A Unified Framework for Training Point Selection and Error Estimation for Surrogate Models

Komahan Boopathy *
University of Dayton, Ohio, 45469, USA

Markus P. Rumpfkeil †
University of Dayton, Ohio, 45469, USA

A unified framework for surrogate model training point selection and error estimation is proposed. Building auxiliary local surrogate models over sub-domains of the global surrogate model forms the basis of the proposed framework. A discrepancy function, defined as the absolute difference between response predictions from local and global surrogate models for randomly chosen test candidates, drives the framework – thereby not requiring any additional exact function evaluations. The benefits of this new approach are demonstrated with analytical test functions and the construction of a two-dimensional aerodynamic database. The results show that the proposed training point selection approach improves the convergence monotonicity and produces more accurate surrogate models compared to random and quasi-random training point selection strategies. The introduced root mean square discrepancy (RMSD) and maximum absolute discrepancy (MAD) exhibit close agreement with the actual root mean square error (RMSE) and maximum absolute error (MAE) respectively, and is therefore proposed as a measure for the approximation accuracy of surrogate models in applications of practical interest. Multivariate interpolation and regression is employed to build local surrogates, whereas kriging and polynomial chaos expansions serve as global surrogate models in demonstrating the applicability of the proposed framework.

*Master's Candidate, Dept. of Mechanical and Aerospace Eng., KomahanBoopathy@gmail.com, Student Member AIAA

†Assistant Professor, Dept. of Mechanical and Aerospace Eng., Markus.Rumpfkeil@udayton.edu, Senior Member AIAA

Nomenclature

C_D	Coefficient of drag
C_L	Coefficient of lift
f	Exact function
$\widehat{f}$	Approximated function
$\boldsymbol{\nabla}\widehat{f}$	Approximated gradients
$\boldsymbol{\nabla}^2\widehat{f}$	Approximated Hessian
M	Number of input parameters
M_∞	Mach number
N	Number of training points for global surrogate model
N_{init}	Number of initial training points
N_{cyc}	Number of training points added per cycle
N_t	Number of nodes in Cartesian mesh
p	Order of expansion
T	Number of terms in the expansion
$\boldsymbol{u}$	Polynomial chaos coefficients
$\boldsymbol{x}$	Training point location
$\boldsymbol{x}^*$	Closest training point to a test candidate
α	Angle of attack
δ	Discrepancy function at a given location
ϵ	Actual error at a given location
$\boldsymbol{\xi}$	Test candidate location
ρ	Distance between the test candidate and closest training point
ψ	Orthogonal polynomial
Subscripts	
$(.)_{global}$	Corresponds to global surrogate
$(.)_i$	i-th variable of multidimensional variable (.)
$(.)_{local}$	Corresponds to local surrogate
Superscripts	
$(.)^{(j)}$	j-th training point or test candidate
$\bar{(.)}$	Mean of (.)
$\widehat{(.)}$	Surrogate approximated value of (.)

I. Introduction and Motivation

Non-availability of theoretical solutions to many engineering problems as well as the high expenses associated with performing laboratory testing have caused an increased reliance upon numerical simulations. Owing to an increasing availability of computational resources and sophisticated algorithms, numerical methods are extensively used in engineering research and development to solve many real-life problems. In spite of many advances in computer hardware and parallel computing, a striking imbalance still exists between the requirements and availability of computational power. For example, the number of required mesh points for direct numerical simulations (DNS) scales with Reynolds number as $Re^{9/4}$, but current state-of-the-art computing can only support on the order of 10^9 - 10^{10} mesh points and is thus limited to Reynolds numbers of about $10,000$ or so. Other high-fidelity physics-based simulations, such as large eddy simulations (LES) and unsteady Reynolds-averaged Navier-Stokes (RANS) computations, can also require significant amount of computational resources and time. In order to reduce the computational burden from multiple simulations, e.g. for optimization or uncertainty quantification, surrogate models (also called metamodels or response surfaces) have been employed by the scientific community. Surrogate models provide an approximate but inexpensive to evaluate representation of the output quantity of interest as a function of input variables, and are thus bound to have approximation errors. The surrogate models considered here are sample-based and not physics-based. Major research topics that are extensively pursued related to surrogate modeling are discussed in the following paragraphs (Sections I.A to I.C).

