Statistics

A Journal of Theoretical and Applied Statistics

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/gsta20

Semi-parametric adjustment to computer models

Yan Wang & Rui Tuo

To cite this article: Yan Wang & Rui Tuo (2020) Semi-parametric adjustment to computer models, Statistics, 54:6, 1255-1275, DOI: 10.1080/02331888.2020.1862113

To link to this article: https://doi.org/10.1080/02331888.2020.1862113

STATISTICS
2020, VOL. 54, NO. 6, 1255–1275
https://doi.org/10.1080/02331888.2020.1862113

Taylor & Francis
Taylor & Francis Group

Semi-parametric adjustment to computer models

Yan Wang[a] and Rui Tuo[b]

[a]Faculty of Science, College of Statistics and Data Science, Beijing University of Technology, Beijing, People's Republic of China; [b]Department of Industrial and Systems Engineering, Texas A&M University, College Station, TX, USA

ABSTRACT

Computer simulations are widely used in scientific exploration and engineering designs. However, computer outputs usually do not match the reality perfectly because the computer models are built under certain simplifications and approximations. When physical observations are also available, statistical methods can be applied to estimate the discrepancy between the computer output and the physical response. In this article, we propose a semi-parametric method for statistical adjustments to computer models. The proposed method is proven to enjoy nice theoretical properties. We use three numerical studies and a real example to examine the predictive performance of the proposed method. The results show that the proposed method outperforms existing methods.

ARTICLE HISTORY
Received 22 February 2019
Accepted 1 December 2020

KEYWORDS
Computer experiments; reproducing kernel Hilbert spaces; prediction; uncertainty quantification

2010 MATHEMATICS SUBJECT CLASSIFICATION
60G15

1. Introduction

Design of experiments is a powerful tool in understanding complex systems. We refer to Wu and Hamada [1] for an introduction to this area. However, it can be difficult or even infeasible to study some physical systems via conventional experimental design and analysis procedures, because these physical experiments are expensive and time consuming to conduct. The advances in numerical algorithms and computational techniques allow us to study some complex physical systems with computer simulations which are much less costly. Computer simulations have been applied successfully in many areas, like the research of hydrocarbon reservoir Craig et al. [2], food and environment Boukouvalas et al. [3], virtual brain tumours in health and medicine Drignei [4], fluidized-bed coating in materialogy Wang et al. [5], the electrical activity of myocytes in cell biology Plumlee et al. [6] and so on.

A major problem of computer simulation is that, the simulation outputs from a computer model do not usually match the corresponding physical responses perfectly, i.e. there exists a *discrepancy* between the computer output and the physical response. This is because the computer models are built based on certain simplifications and assumptions which do not always hold in reality. For example, a solver to a set of partial differential equations (PDEs) for a physical process may not give an accurate answer when the input

CONTACT Yan Wang yanwang@bjut.edu.cn University of Chinese Academy of Sciences, Beijing 100190, People's Republic of China

initial or boundary conditions are not correct. In the presence of this discrepancy, computer simulation becomes less reliable. Statistical adjustment plays an important role in improving the accuracy of the computer models. Its main idea is to make use of the physical observations, and correct the discrepancy by building a surrogate model with the physical observations.

Existing statistical adjustment methodologies for computer models can be roughly divided into two groups: parametric and non-parametric methods. Parametric methods are more classic. Roache [7], Trucano et al. [8], Xiong et al. [9] estimate the discrepancy between the physical system and the computer model by building a regression model. Joseph and Melkote [10] propose an engineering-driven parametric model to capture the discrepancy. Marzouk and Xiu [11] parameterize the discrepancy by generalized polynomial chaos which is an orthogonal approximation to random functions.

Non-parametric methods are widely used to adjust the computer models. Most of the existing non-parametric methods are based on Gaussian Process models, which can be regarded as a non-parametric Bayesian approach [12]. Kennedy and O'Hagan [13] propose a Bayesian method to compute the posterior prediction distribution of the physical process. This method is abbreviated as KO's method. A number of modifications and extensions of the KO's method have been proposed in the literature. Here we review a few of them. Higdon et al. [14] and Bayarri et al. [15] predict the physical process by using the posterior estimation of the discrepancy between the computer model and the physical response. Bayarri et al. [16] use a wavelet decomposition of both the computer model and the discrepancy function, and then estimate this coefficients using the maximum likelihood method. Qian and Wu [17] extend the KO's method by replacing a real-valued regression coefficients in the KO's model by a general linear regression function. Chang and Joseph [18] consider a different kind of experimental error, which may occur randomly during the physical experiment, and use a non-parametric method to get the posterior distribution of the physical process. Joseph and Yan [19] propose an engineering-driven adjustment model which applies a transformation to the input of the computer model, so that the computer outputs can better match the physical responses.

In this article, we propose a semi-parametric statistical adjustment method, which enjoys the advantages of both the parametric and non-parametric methods for reducing the discrepancy. The methodology is inspired by the L_2 estimation method proposed by Tuo and Wu [20]. We model the true discrepancy in a semi-parametric manner, while assuming that its main trend can be captured using a linear combination of a finite set of regression functions. We use theoretical analysis, numerical simulations and real data study to show that the proposed method estimates these regression coefficients more efficiently.

This article is organized as follows. In Section 2, we review the existing discrepancy-reducing methods and demonstrate their advantages and disadvantages. In Section 3, we propose a new statistical adjustment methodology for computer simulations, called the *semi-parametric adjustment*. In Section 4, the asymptotic behaviours of the proposed method are studied. In Section 5, we compare the proposed method with existing ones in three numerical examples and a Laser-Assisted Mechanical Micro-machining(LAMM) example.

2. Backgrounds

Denote the input domain of the physical experiment by Ω, which is assumed to be a convex and compact subset of R^d. Let $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subseteq \Omega$ be the set of design points for the physical experiment, where $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{id})^T$ denote the vector of the ith input variable, and $\boldsymbol{Y}^p = \{y_1^p, \ldots, y_n^p\}$ the corresponding physical responses.

In general, we suppose the physical responses come from the following model

$$y_i^p = \zeta(\boldsymbol{x}_i) + e_i, 1 \leq i \leq n, \tag{1}$$

where the errors e_i's are independent and identically distributed following $N(0, \sigma^2)$ with unknown $\sigma^2 > 0$, $\zeta(\cdot)$ is an underlying deterministic function, which is referred to as the *true process* [13].

Suppose our goal is to reconstruct the function $\zeta(\cdot)$. Sometimes this could be done by data-driven methods, such as the traditional methods in design of experiments. However, when the physical experiment is very costly to conduct, a full data-driven approach may be unrealistic to carry out, especially when $\zeta(\cdot)$ is highly nonlinear or high dimensional. Another possible solution is to seek for a knowledge-based approach, such as a computer simulation. Computer simulators are built using scientific knowledge. For example, most computational fluid dynamics (CFD) problems can be simulated by solving the Navier-Stokes equations. By using a computer simulation, we may avoid the expensive physical runs and save the cost of the experiment. In this work, we assume that there is a computer code available to simulation the physical responses.

Computer simulations also have some limitations. The computer simulation models are usually built based on assumptions and simplifications which do not hold true in reality. Consequently, the computer simulation outputs normally cannot match the physical responses perfectly. The difference between the computer outputs and the physical responses is known as the discrepancy. Sometimes, the discrepancy can be large so that the accuracy of the computer simulation does not meet the requirements for practical use.

Statistical adjustment methods tackle this problem by combining the knowledge-based and data-driven approaches. Although the computer outputs and the physical responses are not the same in most scenarios, the computer model can usually capture the main trend of the physical response surface. In this case, the discrepancy function between the physical response surface and the computer output surface is usually much 'simpler' to model than the original physical process and can thus be estimated using a few data points.

Denote the computer model by $y^s(\cdot)$, and assume that the computer code is deterministic. Let $\boldsymbol{X}^s = \{\boldsymbol{x}_1^s, \ldots, \boldsymbol{x}_N^s\} \subseteq \Omega$ be the set of design points for the computer experiment, where $\boldsymbol{x}_i^s = (x_{i1}^s, \ldots, x_{id}^s)^T$ denote the vector of the ith input variable, and $\boldsymbol{Y}^s = \{y_1^s, \ldots, y_N^s\}$ the corresponding computer outputs. In this paper, we consider two types of computer simulations. Following the terminologies introduced by Tuo and Wu [21], we call a computer code *cheap* if each run of the computer code costs very little so that we can regard $y^s(\cdot)$ as a known function. In contrast, we call the computer code *expensive* when the computer code is costly so that we can only run the code with a limited number of input points. In this section, we suppose $y^s(\cdot)$ is cheap. This assumption will be relaxed in Section 3.

The most important step in statistical adjustments is how to estimate $\zeta(\cdot) - y^s(\cdot)$, the discrepancy function between physical system and computer model. There are two major

approaches: non-parametric and parametric. We call a statistical adjustment approach parametric, if $\zeta(\cdot)$ is modelled in a parametric way, i.e. $\zeta(\cdot)$ belongs to a set of functions indexed by a finite number of parameters. Otherwise, we call the approach non-parametric.

2.1. Review of parametric methods

In general, a parametric model for $\zeta(\cdot)$ can be expressed as

$$\zeta(\boldsymbol{x}) = h(\boldsymbol{x}, y^s, \boldsymbol{\beta}), \tag{2}$$

where h is a *known* function and $\boldsymbol{\beta}$ is a vector of unknown parameters. It is common to choose h as a linear combination of certain regressors, i.e.

$$\zeta(\boldsymbol{x}) = \beta_0 y^s(\boldsymbol{x}) + \sum_{i=1}^{m} \beta_i f_i(\boldsymbol{x}), \tag{3}$$

where $\{f_1, \ldots, f_m\}$ is a pre-specified set of basis functions, and $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_m)$ is a vector of regression coefficients to be estimated. A natural estimator for the parametric models is the least squares estimator.

