This is a preprint of the following article, which is available from http://mdolab.engin.umich.edu
Mohamed A. Bouhlel and J. R. R. A. Martins. Gradient-enhanced kriging for high-dimensional problems. Engineering with Computer, 2018. (In press).
The published article may differ from this preprint, and is available by following the DOI above.

Gradient-enhanced kriging for high-dimensional problems

Mohamed A. Bouhlel, J. R. R. A. Martins

Abstract Surrogate models provide an affordable alternative to the evaluation of expensive deterministic functions. However, the construction of accurate surrogate models with many independent variables is currently prohibitive because they require a large number of function evaluations for the desired accuracy. Gradient-enhanced kriging has the potential to reduce the number of evaluations when efficient gradient computation, such as an adjoint method, is available. However, current gradient-enhanced kriging methods do not scale well with the number of sampling points because of the rapid growth in the size of the correlation matrix, where new information is added for each sampling point in each direction of the design space. Furthermore, they do not scale well with the number of independent variables because of the increase in the number of hyperparameters that must be estimated. To address this issue, we develop a new gradient-enhanced surrogate model approach that drastically reduces the number of hyperparameters through the use of the partial least squares method to maintain accuracy. In addition, this method is able to control the size of the correlation matrix by adding only relevant points defined by the information provided by the partial least squares method. To validate our method, we compare the global accuracy of the proposed method with conventional kriging surrogate models on two analytic functions with up to 100 dimensions, as well as engineering problems of varied complexity with up to 15 dimensions. We show that the proposed method requires fewer sampling points than conventional methods to obtain the desired accuracy, or it provides more accuracy for a fixed budget of sampling points. In some cases, we get models that are over three times more accurate than previously developed surrogate models for the same computational time, and over 3200 times faster than standard gradient-enhanced kriging models for the same accuracy.

Symbols and notation

Matrices and vectors are in bold type.

Symbol	Meaning
d	Number of dimensions
B	Hypercube expressed by the product between intervals of each direction space
n	Number of sampling points
h	Number of principal components
$\mathbf{x}$, $\mathbf{x}'$	$1 \times d$ vector
x_j	j^{th} element of $\mathbf{x}$ for $j = 1, \ldots, d$
$\mathbf{X}$	$n \times d$ matrix containing sampling points
$\mathbf{y}$	$n \times 1$ vector containing simulation of $\mathbf{X}$
$\mathbf{x}^{(i)}$	i^{th} sampling point for $i = 1, \ldots, n$ ($1 \times d$ vector)
$y^{(i)}$	i^{th} evaluated output point for $i = 1, \ldots, n$
$\mathbf{X}^{(0)}$	$\mathbf{X}$
$\mathbf{X}^{(l-1)}$	Matrix containing residual of the $(l-1)^{\text{th}}$ inner regression
$k(\cdot, \cdot)$	Covariance function
$\mathbf{r}_{\mathbf{x}\mathbf{x}'}$	Spatial correlation between $\mathbf{x}$ and $\mathbf{x}'$
$\mathbf{R}$	Covariance matrix
$s^2(\mathbf{x})$	Prediction of the kriging variance
σ^2	Process variance
θ_i	i^{th} parameter of the covariance function for $i = 1, \ldots, d$
$Y(\mathbf{x})$	Gaussian process
$\mathbf{1}$	n-vector of ones
$\mathbf{t}_l$	l^{th} principal component for $l = 1, \ldots, h$
$\mathbf{w}$	Weight vector for partial least squares
Δx_j	First-order Taylor approximation step in the j^{th} direction

1 Introduction

Surrogate models, also known as metamodels or response surfaces, are approximate functions (or outputs) over a space defined by independent variables (or inputs) based on a limited number of function evaluations (or samples). This modeling approach replaces expensive function evaluations with much cheaper calculations. Surrogate approaches often used in engineering applications include polynomial regression, the support vector machine, radial basis function models, and kriging [14]. More details on the kriging model are given in Section 2.1. Surrogate models are classified based on whether they are noninterpolating, such as polynomial regression, or interpolating, such as kriging. Surrogate models can handle both deterministic and noisy functions, but we consider only deterministic functions.

Surrogate models can be particularly helpful in conjunction with numerical optimization, which requires multiple function evaluations over a design variable space [16, 19, 48]. However, noninterpolating surrogate models are not appropriate for optimization problems because additional points do not necessarily lead to a more accurate surface [19]. On the other hand, interpolating surrogate models become accurate in the specific area where new points are added. One of the most popular interpolating models is the kriging model [12, 26, 38, 44, 47], also known as Gaussian process regression [4, 43, Ch. 3, Sec. 19]. [25] provides a general review of the kriging approach and presents the basic derivation. One of its major advantages is the built-in analytical estimate of the model error, which makes kriging a probabilistic model for which we can use statistical techniques [20].

Several researchers have shown that kriging metamodels can significantly reduce the cost of numerical analysis and optimization. [18], for example, used a kriging metamodel to model a two-dimensional airfoil design including flap position in a multi-element airfoil, where the lift-to-drag ratio was maximized using a genetic algorithm. Since genetic algorithms require a large number of function evaluations, the kriging

surrogate greatly reduced the computational cost. [50] used two kriging-based optimizations with an intermediate step using a proper orthogonal decomposition method to minimize the drag-to-lift ratio of a transonic airfoil. This approach outperformed a traditional kriging-based optimization, producing better designs and a considerable reduction of the optimization cost. This was achieved by restricting the budget of the first optimization and keeping only the relevant points for the second optimization. [5] used a mixture of experts, which is a technique that partitions the input space into different regions, so that a different metamodel can be used for each region. They used several types of kriging to model and minimize the weight of an aircraft wing structure. The structural optimization considered 12 thickness variables (spars, skins, and ribs) and two stress constraints (spars and skins). Their results showed that optimization based on the kriging models required fewer evaluations than a direct optimization method. Many other applications using the kriging model can be found in the literature [10, 22, 23, 29, 30, 31, 46].

A kriging model can be extended to utilize gradient information when available, which improves its accuracy. This is known as gradient-enhanced kriging (GEK) [31], cokriging [11, 27], or first-order kriging [28]. GEK has been shown to be effective in various studies [11, 27, 28, 33], and it is especially advantageous when the gradient is computed with an adjoint method. The adjoint method computes derivatives for models containing implicitly defined equations. The main advantage is that the computational cost, unlike that for finite-difference approximations, is independent of the number of input variables [42]. The adjoint method can be derived by linearizing the residuals of the implicit equation, or alternatively through the application of the inverse function theorem [37]. [27] compared kriging and direct-indirect GEK (using a discrete adjoint method to compute the gradients) and showed a considerable gain in global accuracy when using the indirect GEK on an aerodynamic shape optimization problem. Despite this performance, the number of input variables was kept low (2 to 6) because of the exorbitant computational cost required to build GEK for larger inputs. [31] used a mixture-of-experts method based on GEK to approximate the drag coefficients in a surrogate-based aircraft mission analysis. They showed that GEK models were superior to conventional surrogate models, especially in terms of accuracy. The number of input variables was again low (2 and 4).

GEK is subject to performance degradation when the number of input variables and/or the number of sampling points is high. This degradation is mainly due to the size of the GEK correlation matrix, which increases proportionally with both the number of inputs and the number of sampling points. In addition, sampling points that are close to each other lead to quasi-linearly-dependent columns in the correlation matrix. It becomes ill-conditioned, and the corresponding linear problem becomes difficult to solve. There are other challenges in high-dimensional problems because building a kriging surrogate model involves solving a multimodal optimization problem in which the number of variables is proportional to the problem dimension. We must maximize a function—the *likelihood function*—with respect to variables called *hyperparameters*.

Because it is difficult to find the hyperparameters through optimization, [27] developed a method that guesses an initial solution for the GEK hyperparameters and then uses a gradient-based optimization method to maximize the likelihood function. This method accelerates the construction of the GEK model; however, the initial guess depends on a fixed parameter that defines the strength of the correlation between the two most directional-distant sample points. This parameter depends on the physical function being studied and thus requires trial and error. Therefore, it is not easy to generalize this approach. [28] tried to accelerate the estimation of the GEK hyperparameters by reducing their number to one for all directions. The GEK model gave better results than conventional kriging (using one hyperparameter for all directions) on a borehole flow-rate problem with eight input variables. However, Lewis assumed that the problem is isotropic, which is not the case for the engineering problems we want to tackle.

Bouhlel et al. [7] proposed an approach, KPLS, that combines the kriging model with the partial least squares (PLS) method to accelerate the kriging construction. This method introduces new kernels based on the information extracted from the PLS technique. The number of hyperparameters is then reduced to the number of principal components retained. Experience shows that two or three principal components are usually sufficient to get good accuracy [7]. There is currently no rule of thumb for the maximum number of principal components to retain because it depends on both the problem and the location of the sampling points used to fit the model. The KPLS model was shown to be efficient for several high-dimensional

problems. [7] compared KPLS and conventional kriging models on analytic and engineering problems with up to 100 inputs. Despite the reduced number of hyperparameters in KPLS, they obtained similar results in terms of accuracy for both models. The main benefit of KPLS was the reduction in the computational time to construct the model: it was up to 450 times faster than conventional kriging.

Another variant of KPLS, called KPLSK [6], extends the KPLS method by adding a new step into the construction of the model. Once the KPLS method is built, the solution for the hyperparameters is used as a first guess for a gradient-based optimization applied to a conventional kriging model. The KPLSK method is similar to that developed by Ollar et al. [40], where a gradient-free optimization algorithm is used with an isotropic kriging model, and this is followed by a gradient-based optimization starting from the solution to the first optimization. Compared to KPLS and conventional kriging, KPLSK gives a significant improvement in terms of accuracy on analytic and engineering problems with up to 60 dimensions. In addition, KPLSK is more efficient than kriging (up to 131 times faster using 300 points for a $60D$ analytic function) but slightly less efficient than KPLS (22 s vs. 0.86 s for the same test case). For optimization applications using KPLS and KPLSK see Bartoli et al. [5] and Bouhlel et al. [8].

To further improve the efficiency of KPLS and extend GEK to high-dimensional problems, we integrate the gradient during the construction of KPLS and use the PLS method in a different way. Our approach is based on the first-order Taylor approximation (FOTA) at each sampling point. Using this approximation, we generate a set of points around each sampling point and apply the PLS method to each of these sets. We then combine the information from each set of points to build a kriging model. We call this new model GE-KPLS since it uses both the gradient information and the PLS method. The GE-KPLS method utilizes gradient information and controls the size of the correlation matrix by adding some of the approximating points to that matrix with respect to the relevant directions given by the PLS method at each sampling point. The number of hyperparameters to be estimated remains equal to the number of principal components.

The remainder of the paper is organized as follows. First, we review the key equations for the kriging and KPLS models in Sections 2.1 and 2.2, respectively. Then, we summarize the two GEK approaches that have already appeared in the literature in Sections 3.1 and 3.2, followed by the development of the GE-KPLS approach in Section 3.3. We then compare the proposed GE-KPLS approach to previously developed methods in Section 4. We present the limitations of our approach in Section 5 and summarize our conclusions in Section 6.