I.A. Choice of training points

The accuracy of a given surrogate model is influenced primarily by the non-linearity of the function to be modeled and by the choice of training point locations. Training point selection is typically done using design of experiments (DoE) techniques (e.g. uniform design [1]). Many other methods, which have been originally developed to approximate multi-dimensional integrals, are also being used for training data selection for surrogate models. These include Monte Carlo (MC) [2], latin hypercube (LHS) [3], quadrature nodes [4, 5], and low-discrepancy sequences [6]. However, these strategies tend to suffer from deficiencies, such as the exponential growth in the number of required points with dimensionality (e.g. quadrature nodes), missing important regions by chance (e.g. LHS, MC), poor and correlated distribution of training points in higher dimensions (e.g. Sobol [7] and Halton [6] sequences), etc. Apart from having been developed for a different purpose, all of these strategies are domain-based, *i.e.,* they neither take into account function values nor their non-linearities. Thus, plenty of research has been conducted to address the aforementioned problems and to

develop adaptive strategies which consider response values and similar criteria such as the expected improvement and mean squared error [8–10].

The appropriate choice of training points is an important open research question as pointed out recently by Roderick *et al.* [11, 12] and Cheng *et al.* [13]. Appendix A contains a brief review of some popular methods for training point selection. For a comprehensive review of these topics the reader is referred to Keane and Nair [14], Arora [15], and Forrester *et al.* [16].

I.B. Surrogate model approximation error

Theoretically, the approximation error associated with surrogate models can be quantified by comparison with the exact function values, by means of computable quantities such as the root mean square error (RMSE) and L_∞-norm. However, in real-life applications, it is computationally impractical to calculate any of these quantities since they require too many exact function evaluations. Without having a good measure for the accuracy of surrogate models, the validity of application results involving surrogate models becomes highly questionable. In this article a kriging model which finds the best linear unbiased predictor and minimizes the expected mean squared error (MSE) is employed. One could simply use this MSE to estimate the approximation error, however, in practice MSE is really much more of a measure of how well the training points fill the domain uniformly than an actual model approximation error. Recently, Mehmani *et al.* [9] proposed a method for the regional error estimation of surrogates (REES) (see Appendix A.B.3). Other error measurement techniques include [17]: split sampling, cross validation, bootstrapping and Akaike's information criterion (AIC) [18]. These error measures either require additional exact function evaluations or provide limited information regarding the accuracy of surrogates. A brief note on the widely used cross validation method is provided in Appendix B.

I.C. Curse of dimensionality

The exponential rise in required training data and computational cost as the number of input dimensions to a surrogate model increases is referred to as the "curse of dimensionality". To address this problem, the introduction of higher-order derivative information (gradients and Hessian) within surrogate models as additional training data has attracted a lot of attention. For example, many gradient-enhanced surrogate models have been developed and have shown very beneficial results [19–22]. This is mainly due to the availability of computationally efficient and accurate gradient evaluation methods such as the adjoint formulation [23, 24]. The easiest to implement and popular method for obtaining derivatives is the finite-difference method which is highly sensitive to the choice of step-size as well as

computationally expensive with increasing number of input dimensions, M. The adjoint method, on the other hand, provides a more efficient means of calculating the derivative and features better accuracy. When targeting a single output objective, the effort needed to compute the full gradient is comparable to the effort needed to compute the function itself. The main appeal is to use function values and their derivatives for the construction of surrogate models in higher dimensions, since one can obtain $M + 1$ pieces of information for the constant cost of roughly two function evaluations using adjoint techniques.

