High Dimensional Kriging Metamodelling Utilising Gradient Information

S. Ulaganathan[a,*], I. Couckuyt[a], T. Dhaene[a], J. Degroote[b], E. Laermans[a]

[a] *Ghent University - iMINDS, Department of Information Technology (INTEC), Gaston Crommenlaan 8, 9050 Ghent, Belgium*
[b] *Ghent University, Department of Flow, Heat and Combustion Mechanics, Sint - Pietersnieuwstraat 41, 9000 Ghent, Belgium*

Abstract

Kriging-based metamodels are popular for approximating computationally expensive black-box simulations, but suffer from an exponential growth of required training samples as the dimensionality of the problem increases. While a Gradient Enhanced Kriging metamodel with less training samples is able to approximate more accurately than a Kriging-based metamodel, it is prohibitively expensive to build for high dimensional problems. This limits the applicability of Gradient Enhanced Kriging for high dimensional metamodelling. In this work, this limitation is alleviated by coupling Gradient Enhanced Kriging with High Dimensional Model Representation. The approach, known as Gradient Enhanced Kriging based High Dimensional Model Representation, is accompanied by a highly efficient sequential sampling scheme LOLA-Voronoi and is applied to various high dimensional benchmark functions and one real-life simulation problem of varying dimensionality (10D-100D). Test results show that the combination of inexpensive gradient information and the high dimensional model representation can break or at least loosen the limitations associated with high dimensional Kriging metamodelling.

Keywords: Metamodelling, Kriging, Gradient Enhancement, LOLA-Voronoi, HDMR, FSI

*Corresponding author
Email address: selvakumar.ulaganathan@ugent.be (S. Ulaganathan)

1. Introduction

Usage of accurate high-fidelity physics-based computer simulations over controlled real-life experiments has become increasingly common in the past decades. However, the computational complexity of one single run of such computer simulations with multiple inputs and outputs can be very high. This computational complexity can be reduced with metamodelling where the expensive computer simulation code is replaced by a computationally cheap approximation model. Metamodelling has gained much attention among researchers over the past two decades. Researchers have proposed various strategies to provide accurate metamodels with minimal computational cost spent on collecting the training data, such as incorporating secondary information, employing data of varying fidelities, intelligent sampling schemes etc. [1, 2, 3, 4, 5]

Metamodelling is successfully applied to model deterministic low dimensional problems. An overview of various metamodelling approaches applied to model low dimensional problems is given by Simpson et al. [6], Jin et al. [7] and Wang et al. [8]. Kriging is popular for approximating deterministic data. Kriging was popularised by Sacks et al. [9] and further explored by various researchers [10, 11, 12, 13, 14]. Although Kriging is successfully applied over the years to model low dimensional problems, modelling of high dimensional parameter spaces is often limited by the exponentially growing number of training samples, known as the "curse of dimensionality". One of the approaches to alleviate a part of this issue is to exploit the gradient information. Direct gradient incorporation in Kriging along with function data, later known as direct Gradient Enhanced Kriging (GEK), was introduced by Morris et al. [10]. An alternative formulation of GEK where the gradient data is used to augment the function data was introduced by Chung et al. [11]. GEK is also subject to the "curse of dimensionality" as the size of the "correlation" matrix grows and handling of ill-conditioning of the GEK "correlation" matrix also becomes a major challenge. Although this issue can be alleviated to some extent, by discarding

sample points that contribute the least information to the GEK "correlation" matrix, as demonstrated in [14], this problem still persists in high dimensions limiting the potential usage of GEK for high dimensional metamodelling.

In general, solving high dimensional problems can be addressed with various metamodelling techniques such as projection pursuit regression [15], multivariate adaptive regression splines (MARS) [16], additive Kriging models [17], high dimensional model representation (HDMR) [18, 19] etc. Various metamodelling methodologies to tackle high dimensional problems are discussed in [20].

The HDMR decomposes a high dimensional function $f(\mathbf{x})$ integrable in space D^d (d is the number of input variables, or dimensionality) with a unique finite hierarchical correlated function expansion in terms of $f(\mathbf{x})$. HDMR was initially introduced by [18] and has since been extensively explored by various researchers [21, 22, 23, 24]. Two of the major variants of HDMR, ANOVA-HDMR and Cut-HDMR, were introduced by Rabitz et al. [21, 22]. RS-HDMR was illustrated by Wang et al. [23] and Li et al. [24]. ANOVA-HDMR is constructed by evaluating multidimensional integrals of the output which is usually achieved by Monte Carlo simulations. Although Monte Carlo simulations are viable in high dimensional problems, it calls for a significant number of evaluations to attain a reasonable level of metamodel accuracy. In contrary, Cut-HDMR only calls for simple arithmetic computations and provides least expensive metamodels with similar accuracy level as other HDMR variations. Moreover, ANOVA-HDMR is very helpful when it comes to estimating the contribution of variance of each component function to the overall variance of the output. However, Cut-HDMR is an exact representation of the function $f(\mathbf{x})$ to be modelled in slices (e.g., lines and hyper-planes) passing through a selected cut point in the input space. Thus, the selection of particular HDMR variant is based on what to be known about the function to be modelled while considering the sample budget. In this work, the Cut-HDMR variant is chosen to model high dimensional problems due to its arithmetic simplicity while providing accurate high

dimensional metamodels with least sample budget [20, 25, 27]. A Cut-HDMR variant using radial basis functions was recently introduced in [20] along with an adaptive sampling and model construction algorithms with promising results.

The principal contribution of this work is the introduction of a novel and efficient approach to deal with Kriging metamodelling for high dimensional problems and reaping the advantage of exploiting gradient information. In this context, the additional gradient data when available cheaply (in terms of computational cost and computational resources) are incorporated in Kriging models (termed as Gradient Enhanced Kriging based HDMR) along with function data. Subsequently, the Kriging models with gradient data are compared with Kriging models without gradient data (termed as Ordinary Kriging based HDMR) and also with Radial basis function based models without gradient data (termed as RBF-HDMR) from [27]. The principal motivation for the introduction of Gradient Enhanced Kriging based HDMR (GEK-HDMR) is to investigate how much reduction in number of training sample points can be achieved while modelling high dimensional problems by incorporating the cheaply available gradient data. One of the other intentions of this paper is the introduction of an accompanying modelling strategy which is used to reduce the overall number of training sample points by identifying/classifying the existing correlations within variables of the high dimensional problem to be modelled and distributing the training samples accordingly. It is achieved by inducting LOLA-Voronoi [26] sequential sampling scheme into the accompanying modelling strategy. The feasibility of the LOLA-Voronoi sequential sampling scheme is demonstrated by comparing it with a "Maximin" space-filling criterion based sequential sampling scheme [44]. Further, the feasibility of the accompanying modelling strategy is demonstrated by comparing it with a modelling strategy where the existing correlations within variables of the high dimensional problem are not identified while the training samples are distributed using Latin Hypercube Design (LHD) sampling scheme. Various analytical benchmark functions and one real-life simulation problem of varying dimensionality (10D-100D) are used as test problems.

The remaining part of this paper is organised as follows. The mathematical formulations of GEK-HDMR are elaborated in Section 2. Section 3 discusses the LOLA-Voronoi sequential sampling methodology. Section 4 is dedicated to the introduction of the LOLA-Voronoi sampling based GEK-HDMR modelling algorithm. Section 5 lists the test problems and the error metrics used to assess the metamodel accuracy. Test results are presented and discussed in Section 6 followed by the conclusions.

2. GEK-HDMR

2.1. Cut-HDMR

A HDMR expresses the mapping between the input variables $\mathbf{x} = (x_1, x_2, \ldots, x_d)$ and the output $f(\mathbf{x})$ as [25],

$$f(\mathbf{x}) = f^0 + \sum_{i=1}^{d} f_i(x_i) + \sum_{1 \le i < j \le d} f_{ij}(x_i, x_j) + \sum_{1 \le i < j < k \le d} f_{ijk}(x_i, x_j, x_k) + \ldots + f_{12\ldots d}(x_1, x_2, \ldots, x_d). \tag{1}$$