Roache [7], Trucano et al. [8] and Xiong et al. [9] adjust the computer outputs by using the regression model (3). Marzouk and Xiu [11] parameterize the discrepancy by using the generalized polynomial chaos. Driven by engineering knowledge, Joseph and Melkote [10] propose more complicated estimation approaches. These methods can be regarded as variations of the general parametric model (2).

2.2. Review of non-parametric methods

A prevalent non-parametric discrepancy-reducing method is KO's method proposed by Kennedy and O'Hagan [13], which employs a hierarchical Bayesian approach. The true process is modelled as

$$\zeta(\cdot) = \rho y^s(\cdot) + \delta(\cdot), \tag{4}$$

where ρ is an unknown regression coefficient and $\delta(\cdot)$ is called the *discrepancy function*. Then $\delta(\cdot)$ is modelled as a realization of a stationary Gaussian process:

$$\delta(\cdot) \sim GP(0, \tau^2 \Phi(\cdot)), \tag{5}$$

with the variance τ^2 and correlation function Φ. Common choices of Φ include the Gaussian correlation functions with

$$\Phi(\boldsymbol{x}_i, \boldsymbol{x}_j; \phi) = \exp(-\phi \| \boldsymbol{x}_i - \boldsymbol{x}_j \|^2), \tag{6}$$

and the Matérn correlation functions with

$$\Phi(\boldsymbol{x}_i, \boldsymbol{x}_j; \nu, \phi) = \frac{1}{\Gamma(\nu)} \left(\frac{2\sqrt{\nu} \| \boldsymbol{x}_i - \boldsymbol{x}_j \|}{\phi} \right)^{\nu} K_\nu \left(\frac{2\sqrt{\nu} \| \boldsymbol{x}_i - \boldsymbol{x}_j \|}{\phi} \right), \tag{7}$$

where $\phi > 0$ is the correlation parameter and K_ν denotes the modified Bessel function of the second kind with order ν. We refer to Santner et al. [22], Rasmussen [12] and Stein [23]

for more discussions about Gaussian process modelling. In KO's method, the prediction for the true process reduces to calculating the posterior distribution of $\zeta(\cdot)$. Higdon et al. [14] and Bayarri et al. [15] set $\rho = 1$ and predict the true process $\zeta(\cdot)$ at any given $\boldsymbol{x}_0$ by $y^s(\boldsymbol{x}_0) + \hat{\delta}(\boldsymbol{x}_0)$, where $\hat{\delta}(\boldsymbol{x}_0)$ is the posterior estimation of $\delta(\boldsymbol{x})$ on $\boldsymbol{x}_0$.

2.3. Comparation between parametric and non-parametric methods

In this part, we compare the parametric and non-parametric methods and demonstrate their advantages and disadvantages. The parametric methods suffer from a glaring deficiency: the assumptions of the these models are usually unacceptably strong. Compared with the parametric models, the model assumptions of the non-parametric approaches are much relaxed. Although KO's method is consistent for prediction, it is less accurate when the sample size of physical observations is small, because the rate of convergence of KO's method is much slower than the standard rate of parametric methods $O_p(n^{-1/2})$. See Theorem 3 of Tuo and Wu [24].

Now we illustrate the above points in a numerical study. Consider the following simple example. Suppose the true process is

$$\zeta(x) = x + 0.12\sin(2\pi x), x \in [0, 1]. \tag{8}$$

The physical observations are obtained by (1) with $\sigma = 0.12$ and the design point $x_i = (i-1)/(n-1), i = 1, \ldots, n$. Suppose the computer model is

$$y^s(x) = \sin(\frac{\pi}{2}x), x \in [0, 1]. \tag{9}$$

The parametric and non-parametric methods are considered.

- Parametric method:

  We use the standard linear regression model to predict the true process.

  $$\hat{\zeta}(x) = \hat{\beta}_0 y^s + \hat{\beta}_1 + \hat{\beta}_2 x, \tag{10}$$

  where $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1)^T$ can be obtained from the OLS estimator

  $$\hat{\boldsymbol{\beta}}_n^{\text{OLS}} = \underset{\boldsymbol{\beta}}{\text{argmin}} \frac{1}{n}\sum_{i=1}^{n}\left(y^p(x_i) - \beta_0 y^s(x_i) - \beta_1 - \beta_2 x_i\right)^2. \tag{11}$$

- Non-parametric method:

  Ordinary Kriging model with Matérn correlation function with smoothness parameter $\nu = 5/2$ is used to fit the discrepancy between the true process and the computer model. We choose a Matérn correlation function instead of a Gaussian correlation function because (1) To ensure the theoretical guarantees of the proposed method. See Sections 4 and 6 for more discussions about this reason. (2) The left and right subfigure of Figure 1 show 10 realizations of $GP(0, \Phi(\cdot))$, respectively, where Φ is a Matérn correlation function $\Phi(\boldsymbol{x}_i, \boldsymbol{x}_j; \nu = 5/2, \phi = 1)$ (left) and a Matérn correlation function

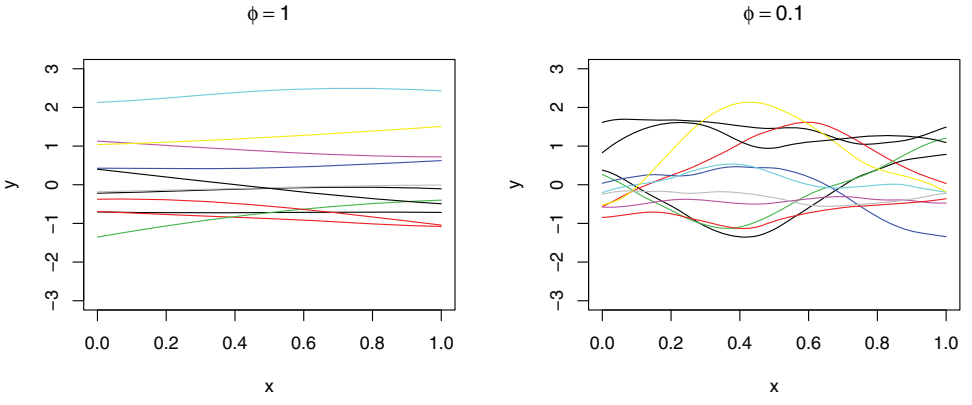

Figure 1. Realizations of $GP(0, \Phi(\cdot))$, where Φ is a Matérn correlation function $\Phi(\boldsymbol{x}_i, \boldsymbol{x}_j; \nu = 5/2, \phi = 1)$ (left) and a Matérn correlation function $\Phi(\boldsymbol{x}_i, \boldsymbol{x}_j; \nu = 5/2, \phi = 0.1)$ (right).

$\Phi(\boldsymbol{x}_i, \boldsymbol{x}_j; \nu = 5/2, \phi = 0.1)$ (right). We can see that the smoothness of the realizations will increase with the increasing of scale parameter ϕ. That is, by choosing appropriate scale parameter ϕ, the realizations of $GP(0, \Phi(\cdot))$ with Matérn correlation function $\Phi(\boldsymbol{x}_i, \boldsymbol{x}_j; \nu = 5/2, \phi)$ are smooth enough to fit the discrepancy in this simulation study.

First we set $n = 6$. The corresponding design points are $\boldsymbol{X} = \{0, 0.2, 0.4, 0.6, 0.8, 1\}^T$. The Maximum likelihood estimate of the scale parameter ϕ is 1.1026, which is larger than 1. That is, the estimation of discrepancy between ζ and y^s is smoother than the realizations which are shown in the left subfigure of Figure 1. It indicates that choosing Matérn correlation function with smoothness parameter $\nu = 5/2$ is suitable. The predictive curves from the parametric and the non-parametric models are plotted in Figure 2. It can be seen that the non-parametric predictive curve is too close to the observed points. In the presence of the observation errors, there is a distance between the observed points and the true process. As a result, the non-parametric predictive curve does not match the true process well. This phenomenon is known as the overfitting problem in the literature [25]. To compare the performance between the non-parametric and the parametric methods, we calculate their Mean Square Prediction Error (MSPE), which is approximated by Monte Carlo as

$$\frac{1}{1000}\sum_{i=1}^{1000}(\zeta(t_i) - \hat{\zeta}_p(t_i))^2, \tag{12}$$

where $\hat{\zeta}_p(\cdot)$ denotes either the non-parametric or the parametric predictor of $\zeta(\cdot)$, and t_i's are random samples from the uniform distribution on $[0, 1]$. The MSPE of the non-parametric method is 0.102, which is greater than 0.090 for the parametric method. This suggests that the simple linear regression method outperforms the non-parametric method, even if the former uses a misspecified model.