2 Kriging surrogate modeling

In this section we introduce the notation and briefly describe the theory behind kriging and KPLS. The first step in the construction of surrogate models is the selection of the sample points $\mathbf{x}^{(i)} \in \mathbb{R}^d$, for $i = 1, \ldots, n$, where d is the number of inputs and n is the number of sampling points. We can denote this set of sample points as a matrix,

$$\mathbf{X} = \left[{\mathbf{x}^{(1)}}^T, \ldots, {\mathbf{x}^{(n)}}^T \right]^T. \tag{1}$$

Then, the function to be modeled is evaluated at each sample point. We write it as $f : B \longrightarrow \mathbb{R}$, where, for simplicity, B is a hypercube expressed by the product of the intervals of each direction space. We obtain the outputs $\mathbf{y} = \left[y^{(1)}, \ldots, y^{(n)} \right]^T$ by evaluating the function

$$y^{(i)} = f\left(\mathbf{x}^{(i)} \right), \quad \text{for } i = 1, \ldots, n. \tag{2}$$

With the choice and evaluation of sample points we have $(\mathbf{X}, \mathbf{y})$, which we can now use to construct the surrogate model.

2.1 Conventional kriging

[38] developed the theoretical basis of the kriging approach based on the work of [26]. The approach has since been extended to the fields of computer simulation [44, 45] and machine learning [53]. The kriging model is essentially an interpolation method. The interpolated values are modeled by a Gaussian process

with mean $\mu(\mathbf{x})$ governed by a prior spatial covariance function $k(\mathbf{x}, \mathbf{x}')$. The covariance function k can be written as

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 r(\mathbf{x}, \mathbf{x}') = \sigma^2 r_{\mathbf{x}\mathbf{x}'}, \quad \forall \mathbf{x}, \mathbf{x}' \in B, \tag{3}$$

where σ^2 is the process variance and $r_{\mathbf{x}\mathbf{x}'}$ is the spatial correlation function between $\mathbf{x}$ and $\mathbf{x}'$. The correlation function r depends on the hyperparameters θ, which must be estimated. In this paper, we use the Gaussian exponential correlation function for all the numerical results presented in Section 4:

$$r_{\mathbf{x}\mathbf{x}'} = \prod_{i=1}^{d} \exp\left(-\theta_i \left(x_i - {x'}_i\right)^2\right), \quad \forall \theta_i \in \mathbb{R}^+, \forall \mathbf{x}, \mathbf{x}' \in B. \tag{4}$$

This function relates the correlation between two points $\mathbf{x}$ and $\mathbf{x}'$ to the distance between them. It quantifies the degree of resemblance between any two points in the design space.

Let us now define the stochastic process $Y(\mathbf{x}) = \mu + Z(\mathbf{x})$, where μ is an unknown constant, and $Z(\mathbf{x})$ is a realization of a stochastic Gaussian process with $Z \sim \mathcal{N}(0, \sigma^2)$. In this study, we use the ordinary kriging model, where $\mu(\mathbf{x}) = \mu = \text{constant}$. To construct the kriging model, we must estimate a set of unknown parameters: θ, μ, and σ^2. We use the maximum likelihood estimation method, and we use the natural logarithm to simplify the likelihood maximization:

$$-\frac{1}{2}\left[n \ln(2\pi\sigma^2) + \ln\left(\det \mathbf{R}\right) + (\mathbf{y} - \mathbf{1}\mu)^{\mathbf{t}} \mathbf{R}^{-\mathbf{1}} (\mathbf{y} - \mathbf{1}\mu)/\sigma^{\mathbf{2}}\right], \tag{5}$$

where $\mathbf{1}$ denotes an n-vector of ones.

First, we assume that the hyperparameters θ are known, so μ and σ^2 are given by

$$\hat{\mu} = \left(\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}\right)^{-1} \mathbf{1}^T \mathbf{R}^{-1} \mathbf{y}, \tag{6}$$

where $\mathbf{R} = [\mathbf{r}_{\mathbf{x}^{(1)}\mathbf{X}}, \ldots, \mathbf{r}_{\mathbf{x}^{(n)}\mathbf{X}}]$ is the correlation matrix with $\mathbf{r}_{\mathbf{x}\mathbf{X}} = [r_{\mathbf{x}\mathbf{x}^{(1)}}, \ldots, r_{\mathbf{x}\mathbf{x}^{(n)}}]^T$, and

$$\hat{\sigma}^2 = \frac{1}{n} \left(\mathbf{y} - \mathbf{1}\hat{\mu}\right)^T \mathbf{R}^{-1} \left(\mathbf{y} - \mathbf{1}\hat{\mu}\right). \tag{7}$$

Equations (6) and (7) are derived by setting the derivatives of the likelihood function to zero. Next, we insert both equations into (5) and remove the constant terms. The so-called concentrated likelihood function that depends only on θ is then given by

$$-\frac{1}{2}\left[n \ln(\sigma^2(\theta)) + \ln\left(\det \mathbf{R}(\theta)\right)\right], \tag{8}$$

where $\mathbf{R}(\theta)$ and $\sigma(\theta)$ denote the dependency on θ, and $\sigma(\theta)$ depends on θ through $\mathbf{R}^{-1}$ in Equation (7). A detailed derivation of these equations is provided by [14] and [24]. Finally, the best linear unbiased predictor, given the output $\mathbf{y}$, is

$$\hat{y}(\mathbf{x}) = \hat{\mu} + \mathbf{r}_{\mathbf{x}\mathbf{X}}^T \mathbf{R}^{-1} \left(\mathbf{y} - \mathbf{1}\hat{\mu}\right), \quad \forall \mathbf{x} \in B. \tag{9}$$

Since there is no analytical solution for the hyperparameters θ, it is necessary to use numerical optimization to find the θ that maximizes the likelihood function. This step is the most challenging in the construction of the kriging model. This is because, as previously mentioned, it involves maximizing the likelihood function, which is often multimodal [36]. Maximizing this function becomes prohibitive for high-dimensional problems ($d > 10$) because of the cost of computing the determinant of the correlation matrix and the high number of evaluations needed to optimize a high-dimensional multimodal problem. This is the main motivation for the development of the KPLS approach, which we describe next.

2.2 KPLS(K): Accelerating kriging construction with partial least squares regression

As mentioned in Section 2.1, the estimation of the kriging hyperparameters can be time-consuming, particularly for high-dimensional problems. Bouhlel et al. [7] developed an approach that reduces the computational cost while maintaining accuracy. They use PLS regression during the hyperparameter estimation process. PLS regression is a well-known method for handling high-dimensional problems that maximizes the variance between the input and output variables in a smaller subspace, formed by the principal components, or latent variables. PLS finds a linear regression model by projecting the predicted variables and the observable variables onto a new space. The elements of the principal direction, which is a vector defining the direction of the associated principal component, represent the influence of each input on the output. On the other hand, the hyperparameters θ represent the range in any direction of the space. If, for instance, certain values are less significant in the i^{th} direction, the corresponding θ_i will have a small value. Thus, the key idea behind the construction of the KPLS model is the use of PLS information to adjust the hyperparameters of the kriging model.

We compute the first principal component $\mathbf{t}_1$ by seeking the direction $\mathbf{w}^{(1)}$ that maximizes the squared covariance between $\mathbf{t}_1 = \mathbf{X}\mathbf{w}^{(1)}$ and $\mathbf{y}$, i.e.,

$$\mathbf{w}^{(1)} = \begin{cases} \arg\max\limits_{\mathbf{w}} \ \mathbf{w}^T\mathbf{X}^T\mathbf{y}\mathbf{y}^T\mathbf{X}\mathbf{w} \\ \text{such that } \ \mathbf{w}^T\mathbf{w} = 1. \end{cases} \tag{10}$$

Next, we compute the residual matrix from $\mathbf{X}^{(0)} \leftarrow \mathbf{X}$ space and from $\mathbf{y}^{(0)} \leftarrow \mathbf{y}$ using

$$\begin{aligned} \mathbf{X}^{(1)} &= \mathbf{X}^{(0)} - \mathbf{t}_1\mathbf{p}^{(1)}, \\ \mathbf{y}^{(1)} &= \mathbf{y}^{(0)} - c_1\mathbf{t}_1, \end{aligned} \tag{11}$$

where $\mathbf{p}^{(1)}$ (a $1 \times d$ vector) contains the regression coefficients of the local regression of $\mathbf{X}$ onto the first principal component $\mathbf{t}_1$ (an $n \times 1$ vector), and c_1 is the regression coefficient of the local regression of $\mathbf{y}$ onto the first principal component $\mathbf{t}_1$. Next, the second principal component—orthogonal to the first principal component—can be sequentially computed by replacing $\mathbf{X}^{(0)}$ by $\mathbf{X}^{(1)}$ and $\mathbf{y}^{(0)}$ by $\mathbf{y}^{(1)}$ to solve the maximization problem (10). The same approach is used to iteratively compute the other principal components.

The computed principal components represent the new coordinate system obtained by rotating the original system with the axes $x_1, \ldots, x_d$ [2]. The l^{th} principal component $\mathbf{t}_l$ is

$$\mathbf{t}_l = \mathbf{X}^{(l-1)}\mathbf{w}^{(l)} = \mathbf{X}\mathbf{w}_*^{(l)}, \quad \text{for } l = 1, \ldots, h. \tag{12}$$

The matrix $\mathbf{W}_* = \left[\mathbf{w}_*^{(1)}, \ldots, .\mathbf{w}_*^{(h)}\right]$ is obtained via the following formula [49, p. 114]:

$$\mathbf{W}_* = \mathbf{W}\left(\mathbf{P}^T\mathbf{W}\right)^{-1}, \tag{13}$$

where $\mathbf{W} = \left[\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(h)}\right]$ and $\mathbf{P} = \left[\mathbf{p}^{(1)^T}, \ldots, \mathbf{p}^{(h)^T}\right]$. If $h = d$, the matrix $\mathbf{W}_* = \left[\mathbf{w}_*^{(1)}, \ldots, \mathbf{w}_*^{(d)}\right]$ rotates the coordinate space $(x_1, \ldots, x_d)$ to the new coordinate space $(t_1, \ldots, t_d)$, which follows the principal directions $\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(d)}$. For more details on the PLS method see Helland [17], Frank and Friedman [15], and Alberto and González [2].

The PLS method gives information on the contribution of any variable to the output. [7] use this information to add weights to the hyperparameters θ. For $l = 1, \ldots, h$, the scalars $w_{*1}^{(l)}, \ldots, w_{*d}^{(l)}$ are interpreted as measuring the importance of $x_1, \ldots, x_d$ in the construction of the l^{th} principal component where its correlation with the output y is maximized.

To construct the KPLS kernel, we first define the linear map F_l via

$$\begin{aligned} F_l : B &\longrightarrow B \\ \mathbf{x} &\longmapsto \left[w_{*1}^{(l)}x_1, \ldots, w_{*d}^{(l)}x_d\right], \end{aligned} \tag{14}$$

for $l = 1, \ldots, h$. By using the mathematical property that the tensor product of several kernels is a kernel, we build the KPLS kernel

$$k_{1:h}(\mathbf{x}, \mathbf{x}') = \prod_{l=1}^{h} k_l(F_l(\mathbf{x}), F_l(\mathbf{x}')), \forall \mathbf{x}, \mathbf{x}' \in B, \tag{15}$$

where $k_l : B \times B \to \mathbb{R}$ is an isotropic stationary kernel, which is invariant when translated. More details of this construction are given by [7].