Here f^0 is a constant term representing the zero-order effect of $f(\mathbf{x})$. The first-order term $f_i(x_i)$ represents the effect of variable x_i acting independently, either linearly or non-linearly, upon the output $f(\mathbf{x})$. The second-order term $f_{ij}(x_i, x_j)$ represents the cooperative effect of variables x_i and x_j, either linearly or non-linearly, upon the output $f(\mathbf{x})$. The subsequent higher order terms represent the interactive effects of input variables acting together upon the output $f(\mathbf{x})$. The last term $f_{12\ldots d}(x_1, x_2, \ldots, x_d)$ gives any residual dependence of all the input variables combined together to influence the output $f(\mathbf{x})$. For most well-defined high dimensional systems, the higher order interactions are expected to be weak and a second-order HDMR,

$$f(\mathbf{x}) = f^0 + \sum_{i=1}^{d} f_i(x_i) + \sum_{1 \le i < j \le d} f_{ij}(x_i, x_j), \tag{2}$$

can often provide an accurate representation of $f(\mathbf{x})$ [21]. In terms of computational cost and accuracy of HDMR modelling, a Cut-HDMR, which is used in this work, is more attractive than other variants of HDMR [25]. In Cut-HDMR, the component functions are estimated with respect to a cutting point $\mathbf{x}^0 = (x_1^0, x_2^0, \dots x_d^0)$ and are expressed as,

$$f^0 = f(\mathbf{x}^0), \tag{3}$$

$$f_i(x_i) = f(x_i, \mathbf{x}_i^0) - f^0, \tag{4}$$

$$f_{ij}(x_i, x_j) = f(x_i, x_j, \mathbf{x}_{ij}^0) - f(x_i) - f(x_j) - f^0, \tag{5}$$

where $\mathbf{x}_i^0$ and $\mathbf{x}_{ij}^0$ are $\mathbf{x}^0$ without elements x_i and x_{ij}, respectively; $\mathbf{x}^0$, $(x_i, \mathbf{x}_i^0) = (x_1^0, x_2^0, \dots, x_i, \dots, x_d^0)$ and $(x_i, x_j, \mathbf{x}_{ij}^0) = (x_1^0, x_2^0, \dots, x_i, \dots, x_j, \dots, x_d^0)$ are zero-order, first-order and second-order model training point(s), respectively, and $f(\mathbf{x}^0)$, $f(x_i, \mathbf{x}_i^0)$ and $f(x_i, x_j, \mathbf{x}_{ij}^0)$ are the corresponding function values, respectively. Modelling of first-order $f(x_i, \mathbf{x}_i^0)$ and second-order $f(x_i, x_j, \mathbf{x}_{ij}^0)$ component functions with Gradient Enhanced Kriging (GEK) leads to a second-order GEK-HDMR which can be expressed as,

$$f(\mathbf{x}) = f^0 + \sum_{i=1}^{d} \hat{f}_i(x_i) + \sum_{1 \le i < j \le d} \hat{f}_{ij}(x_i, x_j), \tag{6}$$

where

$$\hat{f}_i(x_i) = \hat{f}(x_i, \mathbf{x}_i^0) - f^0, \tag{7}$$

$$\hat{f}_{ij}(x_i, x_j) = \hat{f}(x_i, x_j, \mathbf{x}_{ij}^0) - \hat{f}(x_i) - \hat{f}(x_j) - f^0, \tag{8}$$

where $\hat{f}(x_i, \mathbf{x}_i^0)$ and $\hat{f}(x_i, x_j, \mathbf{x}_{ij}^0)$ represent the GEK models of $f(x_i, \mathbf{x}_i^0)$ and $f(x_i, x_j, \mathbf{x}_{ij}^0)$, respectively.

2.2. Gradient Enhanced Kriging (GEK)

2.2.1. Mathematical Formulation

Since many publications on Kriging and Gradient Enhanced Kriging can be found in the literature (see Refs. [1, 9, 14, 28, 29, 30, 31, 32, 33]), we present

only the resultant, however self-contained, equations without their proofs. The mathematical form of GEK is composed of two terms: The first part, $\hat{\mu}$, represents a trend function and the second part is a realisation of a stationary Gaussian random process which captures the local deviations from the trend function. The GEK predictor at a prediction point $\mathbf{x}^*$ for an arbitrary function $f(\mathbf{x})$ can be expressed as,

$$\hat{f}(\mathbf{x}^*) = \hat{\mu} + \dot{\boldsymbol{\psi}}^T \dot{\boldsymbol{\Psi}}^{-1}(\dot{\mathbf{y}} - \mathbf{f}\hat{\mu}), \tag{9}$$

where

$$\dot{\boldsymbol{\Psi}} = \begin{Bmatrix} \boldsymbol{\Psi} & \frac{\partial \boldsymbol{\Psi}}{\partial x_1^{(i)}} & \cdots & \frac{\partial \boldsymbol{\Psi}}{\partial x_v^{(i)}} & \cdots & \frac{\partial \boldsymbol{\Psi}}{\partial x_d^{(i)}} \\ \frac{\partial \boldsymbol{\Psi}}{\partial x_1^{(j)}} & \frac{\partial^2 \boldsymbol{\Psi}}{\partial x_1^{(i)} \partial x_1^{(j)}} & \cdots & \frac{\partial^2 \boldsymbol{\Psi}}{\partial x_1^{(i)} \partial x_v^{(j)}} & \cdots & \frac{\partial^2 \boldsymbol{\Psi}}{\partial x_1^{(i)} \partial x_d^{(j)}} \\ \vdots & \vdots & \ddots & \vdots & \cdots & \vdots \\ \frac{\partial \boldsymbol{\Psi}}{\partial x_u^{(j)}} & \frac{\partial^2 \boldsymbol{\Psi}}{\partial x_1^{(j)} \partial x_u^{(i)}} & \cdots & \frac{\partial^2 \boldsymbol{\Psi}}{\partial x_u^{(i)} \partial x_v^{(j)}} & \cdots & \frac{\partial^2 \boldsymbol{\Psi}}{\partial x_u^{(i)} \partial x_d^{(j)}} \\ \vdots & \vdots & \cdots & \vdots & \ddots & \vdots \\ \frac{\partial \boldsymbol{\Psi}}{\partial x_d^{(j)}} & \frac{\partial^2 \boldsymbol{\Psi}}{\partial x_1^{(j)} \partial x_d^{(i)}} & \cdots & \frac{\partial^2 \boldsymbol{\Psi}}{\partial x_v^{(j)} \partial x_d^{(i)}} & \cdots & \frac{\partial^2 \boldsymbol{\Psi}}{\partial x_d^{(i)} \partial x_d^{(j)}} \end{Bmatrix}, \tag{10}$$

$$\dot{\boldsymbol{\psi}} = \left(\boldsymbol{\psi}^T, \left(\frac{\partial \boldsymbol{\psi}}{\partial x_1} \right)^T, ..., \left(\frac{\partial \boldsymbol{\psi}}{\partial x_d} \right)^T \right)^T, \tag{11}$$

$$\dot{\mathbf{y}} = \left(\mathbf{y}^T, \left(\frac{\partial \mathbf{y}}{\partial x_1} \right)^T, ..., \left(\frac{\partial \mathbf{y}}{\partial x_d} \right)^T \right)^T, \tag{12}$$

$$\mathbf{f} = \left(1_1, ...1_{n_s}, 0_{n_s+1}, ..., 0_{(d+1)n_s} \right)^T, \tag{13}$$

where $\boldsymbol{\psi}$ contains the correlation between the training sample data and a prediction point $\mathbf{x}^*$; $\mathbf{y}$ is the column vector of function values; $\boldsymbol{\Psi}$ is the correlation matrix which contains the correlation between the training sample points; n_s is the number of training sample points; $\dot{\boldsymbol{\psi}}$ contains the correlation between the training sample data and the prediction point $\mathbf{x}^*$ and their derivatives; and $\dot{\mathbf{y}}$ contains both the function values and gradients of the training sample data.

2.2.2. Correlation Function

Correlation between any two sample points is expressed by the correlation function of choice. Various correlation functions can be employed in Kriging to capture the correlation between any two sample points [34, 35, 31, 36]. As correlation functions must be differentiated twice in GEK to provide the correlation between gradient observations, we limit ourselves to the Matérn $\frac{5}{2}$ correlation function. The Matérn $\frac{5}{2}$ correlation function can be expressed as [37],

$$\psi_{\nu=\frac{5}{2}}(d^{\nu}) = (1 + \sqrt{5}a + \frac{5a^2}{3})exp\left(-\sqrt{5}a\right), \tag{14}$$

where $a = \sqrt{\sum_{m=1}^{d} \theta_m (d_m^{\nu})^2}$ and $d^{\nu} = |x_m^i - x_m^j|$. In order to express the correlation between function and gradient data and correlation between gradient data and themselves, the analytical gradient and Hessian of Equation 14 are required in GEK. The analytical expressions for the gradient and the Hessian of the Matérn $\frac{5}{2}$ correlation function with respect to $\mathbf{x}$ can be expressed as,

$$\frac{\partial \mathbf{\Psi}^{(i,j)}}{\partial x_u^{(j)}} = \frac{5\theta d^{\nu}(\sqrt{5}a + 1)exp\left(-\sqrt{5}a\right)}{3} \tag{15}$$

and

$$\frac{\partial^2 \mathbf{\Psi}^{(i,j)}}{\partial x_u^{(i)} \partial x_v^{(j)}} = \begin{cases} \frac{-25\theta_u \theta_v d_u^{\nu} d_v^{\nu} exp\left(-\sqrt{5}a\right)}{3} & \text{if } u \neq v \\ \left[\frac{-25\theta^2 (d^{\nu})^2 + 5\theta(\sqrt{5}a+1)}{3}\right] exp\left(-\sqrt{5}a\right) & \text{if } u = v, \end{cases} \tag{16}$$

respectively. The notations $\frac{\partial \mathbf{\Psi}^{(i,j)}}{\partial x_u^{(j)}}$ and $\frac{\partial^2 \mathbf{\Psi}^{(i,j)}}{\partial x_u^{(i)} \partial x_v^{(j)}}$ denote the correlation between function and u^{th} dimension gradients and correlation between u^{th} dimension gradients and v^{th} dimension gradients, respectively. The direction of differentiation is denoted by i and j with $x^{(i)}$ and $x^{(j)}$ denoting two different samples. For more elaborate information on deriving the analytical gradients and Hessians of the correlation matrix $\dot{\boldsymbol{\psi}}$ for a correlation function of choice, the reader is referred to [3, 14, 32, 38].