The ability to compute second-order sensitivity derivatives is also highly desirable for many science and engineering simulation problems [25–28]. For example, the availability of Hessian information allows the use of much stronger Newton optimization strategies which hold the potential for greatly reducing the expense of solving difficult optimization problems. An efficient Hessian evaluation method has been developed by Rumpfkeil and Mavriplis [29, 30] and using the same logic as above, it is also very promising to utilize Hessian information within surrogate models in addition to gradients. The Hessian provides $M \cdot (M + 1)/2$ pieces of information for roughly the cost of M function evaluations, since, in general, the most efficient full Hessian constructions require the solution of M forward linear problems (one corresponding to each input variable) [25, 27, 30]. In summary, when using gradient and Hessian information, one would compute the output function far fewer times to obtain an accurate surrogate model thereby reducing the "curse of dimensionality".

Another promising avenue is the use of variable-fidelity surrogate modeling [19, 31–36], where one tries to reduce the computational burden by employing a larger amount of low-fidelity data (e.g. Euler evaluations) in conjunction with a smaller amount of high-fidelity data (e.g. Navier-Stokes evaluations). This indeed offers a great potential for savings since, for example, Euler evaluations are 50–100 times cheaper to obtain compared to equivalent RANS evaluations [37].

Scope and outline of the article

This work aims to provide a better selection strategy for training point locations that are most viable to improve the accuracy of the surrogate model. It also aims to provide a surrogate model approximation error which is computed in conjunction with the training point selection process providing a computational advantage. The proposed framework does not warrant additional exact function evaluations in the process. To show the applicability of this framework to any surrogate model two different surrogate models are employed, namely kriging [38] and polynomial chaos expansions [4] (PCE). In the process, the polynomial chaos is newly enhanced with Hessian information, while a gradient- and Hessian-enhanced kriging surrogate model is adopted from the literature [22,30]. The process of training point selection will be discussed with two different scenarios in mind, as pointed out by Roderick *et al.* [11]:

(i) selection in the absence of derivative information (function values only) and (ii) selection in the presence of derivative information (function, gradient and Hessian values). The proposed framework is studied in the context of building globally accurate surrogate models, whereas the investigation pertaining to its application for surrogate-based optimization is deferred to future research. The "curse of dimensionality" is not directly addressed here, although an appropriate choice of training points and the inclusion of derivative information can help to alleviate some of the computational burden.

The remainder of the article is organized as follows. Section II reviews kriging and polynomial chaos surrogate modeling approaches. A brief note on multivariate interpolation and regression used to construct auxiliary local surrogate models is also included. Section III details the proposed dynamic training point selection and error estimation algorithm. Section IV presents results for analytical test functions and Section V demonstrates the performance of the framework for the construction of aerodynamic databases of drag and lift coefficients. Section VI concludes this article.

II. Surrogate Models

In this work, kriging [38] and polynomial chaos expansions (PCE) [11, 12] are employed as global surrogate models, whereas multivariate interpolation and regression (MIR) [39, 40] is used to build local surrogate models. In the following paragraphs, brief overviews of kriging and MIR are provided, whereas PCE is discussed in more detail to explain the newly-developed Hessian enhancement.

II.A. Kriging surrogate models

II.A.1. Original kriging surrogate model

The kriging surrogate model was originally developed in the field of geostatistics by the South-African mining engineer Danie G. Krige [41]. Kriging was introduced in engineering design following the work of Sacks *et al.* [42] and has found a lot of aerospace applications [20, 21, 43, 44]. Kriging predicts a function value by using stochastic processes and has the flexibility to represent multi-modal and highly non-linear functions. The formulation based on the conventional "ordinary kriging" model (used here for simplicity of description) is given by [15],

$$\widehat{f}(\boldsymbol{x}) = \mu + Z(\boldsymbol{x}), \tag{1}$$

where $\widehat{f}$ is the approximated function value, μ is the constant mean model, and $Z(\boldsymbol{x})$ represents a local variation from the mean behavior using a Gaussian process. Kriging uses spatial correlation between data points within its correlation matrix.