Next we consider larger sample sizes. The results for $n = 6 + 3i$ for $i = 0, \ldots, 7$ are shown in Figure 3. It can be seen that the non-parametric method outperforms the parametric method when the sample size is greater than 12. This can be explained by the consistency of the non-parametric method. However, in many applications, the number

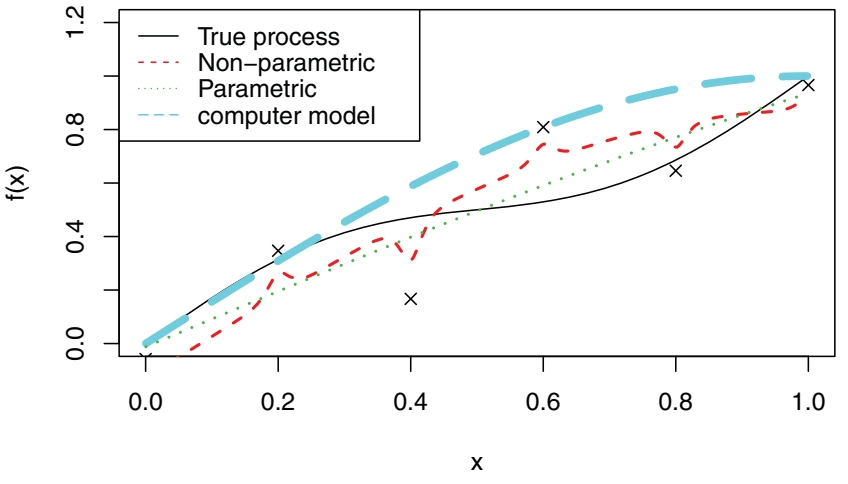

Figure 2. A comparison between the predictive curves given by the parametric and non-parametric models when $n = 6$. The solid line is the true process (8), the long dashed line is the computer model (9), the dashed line is the non-parametric predictive curve using ordinary Kriging model with a Matérn correlation function with smoothness parameter $\nu = 5/2$, and the dotted line is the parametric prediction given by (10).

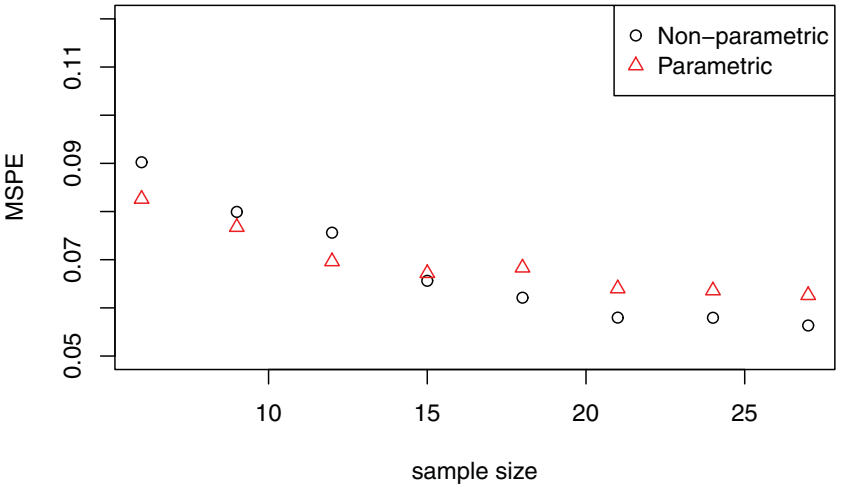

Figure 3. MSPE by the parametric (triangles) and the non-parametric (circles) methods with different sample sizes.

of physical observations can be rather small, and this is why we need computer simulation. In this situation, the parametric methods can perform better than the non-parametric methods, like the results shown in Figure 3 for $n < 12$. This phenomenon becomes even more clear in higher dimensions, because the non-parametric methods suffer more from the curse of dimensionality [26].

As discussed above, both the parametric and the non-parametric methods have certain advantages but also suffer from some deficiencies. A natural question is whether we can find a method which enjoys the advantages of both methods, that is, a method which

is robust as a non-parametric method and has the rate of convergence like a parametric method. In Section 3, we will introduce such a method, called the *semi-parametric adjustment*.

3. Semi-parametric adjustment method

As before, we use model (1) for the physical responses. We do not impose a parametric form for $\zeta(\cdot)$ so that $\zeta(\cdot)$ can be arbitrary. In this section, we suppose the computer code is cheap, i.e. the computer model $y^s(\cdot)$ is known. The proposed method adopts a *semi-parametric* approach to predict the true process, i.e. the predictor depends only on a finite number of unknown parameters. Inspired by KO's method, a natural predictor of this kind is $\rho y^s(\cdot)$, where ρ is the regression coefficient introduced in (4). Here we consider a more flexible class of predictors, given by

$$\boldsymbol{\theta}^T \boldsymbol{F}^s(\cdot) = \theta_0 y^s(\cdot) + \theta_1 f_1(\cdot) + \cdots + \theta_m f_m(\cdot), \tag{13}$$

where $\boldsymbol{F}^s = (y^s, f_1, \ldots, f_m)^T$, $f_1, \ldots, f_m$ are prespecified basis functions, and $\boldsymbol{\theta} = (\theta_0, \ldots, \theta_m)^T \in R^{m+1}$ is a vector of regression coefficients. We remark that the predictor given by (13) differs from that given by the parametric model (3), because in (3) the true process $\zeta(\cdot)$ is modelled as a linear function of the computer model, i.e. ,there exists a "true" regression coefficient vector, but (13) is only an approximation to $\zeta(\cdot)$.

Given the fact that there does not exit a value of $\boldsymbol{\theta}$ such that $\boldsymbol{\theta}^T \boldsymbol{F}^s(\cdot) = \zeta(\cdot)$ in general, we need to define the best 'representative' of $\zeta(\cdot)$ in the set of $\{\boldsymbol{\theta}^T \boldsymbol{F}^s(\cdot) : \boldsymbol{\theta} \in R^{m+1}\}$. Intuitively, the "best" value of $\boldsymbol{\theta}$ should be defined as the minimize of $\| \zeta(\cdot) - \boldsymbol{\theta}^T \boldsymbol{F}^s(\cdot) \|$ for some norm $\| \cdot \|$. We suppose the input domain Ω is a convex and compact subset of R^d. Inspired by Tuo and Wu [20], we choose the L_2 norm and define $\boldsymbol{\theta}^* = (\theta_0^*, \ldots, \theta_m^*)^T$ as

$$\begin{aligned} \boldsymbol{\theta}^* &:= \operatorname*{argmin}_{\boldsymbol{\theta} \in R^{m+1}} \| \zeta(\cdot) - \boldsymbol{\theta}^T \boldsymbol{F}^s(\cdot) \|_{L_2(\Omega)}^2, \\ &= \left(\int_\Omega \boldsymbol{F}^s(\boldsymbol{x}) \{\boldsymbol{F}^s(\boldsymbol{x})\}^T \, \mathrm{d}\boldsymbol{x} \right)^{-1} \int_\Omega \boldsymbol{F}^s(\boldsymbol{x}) \zeta(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}. \end{aligned} \tag{14}$$

In this work, we treat $\boldsymbol{\theta}^*$ as the 'true' value of $\boldsymbol{\theta}$.

One justification for choosing the L_2 norm comes from the common use of the quadratic loss function. Suppose that $\boldsymbol{x}$ follows the uniform distribution on Ω, then $\boldsymbol{\theta}^* \boldsymbol{F}^s(\boldsymbol{x})$ has the minimum mean square predictive error among all predictors with the form $\boldsymbol{\theta} \boldsymbol{F}^s(\boldsymbol{x})$.

In practice, $\zeta(\cdot)$ is unknown, and when the computer code is expensive, $y^s(\cdot)$ is also unknown. Consequently, $\boldsymbol{\theta}^*$ can't be obtained by solving (14) directly. Here we construct the proposed predictor by using the plug-in principle. First we find estimators for $\zeta(\cdot)$ and y^s. Then we estimate θ^* by replacing $\zeta(\cdot)$ and y^s in (14) by their estimates.

We estimate $\zeta(\cdot)$ using the *ridge kernel regression*. Choose a positive definite kernel Φ over $\Omega \times \Omega$. Let $\mathcal{N}_\Phi(\Omega)$ be the *reproducing kernel Hilbert space* (RKHS) generated by the kernel function Φ. See Appendix 1 for details about RKHS. The ridge kernel regression of $\zeta(\cdot)$ is

$$\hat{\zeta}_n(\cdot) := \operatorname*{argmin}_{g \in \mathcal{N}_\Phi(\Omega)} \frac{1}{n} \sum_{i=1}^n (y_i^P(\boldsymbol{x}_i) - g(\boldsymbol{x}_i))^2 + \lambda_n \| g \|_{\mathcal{N}_\Phi(\Omega)}^2, \tag{15}$$

where $\|\cdot\|_{\mathcal{N}_\Phi(\Omega)}$ is the norm of $\mathcal{N}_\Phi(\Omega)$. The smoothing parameter $\lambda_n > 0$ can be selected through a model selection criterion, such as the generalized cross validation (GCV) method [27]. Although $\hat{\zeta}_n$ is defined as the minimizer of an infinite-dimensional optimization problem in (15), it admits a simple closed-form expression. Denote $\boldsymbol{Y}^p = (y_1^p, \ldots, y_n^p)^T$, $\boldsymbol{\Phi} = \{\Phi(x_i, x_j)\}_{1\le i,j\le n}$ and denote the $n \times n$ identity matrix as $\boldsymbol{I}_n$. According to the representer theorem [27,28], for $\boldsymbol{x} \in \Omega$,

$$\hat{\zeta}_n(\boldsymbol{x}) = \sum_{i=1}^{n} u_i \Phi(\boldsymbol{x}, \boldsymbol{x}_i), \tag{16}$$

where $\boldsymbol{u} = (u_1, \ldots, u_n)^T$ is the solution to the linear system

$$\boldsymbol{Y}^p = (\boldsymbol{\Phi} + n\lambda_n \boldsymbol{I}_n)\boldsymbol{u}. \tag{17}$$

When the computer code is expensive, we need a surrogate model to approximate $y^s(\cdot)$. The main idea is to run the computer code over a selected set of points and apply certain function reconstruction method to build a surrogate model. Surrogate modelling is a major topic in computer experiments and uncertainty quantification. One can construct $\hat{y}^s(\cdot)$ by any of the standard methods such as Kriging models and their variations (e.g. [22,29,30]) or generalized polynomial chaos [11,31].