2.2.3. GEK Model Fitting

The constant trend function $\hat{\mu}$ for GEK is calculated by means of generalised least squares,

$$\hat{\mu} = (\mathbf{f}^T \dot{\mathbf{\Psi}}^{-1} \mathbf{f})^{-1} \mathbf{f}^T \dot{\mathbf{\Psi}}^{-1} \dot{\mathbf{y}}. \tag{17}$$

The symbol θ in correlation function (see Equation 14) represents the hyper-parameters of the GEK model ($\theta_m, m = 1, ..., d$) which show how far the influence of a sample point extends. Lower values of θ_m denote higher correlation among the sample points while the higher values denote that function values can change rapidly over a small region. The values of the hyper-parameters are obtained by maximizing the concentrated likelihood function,

$$\phi = \frac{-(d+1)n_s \ ln(\hat{\sigma}^2) - ln|\dot{\mathbf{\Psi}}|}{2}, \tag{18}$$

where $\hat{\sigma}^2$ is the estimated GEK variance which can be expressed as,

$$\hat{\sigma}^2 = \left(\frac{(\dot{\mathbf{y}} - \mathbf{f}\hat{\mu})^T \dot{\mathbf{\Psi}}^{-1} (\dot{\mathbf{y}} - \mathbf{f}\hat{\mu})}{(d+1)n_s} \right). \tag{19}$$

For more information on how the values of θ can influence the overall metamodel accuracy, the reader is referred to [1, 38]. A detailed analysis of the mathematical aspects (derivation and optimisation) of the concentrated likelihood function is discussed in [14] and [33].

3. LOLA-Voronoi

Sequential sampling strategies are commonly used in metamodelling as it is often difficult to know the appropriate size of the training data a priori. Various sequential and adaptive sampling techniques are discussed in [39, 40, 41, 42].

A recent sampling technique, known as LOLA-Voronoi, is introduced in [26]. LOLA-Voronoi is a sequential sampling technique which strategically performs trade-off between exploration and exploitation during the sampling process to achieve globally accurate metamodels [26, 43]. Exploration denotes filling the design space as uniformly as possible with sample points whereas exploitation, in the case of LOLA-Voronoi, denotes more sample points concentrated in the nonlinear regions of the design space. In the LOLA-Voronoi sampling technique,

exploration is performed with a criterion using Monte Carlo Voronoi approximation whereas exploitation is performed with a criterion based on LOcal Linear Approximations of the system (LOLA).

The LOLA-Voronoi sampling algorithm starts with an initial set of n_l sample points. It is best practice to include the corner points of the design space in the initial set of sample points. Then, the hybrid score for a sample point $\mathbf{x_i} \in D^d$ is computed as [26],

$$H(\mathbf{x_i}) = V(\mathbf{x_i}) + \frac{E(\mathbf{x_i})}{\sum_{j=1}^{n_l} E(\mathbf{x_j})}, \tag{20}$$

where $E(\mathbf{x_i})$ is the non-linearity measure which is calculated using LOLA and $V(\mathbf{x_i})$ is the Voronoi cell size which is computed using the Monte Carlo Voronoi approximation. The gradient of a function f at a given point $\mathbf{x_i} \in D^d$ can represent the best local linear approximation of the function around $\mathbf{x_i}$ and is expressed as,

$$\Delta f(\mathbf{x_i}) = \left(\frac{\partial f}{\partial x_i^1}, \frac{\partial f}{\partial x_i^2}, \ldots, \frac{\partial f}{\partial x_i^d} \right). \tag{21}$$

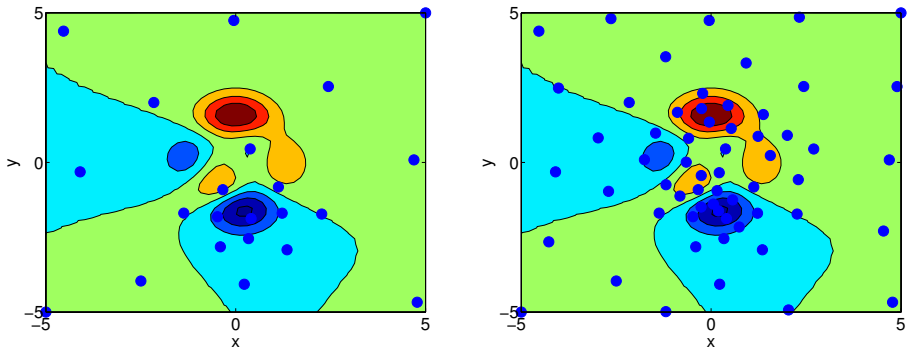
The gradient at a sample point is estimated by applying least-squares regression to the neighbouring sample points. The measure of non-linearity $E(\mathbf{x_i})$ is then estimated from how much the true output value at the neighbours differs from the local linear approximation [26]:

$$E(\mathbf{x_i}) = \sum_{j=1}^{n_{nl}} |f(\mathbf{x_{ij}}) - (f(\mathbf{x_i}) + \Delta f(\mathbf{x_i})(\mathbf{x_{ij}} - \mathbf{x_i}))|, \tag{22}$$

where n_{nl} is the number of neighbouring sample points which are chosen to represent the region around $\mathbf{x_i}$. In order to estimate Voronoi cell size $V(\mathbf{x_i})$, n_r random test points $\mathbf{x^r_{j=1,\ldots,n_r}} \in D^d$ are generated. Among the test points, the one which is closest to $\mathbf{x_i}$ is identified and the (relative) size of the corresponding Voronoi cell is estimated as [26],

$$V(\mathbf{x_i}) = V(\mathbf{x^r_{j \leftarrow closest}}) + \frac{1}{n_r n_l}. \tag{23}$$

Once the non-linearity measure and the Voronoi cell size are calculated, the sample points are ranked according to the values of the hybrid score (i.e., accord-



(a) After 22 iterations of LOLA-Voronoi (b) After 60 iterations of LOLA-Voronoi

Figure 1: Contour plots of the Peaks function with two intermediate data sets generated by the LOLA-Voronoi algorithm. A good trade-off between exploration of the design space and exploitation of the dynamic regions can be observed.

ing to how undersampled the design space is and/or how nonlinear the function behaviour is). Subsequently, new samples are generated around the highest ranked sample points. The LOLA-Voronoi sampling technique is demonstrated with a simple 2D example in Figure 1.

4. LOLA-Voronoi sampling based GEK-HDMR modelling

The steps involved in a second-order GEK-HDMR model construction are described as follows:

Step I: First-order GEK-HDMR model

(1) Choose a cut point $\mathbf{x}^0 = (x_1^0, x_2^0, \ldots x_d^0)$. In the absence of prior knowledge about the function to be modelled, it is usually chosen in the vicinity of the centre of the design space. Estimate $f^0 = f(\mathbf{x}^0)$.

(2) The LOLA-Voronoi algorithm is applied to each first-order component function separately, $f(x_i, \mathbf{x}_i^0) = (x_1^0, x_2^0, \ldots, x_i, \ldots, x_d^0)$. The algorithm starts with two corner sample points of the design space. By estimating the corresponding function and gradients of the function values at the cor-

ner sample points, the first-order component functions can be modelled as $\hat{f}_i(x_i) = \hat{f}(x_i, \mathbf{x}_i^0) - f^0$ with GEK.

(3) The LOLA-Voronoi algorithm chooses a new sample point based on the hybrid score as mentioned in the Section 3. If the GEK approximation is able to accurately predict the function value at the chosen sample point, then the sampling and the modelling are terminated for the current first-order component function. Otherwise, the GEK model is updated with the chosen sample point and a new sample point (or test point) is chosen by the sequential sampling algorithm based on the hybrid score on the updated dataset. If the GEK model achieves a sufficient accuracy level (for example, the relative error is less than a value prescribed by the user), then the sampling and the re-modelling are terminated for the current first-order component function. Otherwise, the process is continued until convergence or the number of sample points reaches the maximum number of sample points. The same procedure is followed for modelling the remaining first-order components, and finally, a first-order GEK-HDMR model is built.

Unlike the current case where the sampling and the modelling of each first-order component function are carried out sequentially, these activities (sampling and modelling/re-modelling) can also be performed in parallel for each first-order component function. Although the final results will be ideal in both the cases (sequential and parallel) for a chosen accuracy level (for example, the relative error is less than a value prescribed by the user), the parallelisation process can certainly result in significant time efficiency when the function evaluation is computationally expensive. Moreover, the parallelisation process enables one to come up with a first-order GEK-HDMR which is as accurate as possible at any given time when the other stopping conditions (convergence of accuracy to the chosen accuracy level and/or sample budget) are not yet met. Modelling of second-order component functions can also be performed in parallel. However, it becomes important to identify the second-order component functions to be modelled before the

parallelisation by classifying the existing two-variable correlation between variables of the problem to be modelled.

Step II: Second order GEK-HDMR model

(4) A new dataset $(x_i, x_j, \mathbf{x}_{ij}^0) = (x_1^0, x_2^0, \ldots, x_i, \ldots, x_j, \ldots, x_d^0)$ with $(d(d-1))/2$ two-variable combinations is formed by combining the thus-far obtained first-order training sample points $(x_i, \mathbf{x}_i^0)$ and $(x_j, \mathbf{x}_j^0)$. Subsequently, function values are predicted for the new dataset with the first-order GEK-HDMR model built and are compared with the true function values at randomly chosen points. If the accuracy of the first-order GEK-HDMR model on this new dataset is good enough (for example, the relative error is less than a value prescribed by the user), then no higher order terms are modelled as the second-order correlation is observed to be weak. Otherwise, the following step is executed.