II.A.2. Enhanced kriging surrogate models

Gradient- and Hessian-enhanced direct as well as indirect co-kriging models have previously been developed and have shown very beneficial results [19,22,38,45]. In the direct co-kriging approach, the correlations between gradient and Hessian values are also included in the correlation matrix as opposed to the original kriging formulation where only the correlations between function values are considered. The formulation of the indirect co-kriging model is the same as that of the original kriging model but additional training points are constructed around a real training point by using Taylor series expansions. The direct approach is preferable since its correlation matrix has a better condition number and due to the lack of tunable parameters. Apart from these derivative enhancements, the kriging model has also been improved through an adaptive training point selection by the authors [46, 47], where Dutch intrapolation [48, 49] and MIR [39, 40] are used as local surrogate models that guide the selection of training points for the global kriging surrogate model, forming the basis of this work which is also extended to PCE in this article. For a more detailed mathematical background on kriging the reader is referred to Sacks *et al.* [42] and other articles in the literature [22, 38, 45].

II.B. Multivariate interpolation and regression

Multivariate interpolation and regression (MIR) [39, 40] is a surrogate model where each data point is represented as a Taylor series expansion, and higher-order derivatives in the Taylor series are treated as random variables. Mathematically, the exact function, f, in an M-dimensional domain is approximated as [39, 40],

$$\widehat{f}(\boldsymbol{x}) = \sum_{i=1}^{N_v} a_{vi}(\boldsymbol{x})\widetilde{f}(\boldsymbol{x}_{vi}) + \sum_{i=1}^{N_g} \boldsymbol{a}_{gi}(\boldsymbol{x})\boldsymbol{\nabla}\widetilde{f}(\boldsymbol{x}_{gi}), \tag{2}$$

where N_v is the number of function data points and N_g is the number of gradient data points (if used). a_{vi} and $\boldsymbol{a}_{gi}$ are the basis functions of the resulting polynomial. $\widetilde{f}$ and $\boldsymbol{\nabla}\widetilde{f}$ are the function, f, and gradient values, $\boldsymbol{\nabla}f$, added with their corresponding measurement errors σ_{vi} and $\boldsymbol{\sigma}_{gi}$ (if any). The scheme produces an interpolatory response surface when the data points are exact, or a regression model if some measurement errors are associated with the data points.

The main user-specified parameter is the Taylor order, n. The choice of an optimum Taylor order is difficult as it depends on the function to be modeled as well as its dimensionality. One would expect to improve the accuracy of the model by using a higher Taylor order, but round-off errors, originating in the solution of the underlying least-squares problem for determining the coefficients, propagate to the approximated function value via Eq. (2). Therefore, a higher Taylor order does not always guarantee an improved approximation and it is also

difficult to know the optimum Taylor order a priori. The reader is referred to Wang *et al.* [39, 40] for more details about the model.

II.C. Polynomial chaos expansions

Polynomial chaos expansions (PCE) refer to the polynomial representation of uncertainty (chaos), where stochastic quantities are represented as spectral expansions of orthogonal polynomials and can be classified into *stochastic Galerkin* (intrusive) and *stochastic collocation* (non-intrusive) methods. The current work focuses on non-intrusive methods and the reader is referred to the literature for more details on intrusive methods [4, 5, 50, 51].

A non-intrusive polynomial chaos regression procedure developed in Roderick *et al.* [11,12] is employed in this work. A truncated spectral expansion of an output function, f, dependent on a multidimensional random variable $\boldsymbol{x}$ takes the form,

$$\widehat{f}(\boldsymbol{x}) = \sum_{k=0}^{P} u_k \psi_k(\boldsymbol{x}), \tag{3}$$

where $\psi(\boldsymbol{x})$ denotes the selected basis function. This representation or expansion is called the PCE surrogate model in this work, created by using a polynomial regression and is closely related [11, 12] to polynomial chaos methods such as interpolation [52], collocation [53], and surface response [54, 55]. The random variable $\boldsymbol{x}$ is assumed to be uniformly distributed and a Legendre basis function is employed in this work since in the absence of any additional information, uniform prior distribution is among the most reasonable assumptions [11, 56]. A data dependent basis selection has recently been developed in Roderick *et al.* [12] but not is used here for simplicity.