If we use the Gaussian correlation function (4) and Equation (15), we obtain

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 \prod_{l=1}^{h} \prod_{i=1}^{d} \exp\left[-\theta_l \left(w_{*i}^{(l)} x_i - w_{*i}^{(l)} {x'}_i\right)^2\right], \forall\ \theta_l \in [0, +\infty), \quad \forall \mathbf{x}, \mathbf{x}' \in B. \tag{16}$$

The KPLS method reduces the number of hyperparameters to be estimated from d to h, where $h << d$, thus drastically decreasing the time to construct the model.

[6] proposed another method to construct a KPLS-based model for high-dimensional problems, the so-called KPLSK. This method is applicable only when the covariance functions used by KPLS are of the exponential type (e.g., all Gaussian). The covariance function used by KPLSK is then exponential with the same form as the KPLS covariance. This method is basically a two-step approach for optimizing the hyperparameters. The first step optimizes the hyperparameters of a KPLS covariance using a gradient-free method on h hyperparameters for a global optimization in the reduced space. The second step optimizes the hyperparameters of a conventional kriging model using a gradient-based method and the solution of the first step. This gives a local improvement in the original space (d hyperparameters) of the solution provided by the first step. The idea is to use an initial guess and a gradient-based method, which is more efficient than a gradient-free method, for the construction of a conventional kriging model.

The solution of the first step with h hyperparameters is expressed in the larger space with d hyperparameters using a change of variables. Using Equation (16) and the change of variable $\eta_i = \sum_{l=1}^{h} \theta_l {w_{*i}^{(l)}}^2$, we get

$$\begin{aligned} \sigma^2 \prod_{l=1}^{h} \prod_{i=1}^{d} \exp\left(-\theta_l {w_{*i}^{(l)}}^2 (x_i - {x'}_i)^2\right) &= \sigma^2 \exp\left(\sum_{i=1}^{d} \sum_{l=1}^{h} -\theta_l {w_{*i}^{(l)}}^2 (x_i - {x'}_i)^2\right) \\ &= \sigma^2 \exp\left(\sum_{i=1}^{d} -\eta_i (x_i - {x'}_i)^2\right) \\ &= \sigma^2 \prod_{i=1}^{d} \exp\left(-\eta_i (x_i - {x'}_i)^2\right). \end{aligned} \tag{17}$$

This is the definition of a Gaussian kernel given by Equation (4). Therefore, each component of the starting point for the gradient-based optimization uses a linear combination of the hyperparameter values from the reduced space. This allows the use of an initial line search along a hypercube of the original space to find a relevant starting point. Furthermore, the final value of the likelihood function is better than that provided by KPLS. The KPLSK model is computationally more efficient than a kriging model and only slightly more costly than KPLS.

3 Gradient-enhanced kriging

If the gradient of the output function at the sampling points is available, we can use it to increase the accuracy of the surrogate model [51, 52]. Since a gradient consists of d derivatives, adding this information to the function value at each sampling point has the potential to enrich the model immensely. Furthermore, when the gradient is computed using an adjoint method, for which the computational cost is similar to that of a single function evaluation and independent of d, this enrichment can be obtained at a cost that is much lower than the cost of evaluating d new function values.

Various approaches have been developed for GEK, and two main formulations exist: indirect and direct. In the following, we start with a brief review of these formulations, and then we present our approach, GE-KPLS.

3.1 Indirect gradient-enhanced kriging

The indirect GEK method uses the gradient information to generate new points around the sampling points via linear extrapolation. In each direction for each sampling point, we add one point by computing the FOTA

$$y\left(\mathbf{x}^{(i)}+\Delta x_j \mathbf{e}^{(j)}\right)=y\left(\mathbf{x}^{(i)}\right)+\frac{\partial y\left(\mathbf{x}^{(i)}\right)}{\partial x_j}\Delta x_j, \tag{18}$$

where $i=1,\ldots,n$, $j=1,\ldots,d$, Δx_j is the step added in the j^{th} direction, and $\mathbf{e}^{(j)}$ is the j^{th} row of the $d\times d$ identity matrix. The indirect GEK method does not require a modification of the kriging code. However, the resulting correlation matrix can rapidly become ill-conditioned, since the columns of the matrix resulting from the FOTA are almost collinear. Moreover, this method increases the size of the correlation matrix from $n\times n$ to $n(d+1)\times n(d+1)$. Thus, the computational cost to build the model becomes prohibitive for high-dimensional problems.

3.2 Direct gradient-enhanced kriging

In the direct GEK method, the derivative values are included in the vector $\mathbf{y}$ from Equation (9). This vector is now

$$\mathbf{y}=\left[y\left(\mathbf{x}^{(1)}\right),\ldots,y\left(\mathbf{x}^{(n)}\right),\frac{\partial y\left(\mathbf{x}^{(1)}\right)}{\partial x_1},\ldots,\frac{\partial y\left(\mathbf{x}^{(1)}\right)}{\partial x_d},\ldots,\frac{\partial y\left(\mathbf{x}^{(n)}\right)}{\partial x_d}\right]^T, \tag{19}$$

with a size of $n(d+1)\times 1$. The vector of ones from Equation (9) has the same size and is

$$\mathbf{1}=[\overbrace{1,\ldots,1}^{n},\overbrace{0,\ldots,0}^{nd}]^T. \tag{20}$$

The size of the correlation matrix increases to $n(d+1)\times n(d+1)$, and it contains four blocks that include the correlation among the data, among the gradients, between the data and the gradients, and between the gradients and the data. Denoting the GEK correlation matrix by $\dot{\mathbf{R}}$, we can write

$$\dot{\mathbf{R}}=\begin{bmatrix} r_{\mathbf{x}^{(1)}\mathbf{x}^{(1)}} & \cdots & r_{\mathbf{x}^{(1)}\mathbf{x}^{(n)}} & \frac{\partial r_{\mathbf{x}^{(1)}\mathbf{x}^{(1)}}}{\partial \mathbf{x}^{(1)}} & \cdots & \frac{\partial r_{\mathbf{x}^{(1)}\mathbf{x}^{(n)}}}{\partial \mathbf{x}^{(n)}} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ r_{\mathbf{x}^{(n)}\mathbf{x}^{(1)}} & \cdots & r_{\mathbf{x}^{(n)}\mathbf{x}^{(n)}} & \frac{\partial r_{\mathbf{x}^{(n)}\mathbf{x}^{(1)}}}{\partial \mathbf{x}^{(1)}} & \cdots & \frac{\partial r_{\mathbf{x}^{(n)}\mathbf{x}^{(n)}}}{\partial \mathbf{x}^{(n)}} \\ \frac{\partial r_{\mathbf{x}^{(1)}\mathbf{x}^{(1)}}}{\partial \mathbf{x}^{(1)}}^T & \cdots & \frac{\partial r_{\mathbf{x}^{(1)}\mathbf{x}^{(n)}}}{\partial \mathbf{x}^{(1)}}^T & \frac{\partial^2 r_{\mathbf{x}^{(1)}\mathbf{x}^{(1)}}}{\partial^2 \mathbf{x}^{(1)}} & \cdots & \frac{\partial^2 r_{\mathbf{x}^{(1)}\mathbf{x}^{(n)}}}{\partial \mathbf{x}^{(1)}\partial \mathbf{x}^{(n)}} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial r_{\mathbf{x}^{(n)}\mathbf{x}^{(1)}}}{\partial \mathbf{x}^{(n)}}^T & \cdots & \frac{\partial r_{\mathbf{x}^{(n)}\mathbf{x}^{(n)}}}{\partial \mathbf{x}^{(n)}}^T & \frac{\partial^2 r_{\mathbf{x}^{(n)}\mathbf{x}^{(1)}}}{\partial \mathbf{x}^{(n)}\partial \mathbf{x}^{(1)}} & \cdots & \frac{\partial^2 r_{\mathbf{x}^{(n)}\mathbf{x}^{(n)}}}{\partial^2 \mathbf{x}^{(n)}} \end{bmatrix}, \tag{21}$$

where, for $i,j=1,\ldots,n$, $\partial r_{\mathbf{x}^{(i)}\mathbf{x}^{(j)}}/\partial \mathbf{x}^{(i)}$, $\partial r_{\mathbf{x}^{(i)}\mathbf{x}^{(j)}}/\partial \mathbf{x}^{(j)}$, and $\partial^2 r_{\mathbf{x}^{(i)}\mathbf{x}^{(j)}}/\partial \mathbf{x}^{(i)}\partial \mathbf{x}^{(j)}$ are given by

$$\frac{\partial r_{\mathbf{x}^{(i)}\mathbf{x}^{(j)}}}{\partial \mathbf{x}^{(i)}}=\left[\frac{\partial r_{\mathbf{x}^{(i)}\mathbf{x}^{(j)}}}{\partial x_k^{(i)}}=-2\theta_k\left(x_k^{(i)}-x_k^{(j)}\right)r_{\mathbf{x}^{(i)}\mathbf{x}^{(j)}}\right]_{k=1,\ldots,d}, \tag{22}$$

$$\frac{\partial r_{\mathbf{x}^{(i)}\mathbf{x}^{(j)}}}{\partial \mathbf{x}^{(j)}}=\left[\frac{\partial r_{\mathbf{x}^{(i)}\mathbf{x}^{(j)}}}{\partial x_k^{(j)}}=2\theta_k\left(x_k^{(i)}-x_k^{(j)}\right)r_{\mathbf{x}^{(i)}\mathbf{x}^{(j)}}\right]_{k=1,\ldots,d}, \tag{23}$$

$$\frac{\partial^2 r_{\mathbf{x}^{(i)}\mathbf{x}^{(j)}}}{\partial \mathbf{x}^{(i)}\partial \mathbf{x}^{(j)}}=\left[\frac{\partial^2 r_{\mathbf{x}^{(i)}\mathbf{x}^{(j)}}}{\partial x_k^{(i)}\partial x_l^{(j)}}=-4\theta_k\theta_l\left(x_k^{(i)}-x_k^{(j)}\right)\left(x_l^{(i)}-x_l^{(j)}\right)r_{\mathbf{x}^{(i)}\mathbf{x}^{(j)}}\right]_{k,l=1,\ldots,d}. \tag{24}$$

Once the hyperparameters θ are estimated, the GEK predictor for any untried $\mathbf{x}$ is given by

$$\hat{y}(\mathbf{x})=\hat{\mu}+\dot{\mathbf{r}}_{\mathbf{xX}}^T\dot{\mathbf{R}}^{-1}\left(\mathbf{y}-\mathbf{1}\hat{\mu}\right),\quad \forall \mathbf{x}\in B, \tag{25}$$

where the correlation vector contains the correlation between an untried point $\mathbf{x}$ and each training point from $\mathbf{X} = \left[\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(n)}\right]$:

$$\dot{\mathbf{r}}_{\mathbf{xX}} = \left[r_{\mathbf{x}\mathbf{x}^{(1)}} \quad \ldots \quad r_{\mathbf{x}\mathbf{x}^{(n)}} \quad \frac{\partial r_{\mathbf{x}^{(1)}\mathbf{x}}}{\partial \mathbf{x}^{(1)}} \quad \ldots \quad \frac{\partial r_{\mathbf{x}^{(n)}\mathbf{x}}}{\partial \mathbf{x}^{(n)}} \right]^T. \tag{26}$$

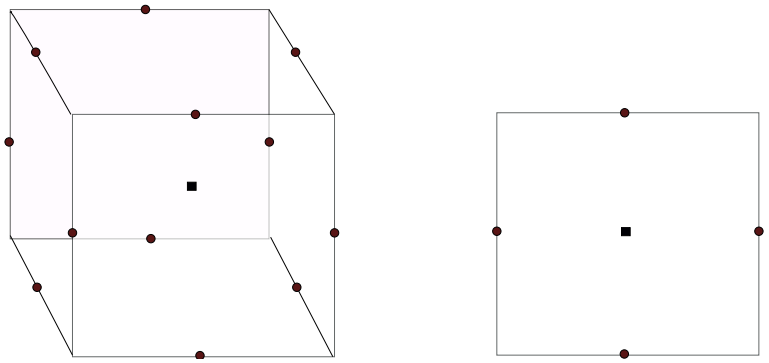
Unfortunately, the correlation matrix $\dot{\mathbf{R}}$ is dense, and its size increases quadratically with both the number of variables d and the number of samples n. In addition, $\dot{\mathbf{R}}$ is not symmetric, which makes it more costly to invert. In the next section, we develop a new approach that uses the gradient information with a controlled increase in the size of $\dot{\mathbf{R}}$.