(5) All the two-variable combinations in the new dataset are ranked according to the accuracy of the first-order GEK-HDMR model on each two-variable combination. The highest ranked combination corresponds to the largest value of the error criterion (i.e., the error is larger than the user defined value). This indicates that these combinations exhibit second-order correlation, and hence, the corresponding second-order component functions need to be modelled in the following step. This step allows discarding nonexistent or insignificant correlations completely.

(6) The LOLA-Voronoi algorithm is applied to each second-order component function $f(x_i, x_j, \mathbf{x}_{ij}^0)$ separately with the similar procedure as mentioned in the steps (2) and (3). The only difference is the generation of two dimensional samples, and GEK models the second-order component functions as $\hat{f}(x_i, x_j) = \hat{f}(x_i, x_j, \mathbf{x}_{ij}^0) - \hat{f}(x_i) - \hat{f}(x_j) - f^0$.

(7) Finally, a second-order GEK-HDMR model is built with all the first-order component functions and the selected second-order component functions.

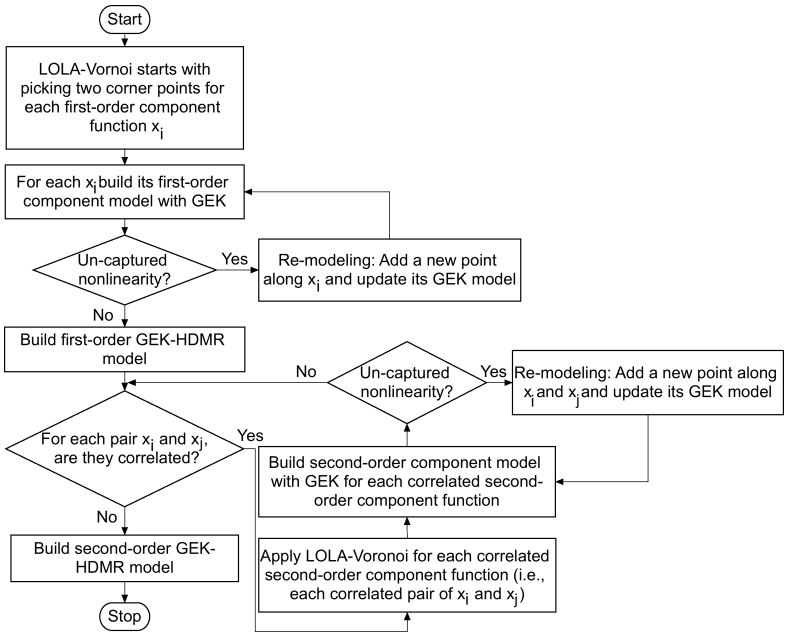


Figure 2: Flowchart of LOLA-Voronoi sampling based GEK-HDMR modelling

Figure 2 shows a simplified work-flow of the LOLA-Voronoi sequential sampling based GEK-HDMR modelling process. The proposed approach is illustrated on a simple three-dimensional ($d = 3$) example function,

$$f(\mathbf{x}) = x_1 + x_2^2 + x_1 x_3 - 4, \quad 0 \leq x_i \leq 1. \tag{24}$$

1. The GEK-HDMR modelling process starts with choosing a cut point, in this case, $\mathbf{x}^0 = (0.5, 0.5, 0.5)$ and estimating the zero-th order effect, $f^0 = -3.0$.
2. The LOLA-Voronoi algorithm starts with the two corner points 0 and 1. After estimating the function and gradient values of the corner points, the initial first-order GEK models $\hat{f}(x_i, \mathbf{x}_i^0)$ are built.
3. A test point, 0.5, is chosen by the LOLA-Voronoi algorithm to assess the accuracy of $\hat{f}(x_i, \mathbf{x}_i^0)$. It can be seen from Figure 3 that the function behaviour is linear in the first and third directions. Hence, the sampling and the modelling processes terminate for $f(x_1, \mathbf{x}_1^0)$ and $f(x_3, \mathbf{x}_3^0)$ as the GEK

models are already accurate. Whereas $\hat{f}(x_2, \mathbf{x}_2^0)$ is not accurate enough as the function behaviour is non-linear in the second direction. Hence, the chosen test point is used as an additional sample point and $f(x_2, \mathbf{x}_2^0)$ is re-modelled with the new dataset (0,1,0.4951). Then the accuracy of the updated $\hat{f}(x_2, \mathbf{x}_2^0)$ is now assessed with a new test point chosen by the LOLA-Voronoi algorithm. As $\hat{f}(x_2, \mathbf{x}_2^0)$ now reaches the user defined accuracy level, absolute error < 0.01, the re-modelling is terminated. Finally, a first-order GEK-HDMR is built as $\hat{f}(\mathbf{x}) = f^0 + \sum_{i=1}^{d}(\hat{f}(x_i, \mathbf{x}_i^0) - f^0)$ using 11 sample points.

4. By performing steps (4) and (5) (an absolute error of 0.01 is used in both steps), it was identified (with 3 test points) that the second-order correlation exist only between variables x_1 and x_3. Hence, the other two-variable correlations are neglected and only $f(x_1, x_3, \mathbf{x}_{13}^0)$ is modelled with 4 sample points (see Figure 3d). Finally, the complete second-order GEK-HDMR model is built, in this case, as $\hat{f}(\mathbf{x}) = f^0 + \sum_{i=1}^{d}(\hat{f}(x_i, \mathbf{x}_i^0) - f^0) + (\hat{f}(x_1, x_3, \mathbf{x}_{13}^0) - \hat{f}(x_1) - \hat{f}(x_3) - f^0)$.

Although the LOLA-Voronoi sequential sampling scheme is inducted into the accompanying sampling strategy, in this paper, to generate new sample point(s) or test point(s) in steps (3) - (6), any other sequential sampling scheme can also be used in the place of LOLA-Voronoi sequential sampling scheme. The reason behind employing LOLA-Voronoi sampling scheme, in this paper, is the fact that it can generate the new sample point(s)/ or test point(s) mentioned in steps (3) - (6) with respect to the behaviour of the function to be modelled by achieving a trade-off between exploration and exploitation as mentioned in Section 3. This can be an added advantage at times over the usage of other existing sequential sampling schemes although it needs not be true at all times [26]. In order to gain more insight from this fact, a "Maximin" space-filling criterion based sequential sampling scheme [44] (termed as Maximin sampling scheme in the rest of this paper) is employed while constructing GEK-HDMR models and the results are discussed in Section 6.

The Maximin sampling scheme is a sequential sampling scheme where the sample points are generated based on the Maximin space-filling criterion (or intersite distance) by maximizing the smallest (Euclidean) distance between any two sets of sample points in the design space:

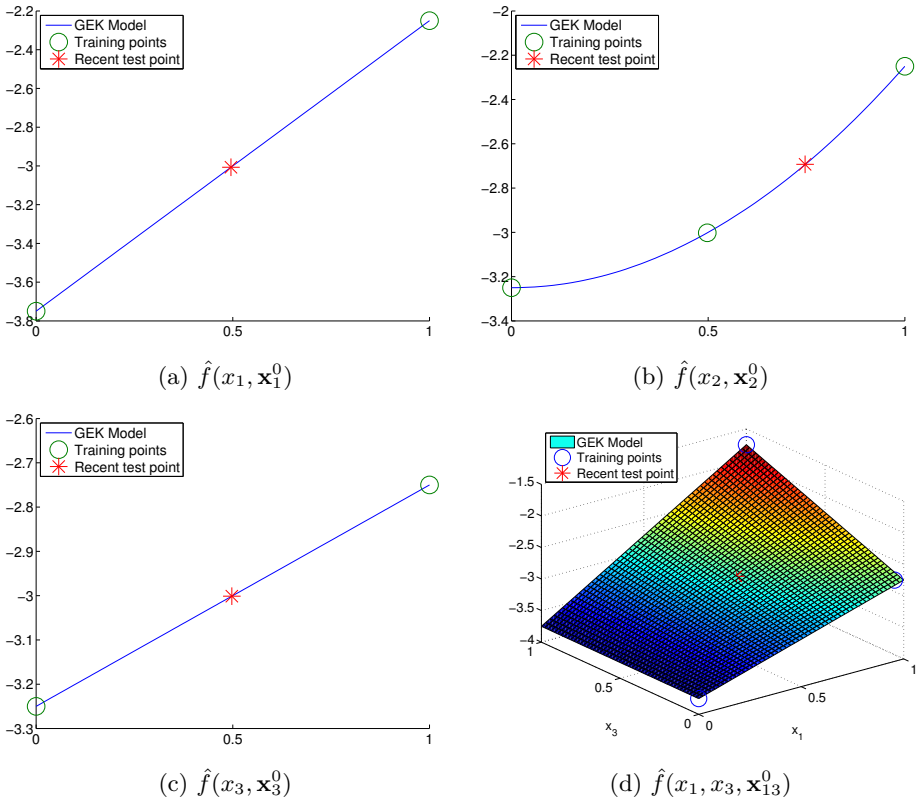
$$min_{x_i,x_j \in D^d} \sqrt{\sum_{k=1}^{d} \left(|x_i^k - x_j^k|^2\right)}. \tag{25}$$

The Maximin sampling scheme has an advantage of generating samples sequentially by bringing a trade-off between exploration (based on Maximin criterion) and exploitation (based on projected distance criterion), thus significantly reducing the overall training sample points as compared to LHD. However, in contrary to LOLA-Voronoi sampling scheme, the trade-off in Maximin sampling scheme is not bound by the behaviour of the function to be modelled. The Maximin sampling scheme starts with initial sample points (usually contains the corner points of the design space) and intervals are created in each dimension by sorting the values of the sample points in each dimension, and subtracting subsequent values. The middle point of the hypercube (has the best projected distance score) which is defined by the largest interval in each dimension is now considered as the best potential sample point. The second best potential sample point is the one created by replacing the one interval by the next largest, and so on. Once sufficient sample points are generated this way, a pattern search optimisation is performed in the 50 largest hypercubes, optimising towards the Maximin criterion. The optimisation is bound by a threshold parameter set by the user which controls the trade-off between exploration and exploitation.