The total number of terms in the expansion is given by,

$$T = P + 1 = \frac{(M+p)!}{M!p!}, \tag{4}$$

where M is the number of input variables and p is the desired polynomial order. Multi-dimensional orthogonal polynomial bases can be easily obtained through tensorization of corresponding one-dimensional polynomial bases using a multi-index notation [4, 5, 50, 51],

$$\psi_k(\boldsymbol{x}) = \boldsymbol{\psi}_k(x_1, x_2, \ldots, x_M) = \prod_{i=1}^{M} \psi_{\alpha_i^k}(x_i), \tag{5}$$

where the multi-index α_i^k denotes the order of the one-dimensional polynomial.

II.C.1. Polynomial Regression Procedure

The polynomial regression procedure based on an available set of exact function values is discussed below.

1. **Setting up the linear system:** The expansion in Eq. (3) can be written in matrix-

form as follows.

$$\begin{bmatrix} \psi_0(\boldsymbol{x}^{(1)}) & \psi_1(\boldsymbol{x}^{(1)}) & \cdots & \psi_P(\boldsymbol{x}^{(1)}) \\ \psi_0(\boldsymbol{x}^{(2)}) & \psi_1(\boldsymbol{x}^{(2)}) & \cdots & \psi_P(\boldsymbol{x}^{(2)}) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_0(\boldsymbol{x}^{(N)}) & \psi_1(\boldsymbol{x}^{(N)}) & \cdots & \psi_P(\boldsymbol{x}^{(N)}) \end{bmatrix} \begin{Bmatrix} u_0 \\ u_1 \\ \vdots \\ u_P \end{Bmatrix} = \begin{Bmatrix} f(\boldsymbol{x}^{(1)}) \\ f(\boldsymbol{x}^{(2)}) \\ \vdots \\ f(\boldsymbol{x}^{(N)}) \end{Bmatrix} \tag{6}$$

Here, polynomial fitting conditions are enforced at points $\boldsymbol{x}^{(j)}$, $j = 1, 2, \ldots, N$, resulting in a linear system with N equations and $T = P + 1$ unknown coefficients, $\boldsymbol{u}$.

2. **Solving the linear system:** An interpolation or regression model is formed depending upon the availability of information. If just enough data is present to solve for T coefficients (*i.e.,* $N = T$), an interpolatory response surface is obtained. To prevent accidental rank deficiency, an oversampling factor of 2 is recommended in the literature [51] and is adopted in this work as well. If oversampling (*i.e.,* $N > T$) is enforced the linear system can only be solved in a least-squares sense and the resulting response surface is a regression model.

3. **Evaluating the response surface:** Once the coefficients, $\boldsymbol{u}$, are solved for, the PCE response surface defined by Eq. (3) is successfully obtained. It can now be used to obtain the approximated function value at any given location, $\boldsymbol{x}$. In addition, gradient and Hessian approximations from the PCE surrogate model at any location are also readily obtained by differentiating Eq. (3). Mathematically, this is represented by,

$$\boldsymbol{\nabla} \widehat{f} = \sum_{k=0}^{P} u_k \frac{\partial \psi_k(\boldsymbol{x})}{\partial \boldsymbol{x}} \tag{7}$$

and

$$\boldsymbol{\nabla}^2 \widehat{f} = \sum_{k=0}^{P} u_k \frac{\partial^2 \psi_k(\boldsymbol{x})}{\partial \boldsymbol{x}^2}, \tag{8}$$

where $\frac{\partial \psi_k(\boldsymbol{x})}{\partial \boldsymbol{x}}$ and $\frac{\partial^2 \psi_k(\boldsymbol{x})}{\partial \boldsymbol{x}^2}$ again involve the multi-index notation given by Eq. (5).