Now we define the L_2 estimator of $\boldsymbol{\theta}$ by

$$\hat{\boldsymbol{\theta}}_n^{L_2} := \underset{\boldsymbol{\theta}\in R^{m+1}}{\operatorname{argmin}} \| \hat{\zeta}_n(\cdot) - \boldsymbol{\theta}^T \hat{\boldsymbol{F}}^s(\cdot) \|_{L_2(\Omega)}, \tag{18}$$

where $\hat{\boldsymbol{F}}^s = (\hat{y}^s, f_1, \ldots, f_m)^T$ and the semi-parametric prediction is

$$\hat{\zeta}_n^{L_2}(\cdot) = \{\hat{\boldsymbol{\theta}}_n^{L_2}\}^T \hat{\boldsymbol{F}}^s. \tag{19}$$

Direct calculations show that $\hat{\boldsymbol{\theta}}_n^{L_2}$ can be expressed as:

$$\begin{aligned} \hat{\boldsymbol{\theta}}_n^{L_2} &= \underset{\boldsymbol{\theta}\in R^{m+1}}{\operatorname{argmin}} \int_\Omega (\hat{\zeta}_n(\boldsymbol{x}) - \boldsymbol{\theta}^T \hat{\boldsymbol{F}}^s(\boldsymbol{x}))^2 \, \mathrm{d}\boldsymbol{x}, \\ &= \left(\int_\Omega \hat{\boldsymbol{F}}^s(\boldsymbol{x}) \{\hat{\boldsymbol{F}}^s(\boldsymbol{x})\}^T \, \mathrm{d}\boldsymbol{x} \right)^{-1} \int_\Omega \hat{\boldsymbol{F}}^s(\boldsymbol{x}) \hat{\zeta}_n(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}. \end{aligned} \tag{20}$$

When we choose $m = 1, \boldsymbol{F}^s = (\hat{y}^s, 1)$ and $\boldsymbol{\theta} = (\theta_0, \theta_1)^T$, (20) becomes

$$\begin{aligned} \hat{\theta}_{0n}^{L_2} &= \frac{\int_\Omega \hat{\zeta}_n(\boldsymbol{x}) \hat{y}^s(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} - \int_\Omega \hat{\zeta}_n(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \int_\Omega \hat{y}^s(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}}{\int_\Omega (\hat{y}^s(\boldsymbol{x}))^2 \, \mathrm{d}\boldsymbol{x} - (\int_\Omega \hat{y}^s(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x})^2}, \\ \hat{\theta}_{1n}^{L_2} &= \int_\Omega \hat{\zeta}_n(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} - \hat{\theta}_{0n} \int_\Omega \hat{y}^s(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}. \end{aligned}$$

We summarize of main steps of the proposed methodology as follows.

Step 1. Build a surrogate model for the computer code using the computer experiment $(\boldsymbol{X}^s, \boldsymbol{Y}^s)$. Skip this step if the computer code is cheap.

Step 2. Compute the ridge kernel regression estimator for $\zeta(\cdot)$ using (16).
Step 3. Choose the basis functions $f_i, i = 1, \ldots, m$.
Step 4. Calculate $\hat{\boldsymbol{\theta}}_n^{L_2}$ using (20). [Step 5.] Predict the true process using (19).

4. Asymptotic results

In this section, the asymptotic properties of the proposed method are studied. To establish suitable results, we need to require the surrogate model $\hat{y}^s$ depending also on n. This assumption agrees with a general principle in computer experiments: when we increase our total budget to run additional experiment trials, we should run both the physical and the computer experiments.

We will show the asymptotic normality of $\hat{\boldsymbol{\theta}}_n^{L_2}$ and $\hat{\zeta}_n^{L_2}$ in Theorems 4.1 and 4.2. First, we introduce technical conditions (C1)–(C6).

(C1) $y^s, f_1, \ldots, f_m \in \mathcal{N}_\Phi(\Omega)$;
(C2) $\mathcal{N}_\Phi(\Omega, \rho) = \{f : \| f \|_{\mathcal{N}_\Phi(\Omega)} \leq \rho, \rho \geq 0\}$ is a Donsker class;
(C3) $\| \hat{\zeta}_n(\cdot) \|_{\mathcal{N}_\Phi(\Omega)} = O_p(1)$;
(C4) $\| \hat{\zeta}_n(\cdot) - \zeta(\cdot) \|_{L_2(\Omega)} = o_p(1)$;
(C5) $\lambda_n = o_p(n^{-1/2})$;
(C6) $\| \hat{y}^s(\cdot) - y^s(\cdot) \|_{L_\infty(\Omega)} = o_p(n^{-1/2})$.

Condition (C1) can be met easily if Φ is chosen to be a Matérn kernel. It is known that the reproducing kernel Hilbert space generated by a Matérn kernel with smoothness ν is equivalent to the Sobolev space $H^{\nu+d/2}(\Omega)$. See Wendland [32] and Tuo and Wu [21] for details. In this case, all smooth functions, such as polynomials, lie in $\mathcal{N}_\Phi(\Omega)$. However, the reproducing kernel Hilbert spaces generated by Gaussian kernels are rather small. Even nonzero constants are not contained in these spaces [33]. Therefore, we recommend using Matérn kernels to ensure the theoretical guarantees. See Section 6 for more discussions.

If a Matérn kernel with smoothness ν is used, Conditions (C2)–(C4) are fulfilled provided that $\lambda_n^{-1} = O(n^{2\nu+d/(2\nu+2d)})$. See Van der Vaart [34], Tuo and Wu [21,24]. Condition (C6) is a requirement of the convergence of the surrogate model. Given the fact that a computer code run is much cheaper than a corresponding physical trial, a typical computer experiment should have much more computer runs than the physical runs. Thus it is reasonable to assume that the approximation error of the surrogate model decays at a fast rate as in (C6).

The proofs of Theorems 4.1–4.2 are given in Appendix 2.

Theorem 4.1: *Suppose the physical design points* $\{\boldsymbol{x}_i\}_{i=1}^n$ *are independent and identically distribution and following a uniform distribution over* Ω. *Under conditions* (C1)–(C6), *we have*

$$\hat{\boldsymbol{\theta}}_n^{L_2} - \boldsymbol{\theta}^* = -2V^{-1} \left\{ \frac{1}{n} \sum_{i=1}^n e_i \boldsymbol{F}^s(\boldsymbol{x}_i) \right\} + o_p(n^{-1/2}). \tag{21}$$

Clearly, $\hat{\boldsymbol{\theta}}_n^{L_2} \xrightarrow{P} \boldsymbol{\theta}^*$ *and*

$$\sqrt{n}(\hat{\boldsymbol{\theta}}_n^{L_2} - \boldsymbol{\theta}^*) \xrightarrow{d} N(0, 4\sigma^2 V^{-1}), \tag{22}$$

provided that $V := \mathrm{E}[F^s(\boldsymbol{x}_1)F^s(\boldsymbol{x}_1)^T]$ *is invertible.*

Theorem 4.2: *Under the same conditions as Theorem* 4.1*, for any fixed* $\boldsymbol{x}_0 \in \Omega$*, we have*

$$\hat{\zeta}_n^{L_2}(\boldsymbol{x}_0) - \zeta^*(\boldsymbol{x}_0) = -2\left\{\frac{1}{n}\sum_{i=1}^{n} e_i \boldsymbol{F}^s(\boldsymbol{x}_i)\right\}^T V^{-1}\boldsymbol{F}^s(\boldsymbol{x}_0) + o_p(n^{-1/2}), \tag{23}$$

where $\zeta^*(\boldsymbol{x}_0) = \boldsymbol{\theta}^{*T}\boldsymbol{F}^s(\boldsymbol{x}_0)$, $\hat{\zeta}_n^{L_2}(\boldsymbol{x}_0) \xrightarrow{P} \zeta^*(\boldsymbol{x}_0)$ *and*

$$\sqrt{n}(\hat{\zeta}_n^{L_2}(\boldsymbol{x}_0) - \zeta^*(\boldsymbol{x}_0)) \xrightarrow{d} N\left(0, 4\sigma^2\{\boldsymbol{F}^s(\boldsymbol{x}_0)\}^T V^{-1}\{\boldsymbol{F}^s(\boldsymbol{x}_0)\}\right). \tag{24}$$

5. Numerical studies

In this section, we compare the numerical behaviours of different predictors: the proposed semi-parametric predictor, the KO's predictor, the engineering-driven adjusted predictor and the predictor based on the OLS estimator [21,35]. The predictor based on the OLSE is defined as

$$\hat{\zeta}^{\mathrm{OLS}}(\cdot) = \left\{\hat{\boldsymbol{\theta}}_n^{\mathrm{OLS}}\right\}^T \hat{\boldsymbol{F}}^s(\cdot), \tag{25}$$

where $\hat{\boldsymbol{\theta}}_n^{\mathrm{OLS}}$ is the OLS estimator to $\boldsymbol{\theta}^*$, which is defined as

$$\begin{aligned}\hat{\boldsymbol{\theta}}_n^{\mathrm{OLS}} &= \underset{\boldsymbol{\theta}\in R^{m+1}}{\mathrm{argmin}} \frac{1}{n}\sum_{i=1}^{n}(y_i^p - \boldsymbol{\theta}^T\hat{\boldsymbol{F}}^s(\boldsymbol{x}_i))^2, \\ &= \left\{\hat{\boldsymbol{F}}^s(\boldsymbol{X})\{\hat{\boldsymbol{F}}^s(\boldsymbol{X})\}^T\right\}^{-1}\hat{\boldsymbol{F}}^s(\boldsymbol{X})\boldsymbol{Y}^p,\end{aligned} \tag{26}$$

where $\hat{\boldsymbol{F}}^s(\boldsymbol{X}) = (\hat{\boldsymbol{F}}^s(\boldsymbol{x}_1), \hat{\boldsymbol{F}}^s(\boldsymbol{x}_2), \ldots, \hat{\boldsymbol{F}}^s(\boldsymbol{x}_n))^T$.