5. Problem Formulation

5.1. Analytical Examples: Benchmark Test Problems

Five analytical benchmark functions, which are listed below, are used as test functions. The gradient values of the benchmark functions with respect to the design variables are analytically calculated.



(a) $\hat{f}(x_1, \mathbf{x}_1^0)$ (b) $\hat{f}(x_2, \mathbf{x}_2^0)$

(c) $\hat{f}(x_3, \mathbf{x}_3^0)$ (d) $\hat{f}(x_1, x_3, \mathbf{x}_{13}^0)$

Figure 3: GEK-HDMR modelling process. First-order and correlated second-order component functions and their corresponding GEK models for the 3D example function.

- Function1 [27]

$$f(\mathbf{x}) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45 \quad -10 \leq x_i \leq 10, i = 1, \ldots, 10 \tag{26}$$

- Function2 [27]

$$f(\mathbf{x}) = \sum_{i=1}^{16} \sum_{j=1}^{16} a_{ij}(x_i^2 + x_i + 1)(x_j^2 + x_j + 1) \quad 0 \leq x_i, x_j \leq 5, i, j = 1, \ldots, 16 \tag{27}$$

where,

$$a_{ij} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

- Function3 (Rosenbrock function) [45]

$$f(\mathbf{x}) = \sum_{i=0}^{15} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right] \qquad -2 \leq x_i \leq 2, i = 1, \ldots, 15 \tag{28}$$

- Function4 (Rosenbrock function) [45]

$$f(\mathbf{x}) = \sum_{i=0}^{30} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right] \qquad -2 \leq x_i \leq 2, i = 1, \ldots, 30 \tag{29}$$

- Function5 (Sphere function) [45]

$$f(\mathbf{x}) = \sum_{i=1}^{10} x_i^2 \qquad -1 \leq x_i \leq 1, i = 1, \ldots, 10 \tag{30}$$

5.2. Simulation Example: Fluid Structure Interaction Problem

A numerical simulator [46] that determines the difference between a given wall displacement and a calculated wall displacement for a given stiffness distribution along the length of an artery is used as the simulation example. The numerical simulator uses a simplified fluid-structure interaction (FSI) model to identify the stiffness distribution along the length of an artery. The FSI model is one-dimensional in an axisymmetric coordinate system, as depicted in Figure 4. It consists of various elastic segments, each with its own stiffness.

The cost function of the problem is the sum over all time steps and all elastic segments of the squared difference between the given radius (r which is shown in Figure 4) and the radius obtained during the simulation. The input variables to the solver are the unique parameters (s_i) which define the elasticity modulus (E_i) for each elastic segment as,

$$E_i = E_o \left(1 + \frac{1}{2} s_i\right), \qquad i = 1, \ldots, d \tag{31}$$

and vary from -1 to 1. Hence, the number of input variables is equal to the number of elastic segments present in the artery. The number of input variables can be increased by increasing the number of elastic segments present in the artery, which in turn increases the accuracy of the estimation of wall displacement. E_o in Equation 31 is a constant term whose value is provided to the solver. Inside the artery, there is an incompressible blood flow. Furthermore, the interaction between the blood flow and the elastic wall is taken into account. The governing flow equations and the structural equations, which are formulated, discretised and linearised in [46], are solved separately and the IQN-ILS algorithm [47] is used to perform the coupling iterations to obtain the solution of the coupled problem.

The gradients of the cost function are obtained by solving adjoint flow equations and adjoint structural equations and coupling them using IQN-ILS algo-

rithm [47]. In adjoint formulation, the computational cost of estimating gradients is irrespective of the dimensionality of the problem and is solely based on the implementation of the adjoint equations. In this problem, the computational cost of estimating additional gradients at a given sample point is negligible as compared to the computational cost of estimating the function value. For more information on this problem, the readers are referred to [46].

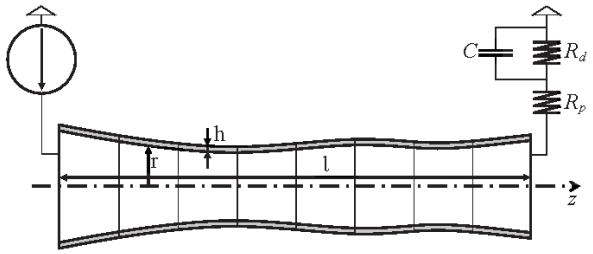


Figure 4: The one-dimensional and axisymmetric model for blood flow in an artery with the prescribed velocity at the inlet (left) and the Windkessel model at the outlet (right). The segments, radius r, wall thickness h and length ℓ are indicated

Problems with different number of input variables are defined by varying the number of elastic segments in the FSI model. Four such problems are defined with 10, 20, 50 and 100 elastic segments which correspond to "Function6", "Function7", "Function8" and "Function9', respectively, in Tables 2-10. The influence and interaction of each elastic segment with unique parameter s_i on cost function makes these problems high dimensional in nature. Moreover, the problems with dimensionality (d) larger than ten are usually considered as high dimensional problems in the context of engineering [27].

5.3. Metamodel Assessment

As Kriging based metamodels are statistically unbiased at training sample points, a validation data set which contains n_p number of untried uniformly distributed pseudorandom test points is generated, in order to estimate the accuracy of the metamodels. The validation data set contains $n_p = 10000$ and

$n_p = 500$ uniformly distributed pseudorandom test points for the analytical and the real-life test problems respectively. Once the validation data set is generated, the metamodel accuracy is assessed on the test points with four different error metrics: R Squared Error (R^2), Relative Average Absolute Error (RAAE), Relative Maximum Absolute Error (RMAE) and Normalised Root Mean Square Error (NRMSE), which can be expressed, respectively, as,

$$R^2 = 1 - \frac{\sum_{i=1}^{n_p} \left(y_t^i - \hat{y}^i\right)^2}{\sum_{i=1}^{n_p} \left(y_t^i - \overline{y}_t^i\right)^2}, \tag{32}$$

$$RAAE = \frac{\sum_{i=1}^{n_p} \left(y_t^i - \hat{y}^i\right)^2}{std(\boldsymbol{y}_t) n_p}, \tag{33}$$

$$RMAE = \frac{max(|y_t^1 - \hat{y}^1|, |y_t^2 - \hat{y}^2|, \ldots, |y_t^{n_p} - \hat{y}^{n_p}|)}{std(\boldsymbol{y}_t)}, \tag{34}$$

$$NRMSE = \frac{\sqrt{\frac{\sum_{i=1}^{n_p}\left(y_t^i - \hat{y}^i\right)^2}{n_p}}}{max(\boldsymbol{y}_t) - min(\boldsymbol{y}_t)}, \tag{35}$$

where $\boldsymbol{y}_t$ is the vector of actual function values, $\hat{\boldsymbol{y}}$ is the vector of estimated function values from the metamodel and $\overline{y}$ is the mean of the actual function values on n_p test points. R^2, RAAE and NRMSE show the overall metamodelling accuracy whereas RMAE is a local error metric. The values of RAAE, RMAE and NRMSE approach zero as the overall metamodel accuracy increases whereas high values are preferred for R^2.

During the metamodel(s) construction as explained in Section 4, an absolute error value of 0.1 is used to assess the accuracy of all first-order and second-order (including during correlation identification) function models in all the benchmark functions. In the case of real-life problems, an absolute error value of 0.01 is used to assess the accuracy of all first-order function models. The values for the absolute error are selected based on the ability of Kriging based metamodels in modelling the benchmark and the real-life problems. In general, a lower value of absolute error (paves the way for more accurate metamodels)

leads to significant increase in the number of overall training sample points. Thus, more care should be taken by considering the available sample budget and the required level of metamodel(s) accuracy while selecting an appropriate value for the absolute error. Absolute error measure, which is employed in this paper, can be replaced with any other feasible error measures such as relative error measure. A relative error measure is useful, especially, when the magnitude of the function values of the problem are not known a priori.

6. Results and Discussion

Tables 1-10 show the results of the benchmark functions and the real-life problems. The results of the benchmark functions represent the mean of 10 independent runs whereas the results of the real-life problems are limited to a single run due to the computational cost of actual function and gradient estimation.