3.3 GE-KPLS: Gradient-enhanced kriging with partial least squares method

GEK methods have a number of weaknesses. There is a rapid growth in the size of the correlation matrix when the number of sampling points and/or the number of inputs becomes large. Moreover, in high-dimensional problems many hyperparameters must be estimated, and this results in challenges in the maximization of the likelihood function. To address these issues, we propose the GE-KPLS approach, which exploits the gradient information with a slight increase in the size of the correlation matrix but reduces the number of hyperparameters.

3.3.1 Model construction

The key idea of our method is to use the PLS method around each sampling point; we apply it several times, each time to a different point. We use the FOTA (18) to generate a set of points around each sampling point. These new approximate points are constructed by a Box–Behnken design [9, Ch. 11, Sec. 6] when $d \geq 3$ (Figure 1a) and one factor at a time when $d = 2$ (Figure 1b).



(a) Box–Behnken design.

(b) Forward and backward variations of each direction.

Figure 1: The rectangles indicate the sampling points, and the circles indicate the newly generated points.

PLS is applied to GEK as follows. Suppose we have sets of points $\mathcal{S} = \{\mathcal{S}_i, \forall i = 1, \ldots, n\}$, where each set is defined by the sampling point $\left(\mathbf{x}^{(i)}, y^{(i)}\right)$ and the set of approximate points generated as described above. We apply PLS to each $\mathcal{S}_i$ to get the local influence of each direction space. Next, we compute the mean of the n coefficients $\left|\mathbf{w}_*^{(l)}\right|$ for each principal component $l = 1, \ldots, h$. Denoting these new coefficients by $\mathbf{w}_{\text{av}}^{(l)}$, we replace Equation (14) by

$$\begin{aligned} F_l : B &\longrightarrow B \\ \mathbf{x} &\longmapsto \left[w_{\text{av}_1}^{(l)} x_1, \ldots, w_{\text{av}_d}^{(l)} x_d \right]. \end{aligned} \tag{27}$$

Finally, we follow the same construction used for the KPLS model, substituting $\mathbf{w}_*^{(l)}$ by $\mathbf{w}_{\text{av}}^{(l)}$. Thus, Equation (16) becomes

$$k(\mathbf{x},\mathbf{x}') = \sigma^2 \prod_{l=1}^{h}\prod_{i=1}^{d} \exp\left[-\theta_l \left(w_{\text{av}_i}^{(l)} x_i - w_{\text{av}_i}^{(l)} {x'}_i\right)^2\right], \quad \forall\, \theta_l \in [0,+\infty), \quad \forall \mathbf{x},\mathbf{x}' \in B. \tag{28}$$

In the next section, we describe how we control the size of the correlation matrix to obtain the best trade-off between the accuracy of the model and the computational time.

3.3.2 Controlling the size of the correlation matrix

We have seen in Section 3.1 that the construction of the indirect GEK model adds d points around each sampling point. Since the size of the correlation matrix is $n(d+1) \times n(d+1)$, this leads to a dramatic increase in the matrix size. In addition, the added sampling points are close to each other, leading to an ill-conditioned matrix. Thus, its inversion becomes computationally prohibitive for large numbers of sampling points. However, adding only relevant points improves both the condition number of the matrix and the accuracy of the model.

In the previous section, we locally applied PLS with respect to each sampling point, which provides the influence of each input variable around that point. We now add only m approximating points ($m \in [1,d]$) around each sampling point, where m is the highest coefficient of PLS. We consider only the coefficients for the first principal component, which usually contains the most useful information. This improves the accuracy of the model with respect to relevant directions and increases the size of the correlation matrix to only $n(m+1) \times n(m+1)$, where $m << d$.

Algorithm 1 summarizes the construction of the GE-KPLS model from the sampling data to the final predictor. After initializing the training points with the associated derivatives, the number of principal components, and the number of extra points, we compute $\mathbf{w}_{\text{av}}^{(1)}, \ldots, \mathbf{w}_{\text{av}}^{(h)}$. To do this, we construct $\mathcal{S}_i$, apply PLS to $\mathcal{S}_i$, and select for each sample point the m most influential Cartesian directions from the first principal component. Then, we maximize the concentrated likelihood function given by Equation (8), and finally, we find the prediction $\hat{y}$ from Equation (9).

Algorithm 1: Construction of GE-KPLS model

input : $\left(\mathbf{X}, \mathbf{y}, \frac{\partial \mathbf{y}}{\partial \mathbf{X}}, h, m\right)$
output: $\hat{y}(\mathbf{x})$
for $i \leq n$ **do**
  $\mathcal{S}_i$; // Generate a set of approximating points
  $\mathcal{S}_i \xrightarrow{\text{PLS}} \left(\mathbf{w}_*^{(1)}, \ldots, \mathbf{w}_*^{(h)}\right) \max \left|\mathbf{w}_*^{(1)}\right|$; // Select the m most influential coefficients
end
$\mathbf{w}_{\text{av}}^{(1)}, \ldots, \mathbf{w}_{\text{av}}^{(h)}$; // Compute the average of the PLS coefficients
$\theta_1, \ldots, \theta_h$; // Estimate the hyperparameters
$\hat{y}(\mathbf{x})$

In GE-KPLS, PLS is locally applied around each sampling point instead of in the whole space, as in KPLS. This enables us to identify the local influence of the input variables where the sampling points are located. By taking the mean of all the local input variable influences, we expect to obtain a good estimate of the global influences. The main computational advantages are the reduced number of hyperparameters to be estimated, $h << d$, and the reduced size of the correlation matrix, $n(m+1) \times n(m+1)$ with $m << d$, compared to $n(d+1) \times n(d+1)$ for the conventional direct and indirect GEK models.

In the next section, we apply our method to high-dimensional benchmark functions and engineering cases.

4 Numerical experiments

To evaluate the computational cost and accuracy of GE-KPLS, we compare it to other models for a number of benchmark functions. The first set of functions consists of two analytic functions given by

$$y_1(\mathbf{x}) = \sum_{i=1}^{d} x_i^2, \qquad -10 \leq x_i \leq 10, \text{ for } i = 1, \ldots, d; \tag{29}$$

$$y_2(\mathbf{x}) = x_1^3 + \sum_{i=2}^{d} x_i^2, \qquad -10 \leq x_i \leq 10, \text{ for } i = 1, \ldots, d. \tag{30}$$

The second set corresponds to the eight engineering problems listed in Table 2.

Table 2: Definition of engineering functions.

	Problem	d	n_1	n_2	Reference
P_1	Welded beam	2	10	20	Deb [13]
P_2	Welded beam	2	10	20	Deb [13]
P_3	Welded beam	4	20	40	Deb [13]
P_4	Borehole	8	16	80	Morris et al. [39]
P_5	Robot	8	16	80	An and Owen [3]
P_6	Aircraft wing	10	20	100	Forrester et al. [14]
P_7	Vibration	15	75	150	Liping et al. [32]
P_8	Vibration	15	75	150	Liping et al. [32]

Since, as discussed previously, the GEK model does not perform well, especially when the number of sampling points is relatively high, we performed three studies. The first study compares GE-KPLS with the indirect GEK and ordinary kriging models on the analytic functions defined by Equations (29) and (30). The second and third studies, which use the analytic functions and the engineering functions respectively, compare GE-KPLS with ordinary kriging, KPLS, and KPLSK with more sampling points than in the first study.

The kriging and GEK models with Gaussian kernel (4) and KPLS(K) models with Gaussian kernel (16) provide the benchmark results that we compare to the GE-KPLS model with Gaussian kernel (28). We use an unknown constant, μ, as a trend for all the models. For the kriging experiments, we use the scikit-learn implementation [41]. The indirect GEK method does not require a modification of the kriging source code, so we again use scikit-learn.

We vary the number of extra points, m, from 0 to 5. In the first study we also use $m = d$. In the third study we use $m = d$ when the number of inputs is less than five, e.g., for P_1 we use $m = 1$ and $m = d = 2$. We denote GE-KPLS with m extra points by GE-KPLSm. We ran preliminary tests varying the number of principal components from 1 to 3 for the KPLS(K) models, and three components always gave the best results. Using more components becomes costly and results in only a slight difference in terms of accuracy (more or less accurate depending on the problem). For simplicity, we consider only three principal components for KPLS(K). GE-KPLS uses only one principal component, which was found to be optimal.

The GEK and GE-KPLS models use additional information (the gradient components) compared to the other models. To make the comparison as fair as possible, the number of sampling points used to construct these models is always half the number of samples used for the other models. This accounts for the cost of computing the gradient; when an adjoint method is available, this cost is about the same as the cost of computing the function itself [21, 37].

For the generation of approximation points with FOTA, Laurenceau and Sagaut [27] recommend a step of $10^{-4} l_i$, where l_i is the length between the upper and lower bounds in the i^{th} direction. Because the GEK model is very expensive in some cases (see Section 4.1), we used this value for the GE-KPLS and GEK methods for the first study. However, our test cases indicated that it is not always the best step, so we

performed an analysis to determine the best value for the second and third studies. The computational time needed to find the step is not considered, and we report only the time needed to construct the GE-KPLS models using this step.

To compare the accuracy of the various surrogate models, we compute the relative error (RE) for n_v validation points as follows:

$$\mathrm{RE} = \frac{\|\mathbf{y} - \hat{\mathbf{y}}\|}{\|\mathbf{y}\|}, \tag{31}$$

where $\hat{\mathbf{y}}$ contains the surrogate model values evaluated at the validation points, $\mathbf{y}$ contains the corresponding reference function values, and $\|.\|$ is the L_2 norm. Since we use explicit functions, the reference values can be assumed to have a machine epsilon of $\mathcal{O}\left(10^{-16}\right)$. In addition, the function computations are fast, so generating a large set of random validation points is tractable. We use $n_v = 5,000$ validation points for all the cases. The sampling points and validation points are generated using the Latin hypercube implementation in the pyDOE toolbox [1] with the *maximin* and *random* criteria, respectively. We perform 10 trials for each case, and we plot the mean of our results. Finally, all the computations were performed on an Intel® Core™ i7-6700K 4.00 GHz CPU.