5.1. Example 1: revisit the example in section 2

In Section 2, we compare the parametric and non-parametric methods using the true process (8). Now we revisit this example and compare the semi-parametric method with parametric and non-parametric methods. In the semi-parametric model, we estimate $\hat{\boldsymbol{\beta}}$ by

$$\hat{\boldsymbol{\beta}}_n^{L_2} = \underset{\boldsymbol{\beta}}{\mathrm{argmin}} \int_0^1 \left(\hat{\zeta}_n(x) - \beta_0 y^s(x) - \beta_1 - \beta_2 x\right)^2 \mathrm{d}x, \tag{27}$$

where $\hat{\zeta}_n$ is obtained by the non-parametric method (15) with Matérn kernel, and the smoothing parameter is selected through generalized cross validation (GCV, [27]). Then

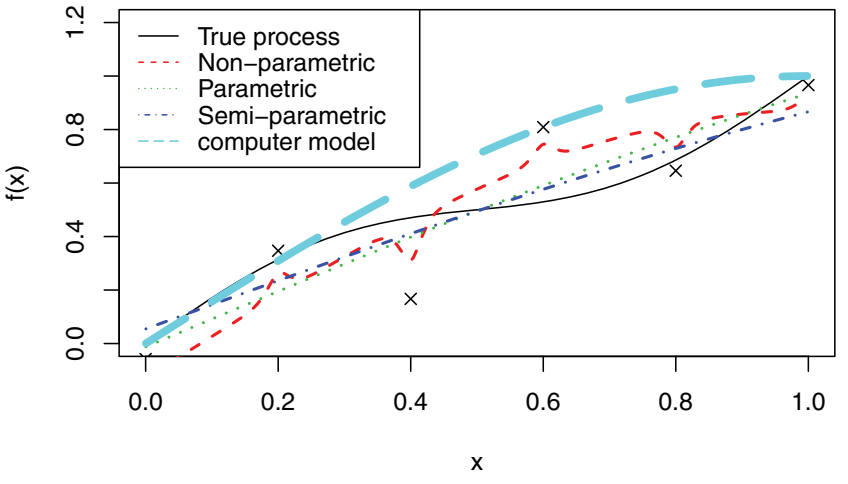

Figure 4. A comparison between the predictive curves given by the parametric, non-parametric and semi-parametric models when $n = 6$. The solid line is the true process (8), the long dashed line is the computer model (9), the dashed line is the non-parametric predictive curve using ordinary Kriging model with a Matérn correlation function with smoothness parameter $\nu = 5/2$, the dotted line is the parametric prediction given by (10), and the dotdash line is the semi-parametric prediction given by (28).

the semi-parametric predictor is

$$\hat{\zeta}_n^{L_2} = \hat{\beta}_{n0}^{L_2} y^s(x) + \hat{\beta}_{n1}^{L_2} + \hat{\beta}_{n2}^{L_2} x, \tag{28}$$

Same as Figure 2, we compare the predictive curves from the parametric, the non-parametric and the semi-parametric models in Figure 4. It can be seen that, the semi-parametric model is closer to the true process, that is, the semi-parametric method gives a better prediction to the true process.

Figure 5 compares the prediction performance of the parametric, the non-parametric and the semi-parametric methods with the different sample sizes. When the sample size is smaller than 18, the semi-parametric method outperforms the other two methods.

To example the prediction error in (19) when the sample size is small, Figure 6 shows the box plots of MSPE given by the parametric, the non-parametric and the semi-parametric methods with the sample size smaller than 18.

Figure 6 shows that the variance of the MSPE given by the semi-parametric method is smaller than other methods. It indicates that the performance of the proposed method is better than the parametric and the non-parametric method when the sample size is small.

5.2. Example 2: one-dimensional function

Suppose the true process is

$$\zeta(x) = 5 * \exp(-1.4x) \cos\left(\frac{7\pi x}{2}\right), x \in [0, 1], \tag{29}$$

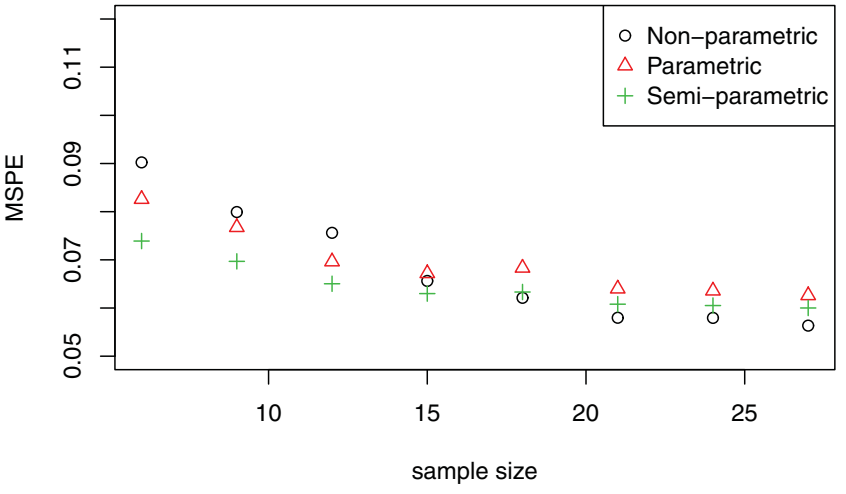

Figure 5. MSPE given by the parametric (triangles), the non-parametric (circles) and the semi-parametric (crosses) methods with different sample size.

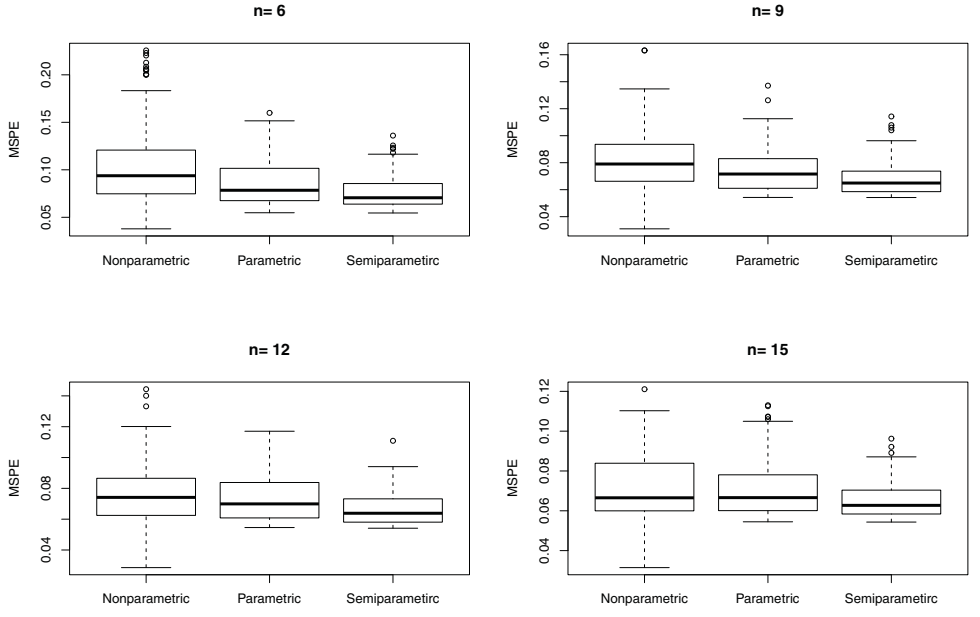

Figure 6. Box plots of MSPE given by the parametric, the non-parametric and the semi-parametric methods with different sample size.

and the physical observations are given by

$$y_i^p = \zeta(x_i) + e_i, i = 1, \ldots, n, \tag{30}$$

with

$$x_i = \frac{i-1}{n-1}, e_i \sim N(0, \sigma^2), \sigma^2 = 0.001.$$

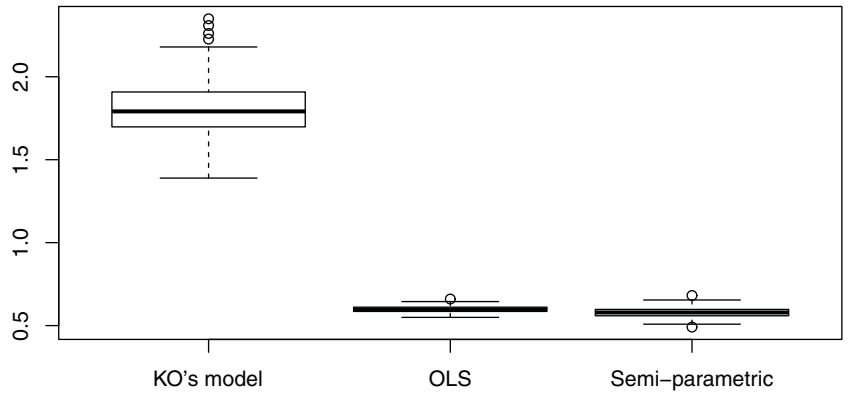

Figure 7. MSPE estimated by three methods, Example 2.

Suppose the computer model is

$$y^s = 100 * \cos\left(\frac{7\pi x}{2}\right), x \in [0,1]. \tag{31}$$

Let $n = 10$, $N = 15$, $\boldsymbol{F}^s(\cdot) = (y^s(\cdot), 1)^T$. Denote the physical observations as $\boldsymbol{Y}^p = (y_1^p, \ldots, y_{10}^p)^T$ and the computer outputs as $\boldsymbol{Y}^s = (y_1^s, \ldots, y_{15}^s)^T$.

The surrogate model $\hat{y}^s$ is built by using a Kriging model with a Matérn correlation function with $\nu = 5/2$. The non-parametric estimate $\hat{\zeta}_n(\cdot)$ is constructed using a Matérn correlation family, where the parameter ϕ in the non-parametric regression is selected by Smoothly Clipped Absolute Deviation (SCAD) method [36].