In general, both LOLA-Voronoi and Maximin sampling based GEK-HDMR significantly reduce the number of training sample points required to provide accurate metamodels as compared to the LOLA-Voronoi and Maximin sampling based OK-HDMR. This is clearly the effect of incorporating additional gradient information at the training sample points. A training sample point for OK contains a function value only whereas a training sample point for GEK contains the function value and the gradient values in all the dimensions. GEK-HDMR benefits from the additional gradient information in two different ways: (i) gradient data provides secondary information about the function to be approximated. Hence, it serves as additional training data which can be advantageous. (ii) GEK is constrained by the fact that it must also interpolate the gradient data in addition to the function data. This leads to a more well-defined likelihood surface and thus easier estimation of the hyper-parameters, which in turn allows the correlation function to capture the underlying function as realistically as possible to the actual function to be approximated [37].

Moreover, the advantage of incorporating additional gradient data is significantly visible from the comparison with the RBF-HDMR in [27] (see "Function1" and "Function2" results in Tables 1, 3, 5 and 9) where both LOLA-Voronoi and Maximin sampling based GEK-HDMR are observed to provide equivalent or more accurate surrogate models with less training sample points than the RBF-HDMR in all the cases except the second-order model of "Function2". This exception indicates a certain degree of model overfitting and is essentially caused by two reasons: (i) errors introduced during the measure of correlation and non-linearity. Although in theory the incorporation of second-order correlation should result in more accurate surrogate models than incorporating the first-order correlation alone, the errors in the correlation identification and modelling may surpass the advantage of incorporating the second-order correlation. This occurs in problems "Function1" and "Function2". This also occurs in all the LHD sampling based OK-HDMR models of "Function1"-"Function4". However, in this case, the additional contribution to surpass the advantage of incorporating second-order correlation comes from the infeasible sample generation which is completely ignorant of the function behaviour and the non-linearity of the component functions. (ii) second and further higher order correlations may not exist in the function to be approximated. For example, in problems "Function5" - "Function9", incorporating the second-order effects in the HDMR modelling is observed to deteriorate the HDMR model accuracy (thus the second-order component functions are not modelled). Whereas in problems "Function3" and "Function4", incorporation of second-order correlation resulted in more accurate HDMR models than incorporating the first-order correlation alone.

However, it is important to note that among the OK-HDMR cases where the additional gradient data are not incorporated, the RBF-HDMR outperforms both the LOLA-Voronoi and the Maximin sampling based OK-HDMR. Additionally, it is observed that both the LOLA-Voronoi and the Maximin sequential

sampling schemes are fairly equal in performance. This fact allows us to persist the earlier intuition of inducting any other sequential sampling scheme (in the place of the proposed LOLA-Voronoi sampling scheme) into the accompanying modelling strategy can lead to metamodels of fair accuracy.

Further, to illustrate the advantages of the proposed modelling strategy, the overall cost of both the LOLA-Voronoi and the Maximin sampling based OK-HDMR/GEK-HDMR, in terms of n_s, is compared with that of the Latin Hypercube Design (LHD) sampling based OK-HDMR/GEK-HDMR in Tables 9 and 10. The overall cost (C) of a full first-order and second-order expansion of a standard Cut-HDMR are

$$C_1^{HDMR} = 1 + (d)\mathrm{n_s} \tag{36}$$

and

$$C_2^{HDMR} = 1 + (d)\mathrm{n_s} + \frac{d(d-1)\mathrm{n_s}}{2} \tag{37}$$

respectively. This is the overall cost incurred by the LHD sampling based OK-HDMR/GEK-HDMR modelling. Here the non-linearity of the first-order and the second-order component functions is not known a priori, thus an equal n_s is used for modelling all the first-order and the second-order component functions. However, n_s for the first-order component functions can be different (usually less) from n_s for the second-order component functions. Whereas the overall cost of a full first-order and second-order expansion of LOLA-Voronoi (or Maximin) sampling based OK-HDMR/GEK-HDMR are

$$C_1^{GEK-HDMR(LOLA)} = 1 + \sum_{i=1}^{d} \mathrm{n_{s_i}} \tag{38}$$

and

$$C_2^{GEK-HDMR(LOLA)} = 1 + \sum_{i=1}^{d} \mathrm{n_{s_i}} + \sum_{i=1}^{N} \mathrm{n_{s_i}} \tag{39}$$

respectively, where $N << (d(d-1))/2)$ represents the number of second-order correlations that exist in a given function of d dimensions. Here the non-linearity of the first-order and the second-order component functions is observed during

the modelling, thus each and every first-order and the second-order components functions are modelled with different n_s based on the non-linearity. By observing the results in Tables 9 and 10, one can see the reduction in n_s achieved by the LOLA-Voronoi sampling based OK-HDMR/GEK-HDMR over the LHD sampling based OK-HDMR/GEK-HDMR. This sample reduction is essentially achieved by the well controlled sample generation with LOLA-Voronoi algorithm in which the samples are generated based on the function behaviour. Thus, a highly nonlinear region is likely to have more sample points whereas samples are highly limited in linear regions of the design space. This is contrary to the LHD sampling based OK-HDMR/GEK-HDMR where the sample generation is completely ignorant of the function behaviour. However, when all the component functions exhibit a similar linearity/non-linearity, the LOLA-Voronoi algorithm based sampling may become less advantageous than its counterpart (see the results which correspond to "Function5"). It is important to note that the extra computational cost spent of acquiring gradient data is neglected while estimating the overall cost in Tables 9 and 10. This is due to the reason that the computational cost of estimating gradient data at a given sample point for all the problems employed in this paper is significantly less than the computational cost of estimating the function data. The gradient data for the benchmark problems are analytically calculated whereas the adjoint formulation is used to estimate the gradient data for the real-life problems. In adjoint formulation, the computational cost of estimating gradient data with respect to all the input variables at a given sample point is completely irrespective of the number of dimensions of the problem [48].

Table 1: Comparison of R Squared Error (R^2) on the validation data set. The values inside the parenthesis correspond to the first-order HDMR models. The corresponding overall cost of the first-order and the second-order HDMR models, in terms of number of training samples (n_s), can be obtained from Table 9 in one-to-one correspondence. Bold values correspond to the technique with the best overall performance.

Test Functions	OK-HDMR (LHD)	OK-HDMR (LOLA)	OK-HDMR (Maximin)	RBF-HDMR[27]
Function1	(0.47)	(0.75)0.74	(0.75)0.75	**(0.98)0.98**
Function2	(0.96)	(0.96)0.97	(0.96)0.97	**(0.96)0.99**
Function3	(0.62)	**(0.62)0.77**	(0.62)0.78	-
Function4	(0.62)	**(0.62)0.70**	(0.62)0.70	-
Function5	(1)	**(1)**	(1)	-
Function6	(0.64)	(0.64)	**(0.64)**	-
Function7	(0.59)	(0.59)	**(0.59)**	-
Function8	(0.65)	**(0.65)**	(0.65)	-
Function9	(0.81)	(0.82)	**(0.82)**	-

Table 2: Comparison of R Squared Error (R^2) on the validation data set. The values inside the parenthesis correspond to the first-order HDMR models. The corresponding overall cost of the first-order and the second-order HDMR models, in terms of number of training samples (n_s), can be obtained from Table 10 in one-to-one correspondence. Bold values correspond to the technique with the best overall performance.

Test Functions	GEK-HDMR (LHD)	GEK-HDMR (LOLA)	GEK-HDMR (Maximin)
Function1	(0.99)0.98	(0.99)0.99	**(0.99)0.99**
Function2	(0.96)0.92	(0.96)0.84	**(0.96)0.96**
Function3	(0.62)0.67	(0.62)0.79	**(0.62)0.78**
Function4	(0.62)0.94	(0.63)0.70	**(0.63)0.70**
Function5	(1)	(1)	**(1)**
Function6	(0.64)	**(0.65)**	(0.41)
Function7	(0.59)	(0.59)	**(0.59)**
Function8	(0.65)	(0.65)	**(0.65)**
Function9	(0.82)	**(0.82)**	(0.82)

Table 3: Comparison of Relative Average Absolute Error (RAAE) on the validation data set. The values inside the parenthesis correspond to the first-order HDMR models. The corresponding overall cost of the first-order and the second-order HDMR models, in terms of number of training samples (n_s), can be obtained from Table 9 in one-to-one correspondence. Bold values correspond to the technique with the best overall performance.

Test Functions	OK-HDMR (LHD)	OK-HDMR (LOLA)	OK-HDMR (Maximin)	RBF-HDMR[27]
Function1	(0.5916)	(0.398)0.402	(0.425)0.423	**(0.110)0.107**
Function2	(0.1430)	(0.143)0.126	(0.143)0.127	**(0.150)0.088**
Function3	(0.4746)	**(0.474)0.364**	(0.474)0.353	-
Function4	(0.4799)	**(0.479)0.430**	(0.480)0.430	-
Function5	(0.0013)	**(0.0013)**	(0.0008)	-
Function6	(0.4246)	(0.4248)	**(0.4254)**	-
Function7	(0.4843)	(0.4837)	**(0.4833)**	-
Function8	(0.4568)	**(0.4556)**	(0.4563)	-
Function9	(0.3113)	(0.3119)	**(0.3129)**	-

Table 4: Comparison of Relative Average Absolute Error (RAAE) on the validation data set. The values inside the parenthesis correspond to the first-order HDMR models. The corresponding overall cost of the first-order and the second-order HDMR models, in terms of number of training samples (n_s), can be obtained from Table 10 in one-to-one correspondence. Bold values correspond to the technique with the best overall performance.