4.1 Numerical results for the first study

We first use the analytic functions (29) and (30) and compare GE-KPLSm, for $m = 1, \ldots, 5$ and $m = d$, to the GEK and kriging models. We consider $m = d$ to determine the usefulness of PLS, since the number of extra points is the same as for the GEK model. We vary the number of inputs for both functions from $d = 20$ to $d = 100$ in increments of 20. In addition, we vary the number of sampling points from $n = 10$ to $n = 100$ in increments of 10 for GEK and GE-KPLS. For the kriging model, we use $2n$ sampling points for each case. We do this because the cost of computing the gradient using an adjoint method is typically the same as the cost of evaluating the function, irrespective of the number of inputs [35]. We do not use the adjoint method (we test only analytic functions, which we differentiate analytically), but we use the factor of two to emulate a real-world engineering situation where the gradients are computed using an adjoint method.

Figure 2 summarizes the results of the first study. The first two columns show the RE for y_1 and y_2, and the other two columns show the computational time. The results are presented in increasing order of dimensionality. The models are color coded as indicated in the legend. In some cases, we could not reach 100 sampling points because of the ill-conditioned covariance matrix provided by the GEK model, which explains the missing results in all cases except for the y_1 function with $d = 40$.

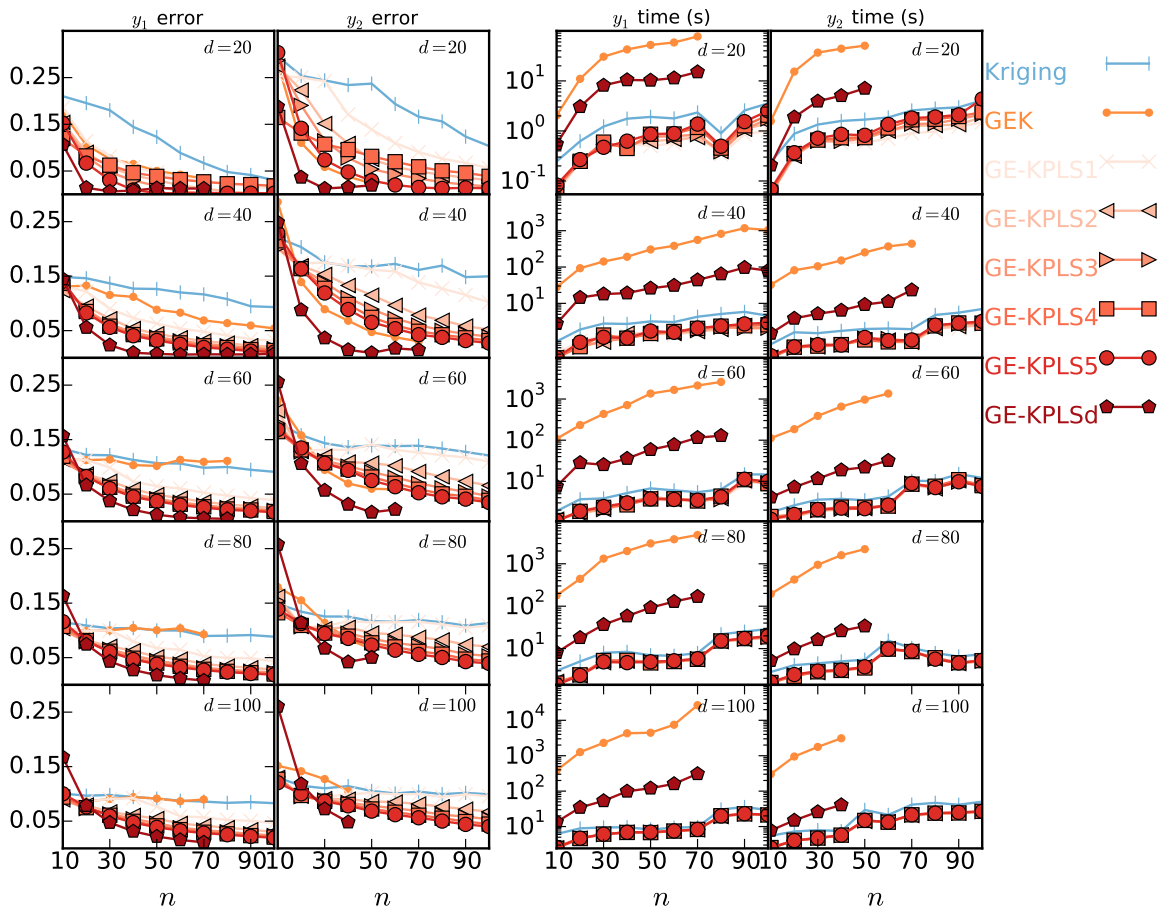


Figure 2: Summary of the mean results for kriging, GEK, and GE-KPLS models applied to the analytic problems, with 10 trials for each case. The models are color coded as shown in the legend.

GE-KPLSd and GEK use the same points (training and approximating points) in their correlation matrices. The difference between the two models is that we reduce the number of hyperparameters using PLS for the first model, so we can verify the scalability of GE-KPLS with the input variables (through the hyperparameters). GE-KPLSd yields a more accurate model than GEK in all cases except for y_1 when $d \geq 40$ and $n = 10$ and for y_2 when $d > 40$ and $n = 10$. These results show the effectiveness of PLS in reducing the computational time, especially when $d > 60$. For example, the time for y_2 with $d = 80$ and $n \leq 50$ is less than 45 s for GE-KPLS whereas GEK requires 42 min. Therefore, PLS improves the accuracy of the model *and* reduces the time needed to construct it.

Even though the RE-convergence of GE-KPLSd is the most accurate, the GE-KPLSm models for $m = 1, \ldots, 5$ are in some cases preferable. For example, the latter are over 37 times faster than the former with about a 1.5% loss in terms of error for y_1 with $d = 100$ and $n = 70$. In addition, including d extra points around each sampling point leads to ill-conditioned matrices, especially when the number of sampling points is high. Furthermore, the GE-KPLSm models for $m = 1, \ldots, 5$ always yield a lower RE and decreased computational time compared to kriging. Compared to kriging, the time for GE-KPLSm is 10 s lower in all cases, and the RE is 10% better in some cases; e.g., y_1 with $d = 20$, $n = 30$ with GE-KPLS5. Compared to GEK, GE-KPLSm has better RE convergence with y_1, and the RE convergence with y_2 is slightly better with GEK when $d \leq 80$. In addition, GE-KPLSm has lower computational times than GEK; e.g., the time needed to construct GE-KPLSm with $m = 1, \ldots, 5$ for y_1 with $100d$ and 70 points is between 7 s and 9 s, compared to about 27000 s for GEK.

We note that y_1 and y_2 differ in only the first term, yet the results for the two functions are different. For example, the RE-convergence values of all the GE-KPLSm models are better than the GEK convergence for y_1 with $d = 40$ but not for y_2 with $d = 40$. Therefore, it is preferable to select the best model for each function separately.

Finally, the construction of the GEK model can be prohibitive in terms of computational time. For instance, we need about 7.4 hours to construct a GEK model for y_1 with $d = 100$ and $n = 70$. Thus, this model is not appropriate when the number of sampling points is high.

In the next section, we increase the number of sampling points and compare the GE-KPLSm models for $m = 1, \ldots, 5$ with kriging, KPLS, and KPLSK.

4.2 Numerical results for the second study

For the second study, we again use the functions y_1 and y_2. For kriging, KPLS, and KPLSK, we set the number of sampling points to $n = kd$, where $k = 2, 10$. Typically $k = 10$ is used [34], but we add $k = 2$ because in many engineering applications function evaluations are expensive, and $k = 10$ is unrealistic. We use $n/2$ sampling points to construct GEKPLSm in order to keep the cost of the sampling points constant for all the test models.

In Figure 3 we plot the computational time versus the RE to analyze the trade-off between them. Each line represents a given surrogate model and has two points corresponding to $n_1 = 2d$ and $n_2 = 10d$ (kriging, KPLS, and KPLSK) or to $\frac{n_1}{2}$ and $\frac{n_2}{2}$ (GE-KPLSm). The models are color coded as indicated in the legend. The upper plots correspond to y_1, and the lower plots correspond to y_2, with $d = 10$ on the left and $d = 100$ on the right. More detailed numerical results for the mean of the RE and the computational time are given in Table 3 in Appendix B.

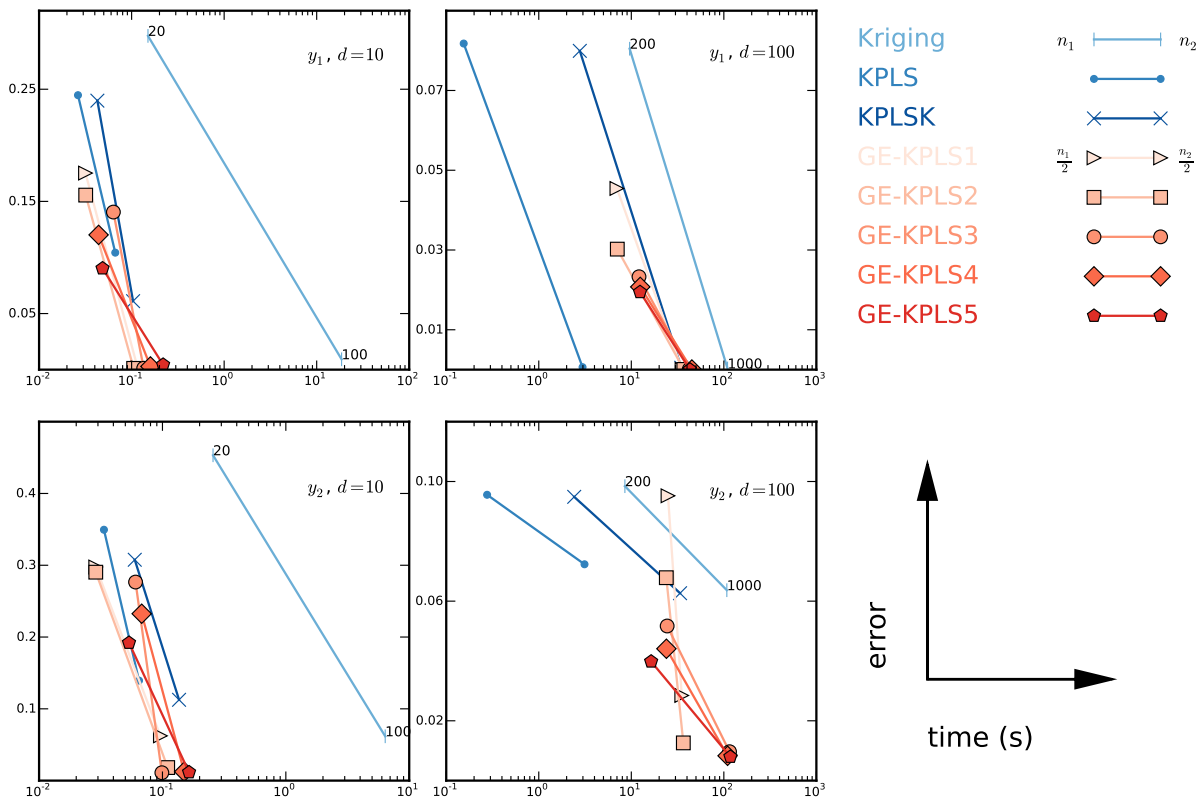


Figure 3: Summary of results for all models applied to the analytic problems, with 10 trials for each case. The models are color coded as shown in the legend. The best trade-off between time and error is always obtained by a GE-KPLS model.