For a comprehensive understanding of the performance of each method, we repeat this simulation for 500 times. In each simulation run, we generate random e_i's. To examine the performance of each predictor, we calculate the mean square prediction error (MSPE), which is approximated by Monte Carlo as

$$\frac{1}{1000}\sum_{i=1}^{1000}(\zeta(t_i) - \hat{\zeta}_p(t_i))^2, \tag{32}$$

where $\hat{\zeta}_p(\cdot)$ denotes either the KO's, the OLS or the semi-parametric predictor of $\zeta(\cdot)$, and t_i's are random samples from the uniform distribution on $[0,1]$. The result is shown in Figure 7.Figure 7 shows that the proposed predictor gives the best prediction with $\text{MSPE}^{L_2} = 0.5797$, and the mean square error of the OLS predictor is $\text{MSPE}^{\text{OLS}} = 0.5978$. There is no much difference between the two estimators. While the mean square error of KO's predictor is $\text{MSPE}^{KO's} = 1.8044$. If we use the computer output to predict the physical process directly, the prediction is poor with $\text{MSPE}^{Com} = 4734.122$. We do not show this result in Figure 7.

According to Theorem 3 of Tuo and Wu [21], $\hat{\boldsymbol{\theta}}_n^{\text{OLS}}$ is also a consistent estimation of $\boldsymbol{\theta}^*$ with a greater asymptotic variance than $\hat{\boldsymbol{\theta}}_n^{L_2}$. We compare the numerical performance of $\hat{\boldsymbol{\theta}}_n^{\text{OLS}}$ and $\hat{\boldsymbol{\theta}}_n^{L_2}$ in Figure 8:Each entry of $\boldsymbol{\theta}^* = (0.0274, 0.1962)$ is shown the values of the horizontal lines in the Figure 8. We can see that the proposed method estimated by L_2

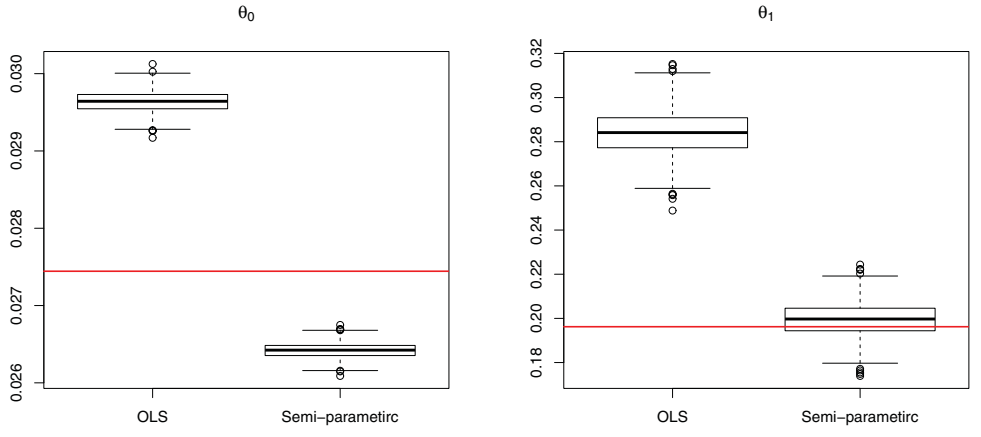

Figure 8. The left plot shows $\hat{\theta}_{0n}^{\text{OLS}}$, $\hat{\theta}_{0n}^{L_2}$ and θ_{0n}^{*} by the horizontal line. The right plot shows $\hat{\theta}_{1n}^{\text{OLS}}$, $\hat{\theta}_{1n}^{L_2}$ and θ_{1n}^{*} by the horizontal line.

estimator gives a much better estimate to $\boldsymbol{\theta}^*$ with more accurate point estimation and less variance. This agrees with the theoretical result that the L_2 estimator is semi-parametric efficient while the OLS is not. See Tuo and Wu [21] for details.

5.3. Example 3: Branin function

Now we choose a frequently used test function, Branin function [37], as the true process:

$$\zeta(x_1, x_2) = (x_2 - 5x_1^2/4\pi^2 + 5\pi x_1 - 6)^2 + 10(1 - 1/8\pi)\cos x_1 + 10,$$
$$0 \le x_1, x_2 \le 1. \tag{33}$$

Suppose the computer model is

$$y^s(x_1, x_2) = (x_2 - x_1^2/6 + 15x_1 - 6)^2. \tag{34}$$

Let $n = N = 100$, and drawing 100 samples from the physical process and the computer model using a full factorial design where $x_1, x_2 \in \{0, \frac{1}{9}, \frac{2}{9}, \ldots, 1\}$. Based on (1), we consider 6 levels of $\sigma^2 := \{0.01, 0.05, 0.1, 0.5, 1, 5\}$ to investigate the stability of different methods. For OLS and the proposed method, we use $\boldsymbol{F}^s(x_1, x_2) = (y^s(x_1, x_2), \cos(x_1), 1)^T$ and $\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2)^T$.

Following the same steps in example 1, we compare the MSPE of the proposed method, the predictor based on OLS estimator and KO's method based on 100 simulations. Denote $U_k[0, 1] = \{T_1, \ldots, T_k\}$, where $\{T_1, \ldots, T_k\}$ are generated uniformly and independently in $[0, 1]$. In this example, we generate 100 testing points using full factorial design factorial design where $t_1, t_2 \in U_{10}([0, 1])$.

From Table 1, we can see that the proposed method outperforms OLS and the KO's methods. We also observe that KO's method is less robust to the increase of the variance of the measurement error σ^2, which is known as a common deficiency of non-parametric methods. If we use the computer outputs to predict the physical process directly, the prediction is rather poor, with $MSPE^{Com} = 485.5705$, when $\sigma^2 = 0.01$. Thus we do not include these results in Table 1.

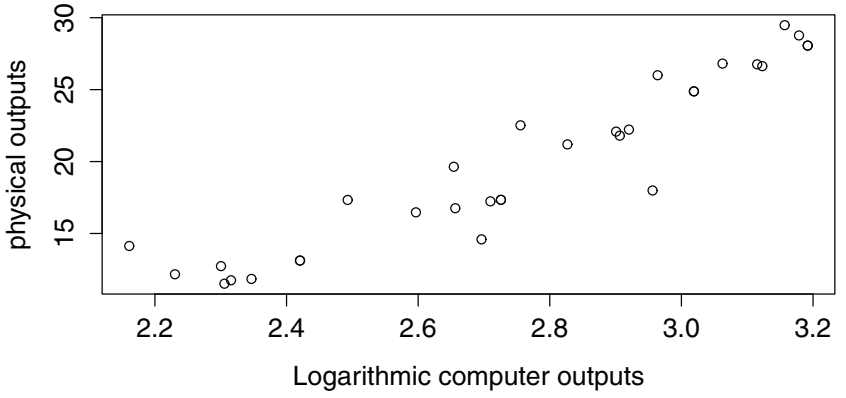

Figure 9. Logarithmic computer outputs versus physical outputs in LAMM process.

Table 1. Numerical results, Example 3.

σ^2	0.01	0.05	0.1	0.5	1	5
Semi-parametric	0.0265	0.0268	0.0271	0.0290	0.0273	0.0333
OLS	0.0268	0.0270	0.0273	0.0288	0.0301	0.0365
KO's	0.0368	0.0655	0.0690	0.0833	0.0933	0.2617

5.4. Example 4: LAMM process

Joseph and Yan [19] discuss the calibration of Laser-Assisted Mechanical Micro-machining (LAMM) process. They analyse the *cutting forces* y with four input variables: *nominal depth of cut* $x_1 \in [10, 25]$, *cutting speed* $x_2 \in [10, 100]$, *laser power* $x_3 \in [0, 35]$ *and laser location* $x_4 \in \{100, 200\}$. And a full factorial $4 \times 2 \times 3 \times 2$ design, in which $x_1 \in \{10, 15, 20, 25\}, x_2 \in \{10, 50\}, x_3 \in \{0, 5, 10\}, x_4 \in \{100, 200\}$, is performed to obtain computer outputs. A physical experiment is carried out on the LAMM process using the same full factorial design with three replicates per run. The data is in Singh et al. [38]. Joseph and Yan [19] propose a engineering-driven adjustment model which makes an adjustment to the computer model:

$$g(\boldsymbol{x}, \boldsymbol{\gamma}) = y^s(\gamma_1 x_1, \ldots, \gamma_d x_d). \tag{35}$$

The adjusted model $g(\boldsymbol{x}, \boldsymbol{\gamma})$ is used to predict the true process, where γ is estimated using a Bayesian approach. We compare the behaviour of the proposed predictor, the OLS predictor, the Engineering-driven method by Joseph and Yan [19], the Bayesian predictor by KO and the computer model. We select 32 out of the 48 runs which also forms an orthogonal design, and make these 32 runs the training data and others testing data. Scatter plot (Figure 9) of logarithmic computer outputs versus physical outputs shows obvious linear relationship between $\ln y^s$ and y^p.

Singh et al. [38] proposed the following nonlinear regression model to approximate the computer model:

$$y^s(\boldsymbol{x}) = \beta_0 x_1^{\beta_1} \exp\{\beta_2 x_2 - \beta_3 x_3 e^{-\beta_4 x_4}\}. \tag{36}$$

Estimations of parameters in (36) are $\hat{\beta}_0 = 1.3272, \hat{\beta}_1 = 0.8962, \hat{\beta}_2 = 0.0016, \hat{\beta}_3 = 0.0293, \hat{\beta}_4 = 0.0039$, respectively, with an $R^2 = 0.9973$ and residual standard error

Table 2. Numerical comparison for LAMM process with modified observations.