Test Functions	GEK-HDMR (LHD)	GEK-HDMR (LOLA)	GEK-HDMR (Maximin)
Function1	(0.027)0.056	(0.040)0.040	**(0.036)0.038**
Function2	(0.143)0.262	(0.146)0.312	**(0.144)0.128**
Function3	(0.474)0.471	(0.47)0.35	**(0.474)0.352**
Function4	(0.479)0.195	(0.48)0.43	**(0.48)0.43**
Function5	(0.0012)	(0.0013)	**(0.0012)**
Function6	(0.4235)	**(0.4279)**	(0.6390)
Function7	(0.4847)	(0.4845)	**(0.4855)**
Function8	(0.4563)	(0.4569)	**(0.4568)**
Function9	(0.3107)	**(0.3105)**	(0.3106)

Table 5: Comparison of Relative Maximum Absolute Error (RMAE) on the validation data set. The values inside the parenthesis correspond to the first-order HDMR models. The corresponding overall cost of the first-order and the second-order HDMR models, in terms of number of training samples (n_s), can be obtained from Table 9 in one-to-one correspondence. Bold values correspond to the technique with the best overall performance.

Test Functions	OK-HDMR (LHD)	OK-HDMR (LOLA)	OK-HDMR (Maximin)	RBF-HDMR[27]
Function1	(2.407)	(1.25)1.22	(1.17)1.16	**(0.33)0.28**
Function2	(1.35)	(1.35)1.26	(1.356)1.363	**(0.91)0.25**
Function3	(3.73)	**(3.73)2.43**	(3.74)2.82	-
Function4	(2.56)	**(2.56)2.47**	(2.56)2.59	-
Function5	(0.006)	**(0.006)**	(0.004)	-
Function6	(2.486)	(2.469)	**(2.511)**	-
Function7	(2.042)	(2.019)	**(2.047)**	-
Function8	(2.605)	**(2.540)**	(2.610)	-
Function9	(2.152)	(2.136)	**(2.125)**	-

Table 6: Comparison of Relative Maximum Absolute Error (RMAE) on the validation data set. The values inside the parenthesis correspond to the first-order HDMR models. The corresponding overall cost of the first-order and the second-order HDMR models, in terms of number of training samples (n_s), can be obtained from Table 10 in one-to-one correspondence. Bold values correspond to the technique with the best overall performance.

Test Functions	GEK-HDMR (LHD)	GEK-HDMR (LOLA)	GEK-HDMR (Maximin)
Function1	(0.109)1.568	(0.19)0.20	**(0.16)0.19**
Function2	(1.361)0.804	(1.38)2.23	**(1.36)1.36**
Function3	(3.74)2.84	(3.73)2.54	**(3.74)2.81**
Function4	(2.57)0.99	(2.56)2.52	**(2.56)2.57**
Function5	(0.003)	(0.003)	**(0.003)**
Function6	(2.438)	**(2.421)**	(2.515)
Function7	(2.040)	(2.069)	**(2.082)**
Function8	(2.610)	(2.613)	**(2.616)**
Function9	(2.154)	**(2.160)**	(2.157)

Table 7: Comparison of Normalised Root Mean Square Error (NRMSE) on the validation data set. The values inside the parenthesis correspond to the first-order HDMR models. The corresponding overall cost of the first-order and the second-order HDMR models, in terms of number of training samples (n_s), can be obtained from Table 9 in one-to-one correspondence. Bold values correspond to the technique with the best overall performance.

Test Functions	OK-HDMR (LHD)	OK-HDMR (LOLA)	OK-HDMR (Maximin)	RBF-HDMR[27]
Function1	(0.1401)	(0.0912)0.0913	(0.0948)0.0950	not
Function2	(0.0273)	(0.0273)0.0240	(0.0273)0.0244	available
Function3	(0.0661)	**(0.0661)0.0510**	(0.0661)0.0503	-
Function4	(0.0727)	**(0.0727)0.0653**	(0.0727)0.0650	-
Function5	(0.0002)	**(0.0002)**	(0.0001)	-
Function6	(0.1209)	(0.1205)	**(0.1211)**	-
Function7	(0.1157)	(0.1156)	**(0.1156)**	-
Function8	(0.1119)	**(0.1116)**	(0.1118)	-
Function9	(0.0810)	(0.0808)	**(0.0806)**	-

Table 8: Comparison of Normalised Root Mean Square Error (NRMSE) on the validation data set. The values inside the parenthesis correspond to the first-order HDMR models. The corresponding overall cost of the first-order and the second-order HDMR models, in terms of number of training samples (n_s), can be obtained from Table 10 in one-to-one correspondence. Bold values correspond to the technique with the best overall performance.

Test Functions	GEK-HDMR (LHD)	GEK-HDMR (LOLA)	GEK-HDMR (Maximin)
Function1	(0.0070)0.0218	(0.01)0.0099	**(0.0088)0.0093**
Function2	(0.0274)0.0376	(0.0281)0.0547	**(0.0275)0.0245**
Function3	(0.0660)0.0611	(0.0661)0.05	**(0.0660)0.0502**
Function4	(0.0727)0.0272	(0.0727)0.0653	**(0.0727)0.0649**
Function5	(0.0002)	(0.0002)	**(0.0002)**
Function6	(0.1198)	**(0.1191)**	(0.1554)
Function7	(0.1157)	(0.1157)	**(0.1159)**
Function8	(0.1118)	(0.1119)	**(0.1119)**
Function9	(0.0809)	**(0.0811)**	(0.0809)

Table 9: Comparison of overall cost of the metamodels in terms of number of training samples (n_s). The values inside the parenthesis denote the overall cost of the first-order HDMR models. The overall cost of the second-order HDMR models (values outside the parenthesis) includes that of the first-order HDMR models. Bold values correspond to the technique with the best overall performance.

Test Functions	OK-HDMR (LHD)	OK-HDMR (LOLA)	OK-HDMR (Maximin)	RBF-HDMR[27]
Function1	(110)	(63)107	(61)78	**(34)40**
Function2	(192)	(175)422	(152)392	**(59)297**
Function3	(270)	**(250)427**	(276)516	-
Function4	(540)	**(514)703**	(561)801	-
Function5	(80)	**(78)**	(90)	-
Function6	(90)	(73)	**(69)**	-
Function7	(180)	(167)	**(159)**	-
Function8	(450)	**(343)**	(356)	-
Function9	(900)	(523)	**(519)**	-

Table 10: Comparison of overall cost of the metamodels in terms of number of training samples (n_s). The values inside the parenthesis denote the overall cost of the first-order HDMR models. The overall cost of the second-order HDMR models (values outside the parenthesis) includes that of the first-order HDMR models. Bold values correspond to the technique with the best overall performance.

Test Functions	GEK-HDMR (LHD)	GEK-HDMR (LOLA)	GEK-HDMR (Maximin)
Function1	(60)960	(25)38	**(26)35**
Function2	(96)816	(56)242	**(80)200**
Function3	(120)2010	(152)275	**(146)259**
Function4	(240)12420	(316)438	**(296)409**
Function5	(50)	(50)	**(50)**
Function6	(60)	**(44)**	(32)
Function7	(120)	(101)	**(86)**
Function8	(300)	(241)	**(202)**
Function9	(600)	**(355)**	(394)

Conclusions

The authors have detailed the implementation of a LOLA-Voronoi sequential sampling based Gradient Enhanced Kriging (GEK) metamodelling algorithm for high dimensional problems. Numerical results demonstrate that the combination of inexpensive gradient information and high dimensional model representation (HDMR) can be sufficient to break or at least attenuate the "curse of dimensionality". The HDMR already shifts the challenge from reducing the modelling cost of GEK (and also handling ill-conditioning of the GEK "correlation" matrix) to reducing the overall number of training samples of GEK-HDMR metamodelling by decomposing a high dimensional function into a combination of various first-order and second-order (or higher order if required) component functions. Moreover, the accompanying metamodelling strategy identifies and filters out the insignificant second-order component functions of the HDMR and, together with the LOLA-Voronoi algorithm, it effectively reduces the required overall training samples. The LOLA-Voronoi algorithm plays a significant role in keeping the overall cost of the GEK-HDMR metamodelling less by exploring the non-linearity of the function in single and/or higher dimensions and controlling the sample distribution accordingly. However, inducting any other sequential sampling scheme (in the place of the LOLA-Voronoi algorithm) into the accompanying metamodelling strategy can lead to metamodels of fair level of accuracy. This fact is persisted by the results of comparing LOLA-Voronoi sequential sampling based OK/GEK-HDMR models with Maximin sequential sampling based OK/GEK-HDMR models. Further, additional gradient incorporation at the training sample points can significantly help the GEK-HDMR to provide accurate metamodels with fewer training sample points than Ordinary Kriging based HDMR. GEK-HDMR essentially benefits from the additional gradient information in terms of improved hyper-parameters estimation and accurate interpolation of function data as the interpolation in GEK is constrained by both function and gradient data.

Acknowledgements

This research has been funded by the Interuniversity Attraction Poles Programme BESTCOM initiated by the Belgian Science Policy Office. Additionally, this research has been supported by the Fund for Scientific Research in Flanders (FWO-Vlaanderen). Ivo Couckuyt and Joris Degroote are post-doctoral research fellows of the Research Foundation Flanders (FWO).