Figure 3 shows that adding m extra points to the correlation matrix improves the accuracy of the results, and the best trade-off between time and error is always obtained by a GE-KPLS model. At the expense of a slight increase in time, increasing the number of extra points yields a lower error in almost all cases. GE-KPLS5 gives a lower error for all cases except for y_1 and y_2 with 10 dimensions and 50 sampling points, where the lowest error is obtained with GE-KPLS2 and GE-KPLS3, respectively. Thus, the number of extra points must be carefully selected.

We can evaluate the performance by either comparing the time required to achieve a certain level of accuracy or comparing the accuracy for a given time. GE-KPLS1, for instance, provides an excellent compromise between error and time: it is able to achieve an RE below 1% in less than 0.1 s for y_1 with 10 dimensions and 50 points. Even better, GE-KPLS5 with 100 sampling points gives a lower RE than the kriging model with 1000 sampling points (1.94% vs. 2.96%) for y_1 with $d = 100$. In this case, the time required to build GE-KPLS3 is lower by a factor of 9 than the time needed by kriging (12.4 s vs. 109.6 s). In addition, GE-KPLS is able to avoid ill-conditioned correlation matrices by reducing the number of extra points through PLS. Thus, this second study confirms the efficiency of the GE-KPLSm methods and their ability to generate accurate models.

4.3 Numerical results for the third study

We now assess the performance of GE-KPLS on the eight engineering functions listed in Table 2. P_1, P_2, and P_3 are the deflection, bending stress, and shear stress of a welded beam problem [13]. P_4 is the water flow rate through a borehole that is drilled from the ground surface through two aquifers [39]. P_5 gives the position of a robot arm [3]. P_6 estimates the weight of a light aircraft wing [14]. P_7 and P_8 are, respectively, the weight and lowest natural frequency of a torsion vibration problem [32]. The number of dimensions for the eight problems varies from 2 to 15; detailed formulations are provided in Appendix A. We have chosen to cover a range of engineering areas using different numbers of dimensions and complexities to verify the applicability of our approach. To build the kriging, KPLS, and KPLSK models, we use $n_1 = 2d$ and $n_2 = 10d$ for all the problems except for P_1, P_2, and P_3 where $n_1 = 5d$ (see Table 2 for more details). We use $n_1/2$ and $n_2/2$ sampling points for the GE-KPLS models. We use GE-KPLS to construct surrogate models for these functions and compare our results to those obtained by kriging, KPLS, and KPLSK. As in the analytic cases, we performed 10 trials for each case and compared the computational time and RE. For our GE-KPLS model we set $m = 1, \ldots, 5$ and use one principal component for all the problems, except for P_1, P_2, and P_3 where we use at most 2, 2, and 4 extra points, respectively.

Figure 4 shows the numerical results for the eight functions. As in the plots for the analytic cases, each line has two points corresponding to n_1 and n_2 sampling points (kriging, KPLS, and KPLSK) or to $n_1/2$ and $n_2/2$ (GE-KPLSm). The models are color coded as indicated in the legend. The means of the computational time and RE are given in Table 4 in Appendix C.

Overall, GE-KPLS gives a more accurate solution except for P_2. For P_2, GE-KPLS2 is almost as accurate as kriging (the model giving the best result) using 10 and 20 sampling points, respectively, with an RE of 0.1866 for the former and 0.1859 for the latter. All the GE-KPLS results have a lower time and/or a lower error than the kriging results for the same sampling cost. This means that despite the augmented size of the GE-KPLS correlation matrices, the time required to build these models is lower than that for the kriging model. The efficiency of GE-KPLS is due to the reduced number of hyperparameters that must be estimated, which is one in our case, compared to d for kriging. In addition, our use of the PLS coefficients to rescale the correlation matrix results in better accuracy. For example, we need only 0.12 s for P_5 with 80 sampling points to build a GE-KPLS5 model with a relative error of 0.3165 compared to 38.9 s for a kriging model (best error given the benchmark) with a relative error of 0.4050. In Figure 4, we notice that the time required to train some models for P_1 and P_2 seems higher as the number of sampling points decreases, e.g., KPLSK for P_2. This is due to the fast construction of the model for problems with low dimensions, and the difference in time for different numbers of sampling points is less than 10^{-3} s.

This study shows that GE-KPLS is accurate and computationally efficient for various engineering applications. In addition, the user can choose the best compromise between time and error. For example, the

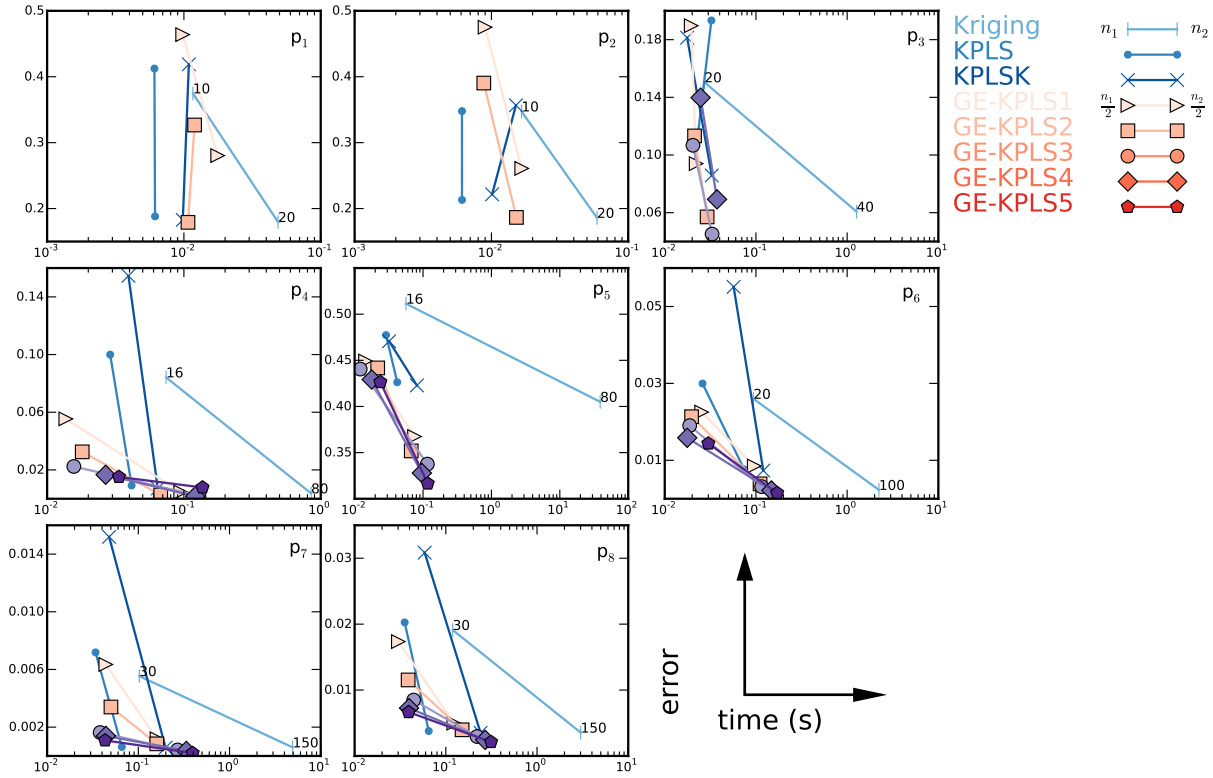


Figure 4: Summary of results for all models applied to the engineering problems, with 10 trials for each case. The models are color coded as shown in the legend. The best trade-off (time vs. error) is always obtained by a GE-KPLS model.

user can start with the construction of a GE-KPLS1 model. Guided by these results, the user can choose a reasonable trade-off between error and time and add more approximating points to achieve this compromise. Another way to select m is to define a threshold and to keep approximating points with higher PLS coefficients. We also note that m should be chosen carefully with regard to the final goal. For example, if the goal is to use a surrogate model within an iterative optimization design process, it is better to select a GE-KPLS model with a relatively low number of approximating points, since the many new sampling points that are close to each other will quickly deteriorate the condition number of the correlation matrix. However, if the final goal is to construct an accurate model over the design space, the number of approximating points can be relatively high.

5 Limitations of GE-KPLS

Despite the numerous advantages of our method, there are some limitations. The main question, as for most methods in the literature, concerns the values to use for the model parameters. For example, the step size of FOTA can influence the final results, since this parameter is sensitive to the type of problem and the sampling points.

In terms of implementation, the current toolbox for building GE-KPLS cannot handle problems with both many dimensions and many sampling points. This is because of the memory required during the inversion of the correlation matrix. An approximation of this memory limit is given by $n = 30550d^{-0.427}$.

6 Conclusions

We have developed a novel approach that uses gradient information at the sampling points to efficiently build accurate kriging surrogate models for high-dimensional problems. The proposed approach differs from classical strategies, such as direct and indirect GEK, in that we exploit the gradient information without dramatically increasing the size of the correlation matrix, and we reduce the number of hyperparameters. We applied the PLS method to each sampling point and selected the most relevant approximating points to include in the correlation matrix based on the PLS information. Through some elementary operations on the kernels, we accelerated the construction of the model by using the average PLS information to reduce the number of hyperparameters. Thus, our approach scales well with the number of independent variables by reducing the number of hyperparameters and with the number of sampling points by selecting only the relevant approximating points.

To demonstrate the computational efficiency and accuracy of our model, we presented a series of comparisons for analytic functions and engineering problems with varying numbers of dimensions and sampling points. We performed three studies. We first compared our approach using m extra points for $m = 1, \ldots, 5$ and $m = d$ to the indirect GEK and ordinary kriging models. With $m = d$, which is the same number of extra points used for GEK, we showed the usefulness of the PLS method in terms of computational time and error. This study also demonstrated the effectiveness and accuracy of the GE-KPLSm models for $m = 1, \ldots, 5$. In some cases, the GE-KPLS model is over three times more accurate and over 3200 times faster than the indirect GEK model. In the second study, we increased the number of sampling points for the analytic functions to compare GE-KPLSm for $m = 1, \ldots, 5$ with kriging, KPLS, and KPLSK. This confirmed the results of the first study: GE-KPLS had excellent performance in terms of both computational time and RE. For the first function and in comparison with both the kriging and KPLS models, the accuracy is an order of magnitude better. The third study focused on eight engineering functions using GE-KPLSm for $m = 1, \ldots, 5$ and the kriging and KPLS(K) models. GE-KPLS gave more accurate models for seven problems regardless of the number of dimensions or sampling points. The improvement in terms of RE provided by GE-KPLS is up to 9% in some cases.

GE-KPLS is able to manage the number of approximating points to avoid ill-conditioned matrices, with an important gain in terms of time and accuracy. This flexibility is convenient in real applications, especially when the surrogate model is used within an iterative sampling method, e.g., design optimization. The user can progressively reduce the number of approximating points after a certain number of iterations to minimize the risk of ill-conditioned problems; this feature is not available with standard GEK. For the effective use of our method, we recommend a preliminary analysis of the step parameter. Unfortunately, this parameter

is problem-dependent and is impossible to guess in advance; this is a common difficulty for gradient-based methods.