ς	0	0.7	1
Semi-parametric	0.3461	0.9558	1.148619
OLS	0.5098	1.0671	1.565882
Engineering-driven	0.4416	1.3983	1.455439
KO's	1.2377	1.8938	2.292385
Computer model	11.3705	12.8469	14.16615

Note: τ is the standard deviation of the modified observations in (37).

$\hat{\sigma} = 0.2752$. Based on the the ANOVA decomposition in Joseph and Yan [19], the main contributors to the discrepancy are the nominal depth of cut x_1 and the laser power x_3, so we suppose that $\boldsymbol{F}^s(\boldsymbol{x}) = (\ln y^s(\boldsymbol{x}), x_1, x_3, x_1 \times x_3, 1)$ and $\boldsymbol{\theta} = (\theta_0, \ldots, \theta_4)$.

In order to compare the prediction accuracy as well as the robustness of different methods, we make some modifications to the LAMM process. Denote y_r^p the physical observations in Singh et al. [38], and y_m^p the modified observations. To simplify the process, let

$$y_m^p(\cdot) = y_r^p(\cdot) + \Delta, \tag{37}$$

where $\Delta \sim N(0, \varsigma^2)$, with standard deviation ς. Joseph and Yan [19] gives the estimation of ς that $\hat{\varsigma} = 0.7$. We can compare the prediction accuracy of the proposed predictor, the predictor based on OLSE, Engineering-driven statistical adjustment predictor in Joseph and Yan [19], Bayesian predictor in Kennedy and O'Hagan [13] and the computer model under the value of $\varsigma := \{0, 0.7, 1\}$. The MSPE of different methods are tabulated in Table 2.

It shows that without modification ($\varsigma = 0$), the OLS and the proposed predictor improve the prediction accuracy significantly than KO's method, the proposed predictor seems to fit the data more excellent than the predictor based on the OLS estimator, and when ς gets larger, the engineering-driven method gives the better prediction than the predictor based on the OLS estimator. The proposed method can obtain the robustest predictions when the physical observation error is large.

6. Concluding remarks

We proposed a new statistical adjustment method for computer models, called the semi-parametric adjustment. The proposed method is proven to enjoy nice theoretical properties and its performance is confirmed by three numerical studies and one real data analysis.

As we said in Section 4, Condition (C1) is rather strong if a Gaussian kernel is used, because many commonly used regression functions are not contained in the reproducing kernel Hilbert spaces generated by Gaussian kernels. However, our numerical experience implies that the estimation behaviour of θ^* is not inferior in the case that a Gaussian kernel is used. This suggests that Condition (C1) may be relaxed. Such a result requires a separate investigation.

Some computer models contain calibration parameters which need to be estimated from the data. We can use the L_2 estimation method [20] to estimate the calibration parameters and the statistical adjustment parameters discussed in this work simultaneously. To do this, we can treat the statistical adjustment parameters as calibration parameters as well and then invoke the L_2 calibration method. In this case, the estimator does not have a closed form

unless the computer output depends linearly on the computer model parameters. Note that identifying calibration parameters is already able improve the prediction performance of the computer model, introducing additional adjustment terms may not lead to a substantial improvement in the prediction accuracy. Also, estimating too many parameters may result in a curse-of-dimensionality issue. Hence, we do not recommend applying the proposed method when several computer models parameters are available for calibration.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by Division of Mathematical Sciences [1914636].

References

[1] Wu CFJ, Hamada MS. Experiments: planning, analysis, and optimization. Vol. 552. Hoboken (NJ): John Wiley & Sons; 2011.
[2] Craig PS, Goldstein M, Rougier JC, et al. Bayesian forecasting for complex systems using computer simulators. J Amer Statist Assoc. 2001;96(454):717–729.
[3] Boukouvalas A, Gosling JP, Maruri-Aguilar H. An efficient screening method for computer experiments. Technometrics. 2014;56(4):422–431.
[4] Drignei D. Fast statistical surrogates for dynamical 3D computer models of brain tumors. J Comput Graph Stat. 2008;17(4):844–859.
[5] Wang S, Chen W, Tsui K-L. Bayesian validation of computer models. Technometrics. 2009;51(4):439–451.
[6] Plumlee M, Joseph VR, Yang H. Calibrating functional parameters in the ion channel models of cardiac cells. J Amer Statist Assoc. 2015;111:500–509.
[7] Roache PJ. Verification and validation in computational science and engineering. Albuquerque (NM): Hermosa; 1998.
[8] Trucano TG, Swiler LP, Igusa T, et al. Calibration, validation, and sensitivity analysis: what's what. Reliab Engn Syst Safety. 2006;91(10):1331–1357.
[9] Xiong Y, Chen W, Tsui K-L, et al. A better understanding of model updating strategies in validating engineering models. Comput Methods Appl Mech Eng. 2009;198(15):1327–1337.
[10] Joseph VR, Melkote SN. Statistical adjustments to engineering models. J Qual Technol. 2009;41(4):362.
[11] Marzouk Y, Xiu D. A stochastic collocation approach to bayesian inference in inverse problems. Commun Comput Phys. 2009;6:826–847.
[12] Williams, CK and Rasmussen CE. Gaussian processes for machine learning. Cambridge (MA): MIT press; 2006.
[13] Kennedy MC, O'Hagan A. Bayesian calibration of computer models. J R Stat Soc Ser B (Stat Methodol). 2001;63(3):425–464.
[14] Higdon D, Kennedy M, Cavendish JC, et al. Combining field data and computer simulations for calibration and prediction. SIAM J Sci Comput. 2004;26(2):448–466.
[15] Bayarri M, Berger JO, Paulo R, et al. A framework for validation of computer models. Technometrics. 2007;49:138–154.
[16] Bayarri M, Berger JO, Cafeo J, et al. Computer model validation with functional output. Ann Stat. 2007;35:1874–1906.
[17] Qian PZ, Wu CFJ. Bayesian hierarchical modeling for integrating low-accuracy and high-accuracy experiments. Technometrics. 2008;50(2):192–204.

[18] Chang C-J, Joseph VR. Model calibration through minimal adjustments. Technometrics. 2014;56(4):474–482.
[19] Joseph VR, Yan H. Engineering-driven statistical adjustment and calibration. Technometrics. 2015;57(2):257–267.
[20] Tuo R, Wu CFJ. A theoretical framework for calibration in computer models: parametrization, estimation and convergence properties. SIAM/ASA J Uncertainty Quant. 2016;4(1):767–795.
[21] Tuo R, Wu CFJ. Efficient calibration for imperfect computer models. Ann Stat. 2015;43(6): 2331–2352.
[22] Santner TJ, Williams BJ, Notz WI. The design and analysis of computer experiments. New York (NY): Springer; 2013.
[23] Stein ML. Interpolation of spatial data: some theory for kriging. New York (NY): Springer; 2012.
[24] Tuo R, Wu CJ. Prediction based on the Kennedy-O'hagan calibration model: asymptotic consistency and other properties. Stat Sin. 2018;28:743–759.
[25] Friedman J, Hastie T, Tibshirani R, The elements of statistical learning. Berlin; 2001. (Springer series in statistics Springer. Vol. 1).
[26] Donoho DL. High-dimensional data analysis: the curses and blessings of dimensionality. AMS Math Challenges Lecture. 2000;1:32.
[27] Wahba G. Spline models for Observational data. Vol. 59. Philadelphia (PA): Siam; 1990.
[28] Schölkopf B, Herbrich R, Smola AJ, A generalized representer theorem. In International Conference on Computational Learning Theory. Springer; 2001. p. 416–426.
[29] Gramacy RB, Lee HKH. Bayesian treed gaussian process models with an application to computer modeling. J Amer Statist Assoc. 2008;103(483):1119–1130.
[30] Gramacy RB, Lee HKH. Adaptive design and analysis of supercomputer experiments. Technometrics. 2009;51(2):130–145.
[31] Xiu D. Numerical methods for stochastic computations: A spectral method approach. Princeton (NJ): Princeton University Press; 2010.
[32] Wendland H. Scattered data approximation. Vol. 17. Cambridge: Cambridge University Press; 2004.
[33] Steinwart I, Hush D, Scovel C. An explicit description of the reproducing kernel Hilbert spaces of gaussian rbf kernels. IEEE Trans Inf Theory. 2006;52(10):4635–4643.
[34] Van der Vaart AW. Asymptotic statistics. Vol. 3. Cambridge: Cambridge University Press; 2000.
[35] Wong RK, Storlie CB, Lee T. A frequentist approach to computer model calibration. J R Stat Soc Ser B (Stat Methodol). 2017;79:635–648.
[36] Wang H, Li R, Tsai C-L. Tuning parameter selectors for the smoothly clipped absolute deviation method. Biometrika. 2007;94(3):553–568.
[37] Dixon LCW, Szegö GP. Towards global optimisation. Amsterdam: North-Holland; 1978.
[38] Singh RK, Joseph VR, Melkote SN. A statistical approach to the optimization of a laser-assisted micromachining process. Int J Adv Manufact Technol. 2011;53(1–4):221–230.
[39] Van Der Vaart AW, Wellner JA. Weak convergence and empirical processes. New York (NY): Springer; 1996.
[40] Kosorok MR. Introduction to empirical processes and semiparametric inference. New York (NY): Springer; 2007.