References

References

[1] A. I. Forrester, A. Sóbester, A. J. Keane, Engineering Design via Surrogate Modelling: A Practical Guide, 1st Edition, Wiley, 2008.

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

[3] W. Yamazaki, M. P. Rumpfkeil, D. J. Mavriplis, Design optimization utilizing Gradient/Hessian enhanced surrogate model, in: 28th AIAA Applied Aerodynamics Conference, AIAA paper 2010-4363, Chicago, Illinois, USA, 2010.

[4] I. Couckuyt, T. Dhaene, P. Demeester, ooDACE toolbox: A flexible object-oriented kriging implementation, Journal of Machine Learning Research 15 (2014) 3183–3186.

[5] H. Zhao, Z. Yue, Y. Liu, Z. Gao, Y. Zhang, An efficient reliability method combining adaptive importance sampling and kriging metamodel, Applied Mathematical Modelling DOI: http://dx.doi.org/10.1016/j.apm.2014.10.015.

[6] T. Simpson, J. Poplinski, P. N. Koch, J. Allen, Metamodels for computer-based engineering design: Survey and recommendations, Engineering with Computers 17 (2) (2001) 129–150.

[7] R. Jin, W. Chen, T. W. Simpson, Comparative studies of metamodeling techniques under multiple modeling criteria, Structural and Multidisciplinary Optimization 23 (2000) 1–13.

[8] G. G. Wang, S. Shan, Review of metamodeling techniques in support of engineering design optimization, Journal of Mechanical Design 129 (4) (2006) 370–380.

[9] J. Sacks, W. J. Welch, T. J. Mitchell, H. P. Wynn, Design and analysis of computer experiments, Statistical Science 4 (4) (1989) 409–423.

[10] M. D. Morris, T. J. Mitchell, D. Ylvisaker, Bayesian design and analysis of computer experiments: Use of gradients in surface prediction, Technometrics 35 (3) (1993) 243–255.

[11] H.-S. Chung, J. J. Alonso, Using gradients to construct cokriging approximation models for high-dimensional design optimization problems, in: Problems, 40th AIAA Aerospace Sciences Meeting and Exhibit, AIAA, Reno, NV, 2002, pp. 2002–0317.

[12] W. Liu, Development of gradient-enhanced kriging approximations for multidisciplinary design optimisation, Ph.D. thesis, University of Notre Dame, Notre Dame, Indiana (2003).

[13] J. Laurenceau, P. Sagaut, Building efficient response surfaces of aerodynamic functions with kriging and cokriging, AIAA 46 (2) (2008) 498–507.

[14] K. R. Dalbey, Efficient and robust gradient enhanced kriging emulators, Tech. Rep. SAND2013-7022, Sandia National Laboratories, Albuquerque, New Mexico 87185 and Livermore, California 94550 (2013).

[15] J. Friedman, W. Stuetzle, Projection pursuit regression, Journal of the American Statistical Association 76 (372) (1981) 817–823.

[16] J. Friedman, Multivariate adaptive regressive splines, The Annals of Statistics 19 (1) (1991) 1–67.

[17] D. Nicolas, G. David, R. Olivier, C. Laurent, Additive covariance kernels for high-dimensional gaussian process modeling (2011). arXiv:1111.6233. URL http://arxiv.org/abs/1111.6233

[18] I. Sobol, Sensitivity estimates for nonlinear mathematical models, Mathematical Modeling and Computational Experiment 1 (4) (1993) 407–414.

[19] R. Chowdhury, S. Adhikari, High dimensional model representation for stochastic finite element analysis, Applied Mathematical Modelling 34 (12) (2010) 3917–3932.

[20] S. Shan, G. G. Wang, Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions, Structural and Multidisciplinary Optimization 41 (2) (2010) 219–241.

[21] H. Rabitz, O. F. Alis, General foundations of high-dimensional model representations, Journal of Mathematical Chemistry 25 (1999) 197–233.

[22] H. Rabitz, O. F. Alis, J. Shorter, K. Shim, Efficient input-output model representations, Computer Physics Communications 117 (1999) 11–20.

[23] S. W. Wang, P. G. Georgopoulos, G. Y. Li, H. Rabitz, Random sampling-high dimensional model representation (RS-HDMR) with nonuniformly distributed variables: Application to an integrated multimedia/multipathway exposure and dose model for trichloroethylene, Journal of Physical Chemistry 107 (2003) 4707–4716.

[24] G. Li, J. Hu, S.-W. Wang, P. G. Georgopoulos, J. Schoendorf, H. Rabitz, Random sampling-high dimensional model representation (RS-HDMR) and orthogonality of its different order component functions, Journal of Physical Chemistry 110 (7) (2006) 2474–2485.

[25] G. Li, C. Rosenthal, H. Rabitz, High dimensional model representations, Journal of Physical Chemistry 105 (33) (2001) 7765–7777.

[26] K. Crombecq, D. Gorissen, D. Deschrijver, T. Dhaene, Novel hybrid sequential design strategy for global surrogate modeling of computer experiments, SIAM Journal on Scientific Computing 33 (4) (2011) 1948–1974.

[27] S. Shan, G. G. Wang, Metamodeling for high dimensional simulation-based design problems, Journal of Mechanical Design 132 (5) (2010) 051009–051009.

[28] J. Sacks, S. B. Schiller, W. J. Welch, Designs for computer experiments, Technometrics 31 (1) (1989) 41–47.

[29] J. Kleijnen, Kriging metamodeling in simulation: A review, European Journal of Operational Research 192 (3) (2009) 707–716.

[30] W. Shao, H. Deng, Y. Ma, Z. Wei, Extended Gaussian kriging for computer experiments in engineering design, Engineering with Computers 28 (2) (2012) 161–178.

[31] M. L. Stein, Interpolation of Spatial Data: Some Theory for Kriging, Springer, New York, 1999.

[32] Z.-H. Han, S. Grtz, R. Zimmermann, Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function, Aerospace Science and Technology 25 (1) (2013) 177 – 189.

[33] R. Zimmermann, On the maximum likelihood training of gradient-enhanced spatial gaussian processes, SIAM Journal on Scientific Computing 35 (6) (2013) A2554–A2574.

[34] W. Näther, J. Šimák, Effective observation of random processes using derivatives, Metrika Springer-Verlag 58 (2003) 71–84.

[35] C. E. Rasmussen, C. K. I. Williams, Gaussian processes for machine learning, The MIT Press, Cambridge, MA, USA, 2006.

[36] J. Šimák, On experimental designs for derivative random fields, Ph.D. thesis, TU Bergakademie Freiberg, Freiberg, Germany (July 2002).

[37] S. Ulaganathan, I. Couckuyt, F. Ferranti, E. Laermans, T. Dhaene, Performance study of multi-fidelity gradient enhanced kriging, Structural and Multidisciplinary Optimization DOI: 10.1007/s00158-014-1192-x.

[38] S. Ulaganathan, I. Couckuyt, T. Dhaene, J. Degroote, E. Laermans, Performance study of gradient enhanced kriging, Engineering with computers (10.1007/s00366-015-0397-y) (2015) 1–20doi:10.1007/s00366-015-0397-y.

[39] Y. Lin, An efficient robust concept exploration method and sequential exploratory experimental design, Ph.D. thesis, Mechanical Engineering, Georgia Institute of Technology, Atlanda (2004).
URL http://www.worldcatlibraries.org/wcpa/top3mset/57467815

[40] R. Jin, W. Chen, A. Sudjianto, An efficient algorithm for constructing optimal design of computer experiments, Journal of Statistical Planning and Inference 134 (1) (2005) 268–287.

[41] R. Jin, W. Chen, A. Sudjianto, On sequential sampling for global metamodeling in engineering design, in: ASME 2002 Design Engineering Technical Conferences and Computer and Information in Engineering Conference, Montreal, Canada, 2002.

[42] M. Sasena, M. Parkinson, P. Goovaerts, P. Papalambros, M. Reed, Adaptive experimental design applied to an ergonomics testing procedure, in: Proceedings of the ASME Design Engineering Technical Conferences, paper DETC2002/DAC-34091, Montreal, Canada, 2002.

[43] D. Gorissen, I. Couckuyt, P. Demeester, T. Dhaene, K. Crombecq, A surrogate modeling and adaptive sampling toolbox for computer based design, Journal of Machine Learning Research 11 (2010) 2051–2055.

[44] K. Crombecq, E. Laermans, T. Dhaene, Efficient space-filling and non-collapsing sequential design strategies for simulation-based modeling, European Journal of Operational Research 214 (3) (2011) 683–696.

[45] E. P. Adorio, Mvfmultivariate test functions library in c for unconstrained global optimization, Available from http://www.geocities.ws/eadorio/mvf.pdf (2005).

[46] J. Degroote, M. Hojjat, E. Stavropoulou, R. Wüchner, K.-U. Bletzinger, Partitioned solution of an unsteady adjoint for strongly coupled fluid-structure interactions and application to parameter identification of a one-dimensional problem, Structural and Multidisciplinary Optimization 47 (1) (2013) 77–94.

[47] J. Degroote, K.-J. Bathe, J. Vierendeels, Performance of a new partitioned procedure versus a monolithic procedure in fluid-structure interaction, Computers & Structures 87 (11–12) (2009) 793–801.

[48] A. Griewank, Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, no. 19 in Frontiers in Appl. Math., SIAM, Philadelphia, PA, 2000.