Although the mean of the PLS coefficients gives good results for our benchmarks, it would be interesting to use the median instead since it is less sensitive to outliers. It would also be interesting to test alternatives to the Box–Behnken design.

The code for the metamodels and test functions used in this paper is available in a GitHub repository under an open-source license[1], allowing other researchers to reproduce these results, add other metamodels and functions, and improve the existing code.

A Definition of the engineering cases

The analytical expressions for the engineering cases are as follows.

A.1 P_1, P_2, and P_3

The responses are the deflection δ, bending stress σ, and shear stress τ, respectively, of a welded beam problem [13].

$$\mathrm{P}_1 : \delta = \frac{2.1952}{t^3 b},$$

$$\mathrm{P}_2 : \sigma = \frac{504000}{t^2 b},$$

$$\mathrm{P}_3 : \tau = \sqrt{\frac{\tau'^2 + \tau''^2 + l\tau'\tau''}{\sqrt{0.25\,(l^2 + (h+t)^2)}}},$$

where

$$\tau' = \frac{6000}{\sqrt{2}hl} \text{ and } \quad \tau'' = \frac{6000(14 + 0.5l)\sqrt{0.25\,(l^2 + (h+t)^2)}}{2\left[0.707hl\left(\frac{l^2}{12} + 0.25(h+t)^2\right)\right]}.$$

The table below gives the ranges of the input variables.

Input variable	Range
h	$[0.125, 1]$
b	$[0.1, 1]$
l, t	$[5, 10]$

A.2 P_4

This problem characterizes the flow of water through a borehole that is drilled from the ground surface through two aquifers [39]. The water flow rate (m^3/yr) is given by

$$\mathrm{P}_4 : y = \frac{2\pi T_u\,(H_u - H_l)}{\ln\left(\frac{r}{r_w}\right)\left[1 + \frac{2LT_u}{\ln\left(\frac{r}{r_w}\right) r_w^2 K_w} + \frac{T_u}{T_l}\right]}.$$

The table below gives the ranges of the input variables.

Input variable	Range	Input variable	Range
r_w	$[0.05, 0.15]$	r	$[100, 50000]$
T_u	$[63070, 115600]$	H_u	$[990, 1110]$
T_l	$[63.1, 116]$	H_l	$[700, 820]$
L	$[1120, 1680]$	K_w	$[9855, 12045]$

A.3 P_5

This function represents the position of a robot arm [3]:

$$P_5 : y = \sqrt{\left(\sum_{i=1}^{4} L_i \cos\left(\sum_{j=1}^{i} \theta_j\right)\right)^2 + \left(\sum_{i=1}^{4} L_i \sin\left(\sum_{j=1}^{i} \theta_j\right)\right)^2}.$$

The table below gives the ranges of the input variables.

Input variable	Range
L_i	$[0, 1]$
θ_j	$[0, 2\pi]$

A.4 P_6

This is an estimate of the weight of a light aircraft wing [14]:

$$P_6 : y = 0.036 S_w^{0.758} W_{fw}^{0.0035} \left(\frac{A}{\cos^2 \Lambda}\right) q^{0.006} \lambda^{0.04} \left(\frac{100tc}{\cos \Lambda}\right)^{-0.3} (N_z W_{dg})^{0.49} + S_w W_p.$$

The table below gives the ranges of the input variables.

Input variable	Range	Input variable	Range
S_w	$[150, 200]$	W_{fw}	$[220, 300]$
A	$[6, 10]$	Λ	$[-10, 10]$
q	$[16, 45]$	λ	$[0.5, 1]$
tc	$[0.08, 0.18]$	N_z	$[2.5, 6]$
W_{dg}	$[1700, 2500]$	W_p	$[0.025, 0.08]$

A.5 P_7 and P_8

These are the weight and lowest natural frequency of a torsion vibration problem [32]:

$$P_7 : y = \sum_{i=1}^{3} \lambda_i \pi L_i \left(\frac{d_i}{2}\right)^2 + \sum_{j=1}^{2} \rho_j \pi T_j \left(\frac{D_j}{2}\right)^2,$$

$$P_8 : y = \frac{\sqrt{\frac{-b-\sqrt{b^2-4c}}{2}}}{2\pi},$$

where

$$K_i = \frac{\pi G_i d_i}{32 L_i}, \quad M_j = \frac{\rho_j \pi t_j D_j}{4g},$$

$$J_j = 0.5 M_j \frac{D_j}{2},$$

$$b = -\left(\frac{K_1 + K2}{J_1} + \frac{K_2 + K3}{J_2}\right),$$

and

$$c = \frac{K_1 K_2 + K_2 K_3 + K_3 K_1}{J_1 J_2}.$$

The table below gives the ranges of the input variables.

Input variable	Range	Input variable	Range
d_1	$[1.8, 2.2]$	L_1	$[9, 11]$
G_1	$[105300000, 128700000]$	λ_1	$[0.252, 0.308]$
d_2	$[1.638, 2.002]$	L_2	$[10.8, 13.2]$
G_2	$[5580000, 6820000]$	λ_2	$[0.144, 0.176]$
d_3	$[2.025, 2.475]$	L_3	$[7.2, 8.8]$
G_3	$[3510000, 4290000]$	λ_3	$[0.09, 0.11]$
D_1	$[10.8, 13.2]$	t_1	$[2.7, 3.3]$
ρ_1	$[0.252, 0.308]$	D_2	$[12.6, 15.4]$
t_2	$[3.6, 4.4]$	ρ_1	$[0.09, 0.11]$

B Results for the analytic cases

Table 3: Mean of the error values (upper table) and computational times (lower table) for kriging, KPLS, KPLSK, and GE-KPLSm for $m = 1, \cdots, 5$ with 10 trials. The best values are highlighted in bold blue type.

	d	n–$\frac{n}{2}$	kriging	KPLS	KPLSK	GE-KPLS1	GE-KPLS2	GE-KPLS3	GE-KPLS4	GE-KPLS5
y_1	10	100–50	0.0092	0.1043	0.1134	0.0020	**0.0011**	0.0013	0.0029	0.0044
		20–10	0.2976	0.2563	0.2555	0.1752	0.1556	0.1405	0.1201	**0.0903**
	100	1000–500	0.0296	0.0615	0.0562	0.0081	0.0051	0.0041	0.0032	**0.0026**
		200–100	0.0805	0.0818	0.0817	0.0454	0.0302	0.0233	0.0207	**0.0194**
y_2	10	100–50	0.0618	0.1393	0.1475	0.0623	0.0182	**0.0110**	0.0123	0.0116
		20–10	0.4532	0.3787	0.3297	0.2976	0.2903	0.2766	0.2325	**0.1920**
	100	1000–500	0.0637	0.0723	0.0695	0.0285	0.0126	0.0097	0.0083	**0.0080**
		200–100	0.0984	0.0956	0.0950	0.0952	0.0678	0.0517	0.0441	**0.0398**
y_1	10	100–50	18.57	**0.07**	0.09	0.12	0.10	0.13	0.16	0.22
		20–10	0.15	**0.02**	0.04	0.03	0.03	0.06	0.04	0.05
	100	1000–500	109.59	**2.97**	33.59	35.31	36.91	42.23	45.11	43.04
		200–100	9.58	**0.15**	2.59	6.99	7.06	12.12	12.49	12.42
y_2	10	100–50	6.39	**0.06**	0.11	0.09	0.11	0.09	0.15	0.16
		20–10	0.26	**0.02**	0.05	0.03	0.03	0.06	0.07	0.05
	100	1000–500	107.70	**3.12**	33.24	35.08	36.49	115.24	109.76	117.65
		200–100	8.52	**0.28**	2.39	24.92	23.89	24.40	23.95	16.37

C Results for the engineering cases

Table 4: Mean of the error values (upper table) and computational times (lower table) for kriging, KPLS, KPLSK, and GE-KPLSm for $m = 1, \cdots, 5$ with 10 trials. The best values are highlighted in bold blue type.

	d	n–$\frac{n}{2}$	kriging	KPLS	KPLSK	GE-KPLS1	GE-KPLS2	GE-KPLS3	GE-KPLS4	GE-KPLS5
P_1	2	20–10	0.1796	0.1881	0.1827	0.2804	**0.1791**	–	–	–
		10–5	0.3747	0.4124	0.4188	0.4639	**0.3268**	–	–	–
P_2	2	20–10	**0.1859**	0.2132	0.2216	0.2609	0.1866	–	–	–
		10–5	**0.3474**	0.3478	0.3567	0.4751	0.3904	–	–	–
P_3	4	40–20	0.0607	0.0981	0.0906	0.0940	0.0571	**0.0451**	0.0692	–
		20–10	0.1504	0.1933	0.1813	0.1897	0.1132	**0.1067**	0.1398	–
P_4	8	80–40	0.0037	0.0118	0.0091	0.0046	0.0026	0.0019	**0.0017**	0.0078
		16–8	8.41	0.0999	0.1546	0.0554	0.0325	0.0224	0.0166	**0.0151**
P_5	8	80–40	0.4050	0.4347	0.4296	0.3674	0.3518	0.3376	0.3278	**0.3165**
		16–8	0.5114	0.4773	0.4707	0.4493	0.4418	0.4405	0.4292	**0.4264**
P_6	10	100–50	0.0023	0.0101	0.0086	0.0085	0.0039	0.0031	0.0022	**0.0015**
		20–10	0.0260	0.0300	0.0551	0.0225	0.0213	0.0190	0.0158	**0.0144**
P_7	15	150–75	0.0006	0.0008	0.0007	0.0012	0.0008	0.0004	0.0003	**0.0002**
		30–15	0.0055	0.0072	0.0152	0.0063	0.0034	0.0016	0.0014	**0.0011**
P_8	15	150–75	0.0035	0.0041	0.0037	0.0050	0.0040	0.0029	0.0024	**0.0021**
		30–15	0.0191	0.0202	0.0308	0.0173	0.0115	0.0085	0.0073	**0.0067**
P_1	2	20–10	0.05	**0.006**	0.01	0.02	0.01	–	–	–
		10–5	0.01	**0.006**	0.01	0.01	0.01	–	–	–
P_2	2	20–10	0.06	**0.006**	0.01	0.02	0.01	–	–	–
		10–5	0.02	**0.006**	0.01	0.01	0.01	–	–	–
P_3	4	40–20	1.27	**0.01**	0.02	0.02	0.03	0.03	0.04	–
		20–10	0.03	**0.01**	0.02	0.02	0.02	0.02	0.02	–
P_4	8	80–40	0.85	**0.03**	0.07	0.09	0.07	0.13	0.12	0.14
		16–8	0.07	0.03	0.04	**0.01**	0.02	0.02	0.03	0.03
P_5	8	80–40	38.90	**0.03**	0.084	0.07	0.07	0.12	0.10	0.12
		16–8	0.06	0.03	0.03	**0.01**	0.02	**0.01**	0.02	0.02
P_6	10	100–50	2.23	**0.04**	0.10	0.10	0.11	0.11	0.15	0.17
		20–10	0.09	0.03	0.06	0.02	0.02	0.02	**0.018**	0.03
P_7	15	150–75	4.92	**0.05**	0.19	0.16	0.16	0.27	0.34	0.39
		30–15	0.10	**0.03**	0.05	0.04	0.05	0.27	0.34	0.39
P_8	15	150–75	3.01	**0.05**	0.18	0.12	0.15	0.22	0.27	0.31
		30–15	0.12	**0.03**	0.06	**0.03**	0.04	0.04	0.04	0.04

References

[1] L. Abraham. pydoe: The Experimental Design Package for Python, 2009. URL https://pythonhosted.org/pyDOE/index.html. https://pythonhosted.org/pyDOE/index.html.

