47 (1991), pp. 575–587.
[40] M. L. Stein, *Interpolation of Spatial Data: Some Theory for Kriging*, Springer Science & Business Media, New York, NY, 2012.
[41] A. M. Tartakovsky, M. Panzeri, G. D. Tartakovsky, and A. Guadagnini, *Uncertainty quantification in scale-dependent models of flow in porous media*, Water Resources Res., 53 (2017), pp. 9392–9401.
[42] M. A. Tatang, *Direct Incorporation of Uncertainty in Chemical and Environmental Engineering Systems*, PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, 1995.
[43] R. Tuo, C. J. Wu, and D. Yu, *Surrogate modeling of computer experiments with different mesh densities*, Technometrics, 56 (2014), pp. 372–380.
[44] C. K. Williams and C. E. Rasmussen, *Gaussian Processes for Machine Learning*, The MIT Press, Cambridge, MA, 2006.
[45] D. Xiu and J. S. Hesthaven, *High-order collocation methods for differential equations with random inputs*, SIAM J. Sci. Comput., 27 (2005), pp. 1118–1139.
[46] X. Yang, D. Barajas-Solano, G. Tartakovsky, N. Baker, and A. Tartakovsky, *Physics-informed CoKriging: A Gaussian-process-regression-based multifidelity method for data-model convergence*, J. Comput. Phys., 395 (2019), pp. 410–431.
[47] X. Yang, M. Choi, G. Lin, and G. E. Karniadakis, *Adaptive ANOVA decomposition of stochastic incompressible and compressible flows*, J. Comput. Phys., 231 (2012), pp. 1587–1614.
[48] X. Yang and G. E. Karniadakis, *Reweighted ℓ_1 minimization method for stochastic elliptic differential equations*, J. Comput. Phys., 248 (2013), pp. 87–108.
[49] X. Yang, G. Tartakovsky, and A. Tartakovsky, *Physics-Informed Kriging: A Physics-Informed Gaussian Process Regression Method for Data-Model Convergence*, https://arxiv.org/abs/1809.03461, 2018.
[50] X. Zhu, E. M. Linebarger, and D. Xiu, *Multi-fidelity stochastic collocation method for computation of statistical moments*, J. Comput. Phys., 341 (2017), pp. 386–396.
[51] X. Zhu, A. Narayan, and D. Xiu, *Computational aspects of stochastic collocation with multifidelity models*, SIAM/ASA J. Uncertain. Quantif., 2 (2014), pp. 444–463.