Appendices

Appendix 1. Reproducing Kernel Hilbert space

Let $\Omega \subseteq \mathrm{R}^{\mathrm{d}}$ denote the region of interest for the input variables, which is convex and compact. Let $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ be a set of points on Ω. Assume that $\Phi : \Omega \times \Omega \to \mathrm{R}$ is a symmetric positive definite function, define the R-linear space

$$F_{\Phi}(\Omega) = \left\{ \sum_{i=1}^{n} \beta_i \Phi(\cdot, \boldsymbol{x}_i) : \beta_i \in \mathrm{R}, \boldsymbol{x}_i \in \Omega \right\},$$

and equip this space with bilinear form

$$\left\langle \sum_{i=1}^{n} \beta_i \Phi(\cdot, \boldsymbol{x}_i), \sum_{j=1}^{m} \gamma_j \Phi(\cdot, \boldsymbol{y}_j) \right\rangle := \sum_{i=1}^{n} \sum_{j=1}^{m} \beta_i \gamma_j \Phi(\boldsymbol{x}_i, \boldsymbol{y}_j).$$

Then the *reproducing kernel Hilbert space* $\mathcal{N}_\Phi(\Omega)$ generated by the function Φ is defined as the closure of $F_\Phi(\Omega)$ under the inner product $< \cdot, \cdot >_\Phi$, and the norm of $\mathcal{N}_\Phi(\Omega)$ is $\| f \|_{\mathcal{N}_\Phi(\Omega)} = \sqrt{< f, f >_{\mathcal{N}_\Phi(\Omega)}}$, where $< \cdot, \cdot >_{\mathcal{N}_\Phi(\Omega)}$ is induced by $< \cdot, \cdot >_\Phi$. More detail about *reproducing kernel Hilbert space* can be found in Wendland [32] and Wahba [27].

Appendix 2. Technical Proofs

In this section, we will give the proofs for Theorem 4.1–4.2. Donsker class is a crucial concept in understanding central limit theorems for stochastic processes. We refer to Van Der Vaart and Wellner [39], Kosorok [40] and Van der Vaart [34] for the definition and detailed properties of Donsker classes. Here we mention some properties which are helpful in the proof of Theorem 4.1. Let $\mathcal{F}$ and $\mathcal{G}$ be Donsker classes, then (a) For any $\mathcal{F}_1 \subset \mathcal{F}$, $\mathcal{F}_1$ is a Donsker. (b)$\mathcal{F} \bigcup \mathcal{G}$ and $\mathcal{F} + \mathcal{G}$ are Donsker. (c) If $\mathcal{F}$ and $\mathcal{G}$ are both uniformly bounded, $\mathcal{F} \cdot \mathcal{G}$ is Donsker.

A Proof of Theorem 4.1

Proof: Denote $L_n = \frac{1}{n} \sum_{i=1}^{n} (y^P(\boldsymbol{x}_i) - g(\boldsymbol{x}_i))^2 + \lambda_n \| g \|^2_{\mathcal{N}_\Phi(\Omega)}$. Let $\boldsymbol{F}^s_j$ be the jth entry of $\boldsymbol{F}^s$. According to (C1), we have that $\boldsymbol{F}^s_j \in \mathcal{N}_\Phi(\Omega), j = 1, \ldots, m+1$. Because $\hat{\zeta}_n = \text{argmin}_{g \in \mathcal{N}_\Phi(\Omega)} L_n(g) \in \mathcal{N}_\Phi(\Omega)$, we have that

$$\begin{aligned}
0 &= \frac{\partial}{\partial t_j} L_n \left(\hat{\zeta}_n(\boldsymbol{x}) + \boldsymbol{t}^T \boldsymbol{F}^s(\boldsymbol{x}) \right) |_{t=0}, \\
&= \frac{\partial}{\partial t_j} \left\{ \frac{1}{n} \sum_{i=1}^{n} (y^P(\boldsymbol{x}_i) - \hat{\zeta}_n(\boldsymbol{x}_i) - \boldsymbol{t}^T \boldsymbol{F}^s(\boldsymbol{x}_i))^2 + \lambda_n \left\langle \hat{\zeta}_n + t\boldsymbol{F}^s, \hat{\zeta}_n + \boldsymbol{t}^T \boldsymbol{F}^s \right\rangle_{\mathcal{N}_\Phi(\Omega)} \right\} |_{t=0}, \\
&= \frac{2}{n} \sum_{i=1}^{n} \boldsymbol{F}^s_j(\boldsymbol{x}_i) \left\{ \hat{\zeta}_n(\boldsymbol{x}_i) - \zeta(\boldsymbol{x}_i) \right\} - \frac{2}{n} \sum_{i=1}^{n} \boldsymbol{F}^s_j(\boldsymbol{x}_i) e_i + 2\lambda_n \left\langle \hat{\zeta}_n, \boldsymbol{F}^s_j \right\rangle_{\mathcal{N}_\Phi(\Omega)}, \\
&= 2(I_j + II_j + III_j). \qquad \text{(A1)}
\end{aligned}$$

First, we consider I_j, let $A_{ij}(g, \theta) = \{g(\boldsymbol{x}_i) - \zeta(\boldsymbol{x}_i)\} \boldsymbol{F}^s_j(\boldsymbol{x}_i)$, where $g \in \mathcal{N}_\Phi(\Omega, \rho)$. Define the empirical process

$$E_{jn}(g, \boldsymbol{\theta}) = n^{-1/2} \sum_{i=1}^{n} \{A_{ij}(g, \boldsymbol{\theta}) - E(A_{ij}(g, \boldsymbol{\theta}))\}. \qquad \text{(A2)}$$

Because that $\mathcal{N}_\Phi(\Omega, \rho)$ is a Donsker for all $\rho > 0$. Thus $\mathcal{F}_1 = \{g - \zeta : g \in \mathcal{N}_\Phi(\Omega, \rho)\}$ is also Donsker. Moreover, $\mathcal{F}_{2j} = \{\boldsymbol{F}^s_j\}, j = 1, \ldots, m+1$ is Donsker as well. For the theory of Donsker classes, we refer to Kosorok [40] and the references therein. Thus the asymptotic equicontinuity property holds, which suggests that for any $\xi > 0$, there exists a $\delta > 0$ such that

$$\lim_{n \to \infty} \sup P \left(\sup_{h \in \mathcal{F}_1 \times \mathcal{F}_2, \|h\|_{L_2(\Omega)} \leq \delta} \frac{1}{\sqrt{n}} \sum_{i=1}^{n} |h(\boldsymbol{x}_i) - E[h(\boldsymbol{x}_i)]| > \xi \right) < \xi. \qquad \text{(A3)}$$

Therefore, the condition $\|\zeta - \hat{\zeta}_n\|_{L_2(\Omega)} = o_p(1)$ implies that

$$\begin{aligned}
o_p(1) = E_{1n}(\hat{\zeta}_n, \hat{\boldsymbol{\theta}}^{L_2}_n) &= n^{-1/2} \sum_{i=1}^{n} \{\hat{\zeta}_n(\boldsymbol{x}_i) - \zeta(\boldsymbol{x}_i)\} \boldsymbol{F}^s_j(\boldsymbol{x}_i, \hat{\boldsymbol{\theta}}^{L_2}_n) \\
&\quad - n^{-1/2} \int_\Omega \{\hat{\zeta}_n(\boldsymbol{z}) - \zeta(\boldsymbol{z})\} \boldsymbol{F}^s_j(\boldsymbol{z}, \hat{\boldsymbol{\theta}}^{L_2}_n) \mathrm{d}z. \qquad \text{(A4)}
\end{aligned}$$

Thus, we have that

$$I_j = \int_\Omega \boldsymbol{F}_j^s(\boldsymbol{x}) \left\{ \hat{\zeta}_n(\boldsymbol{x}) - \zeta(\boldsymbol{x}) \right\} \mathrm{d}\boldsymbol{x} + o_p(n^{-1/2}). \tag{A5}$$

Denote $I = (I_1, \ldots, I_{m+1})^T$. Then we obtain

$$\begin{aligned} I &= \int_\Omega \boldsymbol{F}^s(\boldsymbol{x}) \left\{ \hat{\zeta}_n(\boldsymbol{x}) - \zeta(\boldsymbol{x}) \right\} \mathrm{d}\boldsymbol{x} + o_p(n^{-1/2}), \\ &= \frac{1}{2} \int_\Omega \left\{ \frac{\partial^2}{\partial \boldsymbol{\theta}^T \partial \boldsymbol{\theta}_j} \left(\boldsymbol{\theta}^T \boldsymbol{F}^s(\boldsymbol{z}) - \zeta(\boldsymbol{z}) \right)^2 \right\} (\hat{\boldsymbol{\theta}}_n^{L_2} - \boldsymbol{\theta}^*), \\ &= \frac{1}{2} V(\hat{\boldsymbol{\theta}}_n^{L_2} - \boldsymbol{\theta}^*) + o_p(n^{-1/2}), \end{aligned} \tag{A6}$$

where $V := \mathrm{E}[\boldsymbol{F}^s(\boldsymbol{x}) \boldsymbol{F}^s(\boldsymbol{x})^T]$ is invertible.

For each j, we apply Cauchy-Schwarz inequality to find that

$$|III_j| \leq \lambda_n \parallel \hat{\zeta}_n \parallel_{\mathcal{N}_\Phi(\Omega)} \parallel \boldsymbol{F}_j^s \parallel_{\mathcal{N}_\Phi(\Omega)} = o_p(n^{-1/2}). \tag{A7}$$

So $III = (III_1, \ldots, III_{m+1})^T = o_p(n^{-1/2})$.

Note that $II = (II_1, \ldots, II_{m+1})^T = -\frac{1}{n} \sum_{i=1}^n \boldsymbol{F}^s(\boldsymbol{x}_i) e_i$. By combining (A1)–(A7), we arrive at the desired result. ■

B Proof of Theorem 4.2

Proof: Since (19) holds, we have that $\hat{\zeta}_n^{L_2}(\boldsymbol{x}) - \zeta^*(\boldsymbol{x}) = (\hat{\theta}_n^{L_2} - \theta^*)^T \boldsymbol{F}^s(\boldsymbol{x})$, and thus we obtain the desired result from Theorem 4.1. ■