[2] P. R. Alberto and F. G. González. Partial Least Squares Regression on Symmetric Positive-Definite Matrices. *Revista Colombiana de Estadística*, 36(1):177–192, 2012.

[3] J. An and A. Owen. Quasi-Regression. *Journal of Complexity*, 17(4):588–607, 2001.

[4] D. Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, New York, NY, USA, 2012. ISBN 0521518148, 9780521518147.

[5] N. Bartoli, M. A. Bouhlel, I. Kurek, R. Lafage, T. Lefebvre, J. Morlier, R. Priem, V. Stilz, and R. Regis. Improvement of Efficient Global Optimization with Application to Aircraft Wing Design. In *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Washington, D.C., 2016. AIAA-2016-4001.

[6] M. A. Bouhlel, N. Bartoli, J. Morlier, and A. Otsmane. An Improved Approach for Estimating the Hyperparameters of the Kriging Model for High-Dimensional Problems Through the Partial Least Squares Method. *Mathematical Problems in Engineering*, vol. 2016, Article ID 6723410, 2016.

[7] M. A. Bouhlel, N. Bartoli, A. Otsmane, and J. Morlier. Improving Kriging Surrogates of High-Dimensional Design Models by Partial Least Squares Dimension Reduction. *Structural and Multidisciplinary Optimization*, 53(5):935–952, 2016. ISSN 1615-1488.

[8] M. A. Bouhlel, N. Bartoli, R. G. Regis, A. Otsmane, and J. Morlier. Efficient global optimization for high-dimensional constrained problems by using the kriging models combined with the partial least squares method. *Engineering Optimization*, 0(0):1–16, 2018. doi:10.1080/0305215X.2017.1419344. URL https://doi.org/10.1080/0305215X.2017.1419344.

[9] G. Box, J. Hunter, and W. Hunter. *Statistics for Experimenters: Design, Innovation, and Discovery*. Wiley Series in Probability and Statistics. Wiley-Interscience, 2005. ISBN 9780471718130. URL https://books.google.ca/books?id=oYUpAQAAMAAJ.

[10] S. Choi, H. Chung, and J. Alonso. Design of Low-Boom Supersonic Business Jet With Evolutionary Algorithms Using Adaptive Unstructured Mesh. In *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference.*, Palm Springs, California, 2004. AIAA-2004-1758.

[11] H. S. Chung and J. Alonso. Design of a Low-Boom Supersonic Business Jet Using Cokriging Approximation Models. *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Multidisciplinary Analysis Optimization Conferences*, AIAA-2002-5598, 2002.

[12] N. Cressie. Spatial Prediction and Ordinary Kriging. *Mathematical Geology*, 20(4):405–421, 1988.

[13] K. Deb. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186:311–338, 2000.

[14] A. I. J. Forrester, A. Sóbester, and A. J. Keane. *Engineering Design via Surrogate Modeling: A Practical Guide*. Wiley, 2008.

[15] I. E. Frank and J. H. Friedman. A Statistical View of Some Chemometrics Regression Tools. *Technometrics*, 35:109–148, 1993.

[16] R. Haftka, D. Villanueva, and A. Chaudhuri. Parallel Surrogate-Assisted Global Optimization with Expensive Functions–A Survey. *Structural and Multidisciplinary Optimization*, 54:3–13, 2016.

[17] I. S. Helland. On the Structure of Partial Least Squares Regression. *Communication in Statistics - Simulation and Computation*, 17:581–607, 1988.

[18] S. Jeong, M. Murayama, and K. Yamamoto. Efficient Optimization Design Method Using Kriging Model. *Journal of Aircraft*, 42(2):413–420, 2005.

[19] D. R. Jones. A Taxonomy of Global Optimization Methods Based on Response Surfaces. *Journal of Global Optimization*, 21(4):345–383, 2001.

[20] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

[21] G. K. W. Kenway, G. J. Kennedy, and J. R. R. A. Martins. Scalable Parallel Approach for High-Fidelity Steady-State Aeroelastic Analysis and Derivative Computations. *AIAA Journal*, 52(5):935–951, 2014. doi:10.2514/1.J052255.

[22] J. Kleijnen, W. Van Beers, and I. Van Nieuwenhuyse. Constrained Optimization in Expensive Simulation: Novel Approach. *European Journal of Operational Research*, 202(1):164–174, 2010.

[23] J. Kleijnen, W. Beers, and I. Nieuwenhuyse. Expected Improvement in Efficient Global Optimization Through Bootstrapped Kriging. *Journal of Global Optimization*, 54(1):59–73, 2012.

[24] J. P. C. Kleijnen. *Design and Analysis of Simulation Experiments*, volume 230. Springer, 2015.

[25] J. P. C. Kleijnen. Regression and Kriging Metamodels with their Experimental Designs in Simulation: A Review. *European Journal of Operational Research*, 256(1):1–16, 2017. doi:10.1016/j.ejor.2016.06.04.

[26] D. G. Krige. A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand. *Journal of the Chemical, Metallurgical and Mining Society*, 52:119–139, 1951.

[27] J. Laurenceau and P. Sagaut. Building Efficient Response Surfaces of Aerodynamic Functions with Kriging and Cokriging. *AIAA Journal*, 46:2:498–507, 2008.

[28] R. M. Lewis. Using Sensitivity Information in the Construction of Kriging Models for Design Optimization. *AIAA-98-4799. 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Multidisciplinary Analysis Optimization Conferences*, pages 730–737, 1998.

[29] R. P. Liem, G. K. Kenway, and J. R. R. A. Martins. Multi-Point, Multi-Mission, High-Fidelity Aerostructural Optimization of a Long-Range Aircraft Configuration. In *Proceedings of the 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Indianapolis, IN, Sept. 2012. doi:10.2514/6.2012-5706.

[30] R. P. Liem, G. K. W. Kenway, and J. R. R. A. Martins. Multimission Aircraft Fuel Burn Minimization via Multipoint Aerostructural Optimization. *AIAA Journal*, 53(1):104–122, January 2015. doi:10.2514/1.J052940.

[31] R. P. Liem, C. A. Mader, and J. R. R. A. Martins. Surrogate Models and Mixtures of Experts in Aerodynamic Performance Prediction for Aircraft Mission Analysis. *Aerospace Science and Technology*, 43:126–151, June 2015. 10.1016/j.ast.2015.02.019.

[32] W. Liping, B. Don, W. Gene, and R. Mahidhar. A Comparison of Metamodeling Methods Using Practical Industry Requirements. In *Proceedings of the 47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Newport, RI, 2006.

[33] W. Liu. *Development of Gradient-Enhanced Kriging Approximations for Multidisciplinary Design Optimization*. PhD thesis, University of Notre Dame, 2003.

[34] J. L. Loeppky, S. Sacks, and W. J. Welch. Choosing the Sample Size of a Computer Experiment: A Practical Guide. *Technometrics*, 51(4):366–376, 2009. doi:10.1198/TECH.2009.08040. URL `https://doi.org/10.1198/TECH.2009.08040`.

[35] C. A. Mader, J. R. R. A. Martins, J. J. Alonso, and E. van der Weide. ADjoint: An Approach for the Rapid Development of Discrete Adjoint Solvers. *AIAA Journal*, 46(4):863–873, 2008. doi:10.2514/1.29123.

[36] K. V. Mardia and A. J. Watkins. On Multimodality of the Likelihood in the Spatial Linear Model. *Biometrika*, 76(2):289, 1989. doi:10.1093/biomet/76.2.289. URL `+http://dx.doi.org/10.1093/biomet/76.2.289`.

[37] J. R. R. A. Martins and J. T. Hwang. Review and Unification of Methods for Computing Derivatives of Multidisciplinary Computational Models. *AIAA Journal*, 51(11):2582–2599, 2013. doi:10.2514/1.J052184.

[38] G. Matheron. Principles of Geostatistics. *Economic Geology*, 58(8):1246–1266, 1963.

[39] M. D. Morris, T. J. Mitchell, and D. Ylvisaker. Bayesian Design and Analysis of Computer Experiments: Use of Derivatives in Surface Prediction. *Technometrics*, 35(3):243–255, 1993.

[40] J. Ollar, C. Mortished, R. Jones, J. Sienz, and V. Toropov. Gradient Based Hyper-Parameter Optimisation for Well Conditioned Kriging Metamodels. *Structural and Multidisciplinary Optimization*, 55: 2029–2044, 2017.

[41] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Rettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

[42] O. Pironneau. On Optimum Design in Fluid Mechanics. *Journal of Fluid Mechanics*, 64(1):97–110, 1974.

[43] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, USA, 2006.

[44] J. Sacks, S. B. Schiller, and W. J. Welch. Designs for Computer Experiments. *Technometrics*, 31(1): 41–47, 1989.

[45] J. Sacks, W. J. Welch, W. J. Mitchell, and H. P. Wynn. Design and Analysis of Computer Experiments. *Statistical Science*, 4(4):409–435, 1989.

[46] S. Sakata, F. Ashida, and M. Zako. Structural Optimization Using Kriging Approximation. *Computer Methods in Applied Mechanics and Engineering*, 192(417):923–939, 2003.

[47] T. W. Simpson, T. M. Mauery, J. J. Korte, and F. Mistree. Kriging Models for Global Approximation in Simulation-Based Multidisciplinary Design Optimization. *AIAA Journal*, 39(12):2233–2241, 2001.

[48] T. W. Simpson, J. D. Poplinski, P. N. Koch, and J. K. Allen. Metamodels for Computer-Based Engineering Design: Survey and Recommendations. *Engineering with Computers*, 17(2):129–150, 2001.

[49] M. Tenenhaus. *La Régression PLS: Théorie et Pratique*. Éd. Technip, 1998.

[50] D. J. J. Toal, N. W. Bressloff, and A. J. Keane. Geometric Filtration using POD for Aerodynamic Design Optimization. In *26th AIAA Applied Aerodynamics Conference*. August 2008. URL http://uos-app00353-si.soton.ac.uk/59225/.

[51] S. Ulaganathan, I. Couckuyt, T. Dhaene, E. Laermans, and J. Degroote. On the Use of Gradients in Kriging Surrogate Models. In *Proceedings of the 2014 Winter Simulation Conference*, pages 2692–2701, Savannah, GA, USA, December 7–10, 2014. doi:10.1109/WSC.2014.7020113. URL https://doi.org/10.1109/WSC.2014.7020113.

[52] F. A. C. Viana, T. W. Simpson, V. Balabanov, and V. Toropov. Metamodeling in Multidisciplinary Design Optimization: How Far Have We Really Come? *AIAA Journal*, 52:670–690, 2014. doi:10.2514/1.J052375.

[53] W. J. Welch, R. J. Buck, J. Sacks, H. P. Wynn, T. J. Mitchell, and M. D. Morris. Screening, Predicting, and Computer Experiments. *Technometrics*, 34(1):15–25, 1992.