II.C.2. Enhancing PCE with derivative information

Polynomial chaos enhanced with gradient information is shown to produce promising results to reduce the curse of dimensionality in Roderick *et al.* [11, 12]. These authors have named their approach polynomial regression with derivative information (PRD), which is extended here to incorporate Hessian information. When gradient and Hessian information are included, the resulting linear system is generally over-determined and can only be solved in a least-squares sense resulting in a regression model.

Augmenting gradient information:

When the linear system in Eq. (6) is augmented with gradient information, each row of the matrix in Eq. (6) gives rise to M additional rows, where M is the number of components in $\boldsymbol{x}$. Thus, if there are N training points the size of the matrix becomes $N' \times T$, where $N' = N \cdot (1 + M)$.

$$\left[\begin{array}{c} \begin{pmatrix} \psi_0(\boldsymbol{x}^{(1)}) & \psi_1(\boldsymbol{x}^{(1)}) & \cdots & \psi_P(\boldsymbol{x}^{(1)}) \\ \frac{\partial\psi_0(\boldsymbol{x}^{(1)})}{\partial x_1} & \frac{\partial\psi_1(\boldsymbol{x}^{(1)})}{\partial x_1} & \cdots & \frac{\partial\psi_P(\boldsymbol{x}^{(1)})}{\partial x_1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial\psi_0(\boldsymbol{x}^{(1)})}{\partial x_M} & \frac{\partial\psi_1(\boldsymbol{x}^{(1)})}{\partial x_M} & \cdots & \frac{\partial\psi_P(\boldsymbol{x}^{(1)})}{\partial x_M} \end{pmatrix} \\ \vdots \end{array}\right] \begin{Bmatrix} u_0 \\ u_1 \\ \vdots \\ u_P \end{Bmatrix} = \left\{\begin{array}{c} \begin{pmatrix} f(\boldsymbol{x}^{(1)}) \\ \frac{\partial f(\boldsymbol{x}^{(1)})}{\partial x_1} \\ \vdots \\ \frac{\partial f(\boldsymbol{x}^{(1)})}{\partial x_M} \end{pmatrix} \\ \vdots \end{array}\right\} \tag{9}$$

The resulting surrogate model matches the function values and their corresponding first-derivatives in a least-squares sense.

Augmenting Hessian information:

Similarly, the addition of Hessian information increases the size of the matrix to $N' \times T$, where $N' = N \cdot (1 + M + \frac{M(M+1)}{2})$.

$$\left[\begin{array}{c} \begin{pmatrix} \psi_0(\boldsymbol{x}^{(1)}) & \psi_1(\boldsymbol{x}^{(1)}) & \cdots & \psi_P(\boldsymbol{x}^{(1)}) \\ \frac{\partial\psi_0(\boldsymbol{x}^{(1)})}{\partial x_1} & \frac{\partial\psi_1(\boldsymbol{x}^{(1)})}{\partial x_1} & \cdots & \frac{\partial\psi_P(\boldsymbol{x}^{(1)})}{\partial x_1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial\psi_0(\boldsymbol{x}^{(1)})}{\partial x_M} & \frac{\partial\psi_1(\boldsymbol{x}^{(1)})}{\partial x_M} & \cdots & \frac{\partial\psi_P(\boldsymbol{x}^{(1)})}{\partial x_M} \\ \frac{\partial^2\psi_0(\boldsymbol{x}^{(1)})}{\partial x_1^2} & \frac{\partial^2\psi_1(\boldsymbol{x}^{(1)})}{\partial x_1^2} & \cdots & \frac{\partial^2\psi_P(\boldsymbol{x}^{(1)})}{\partial x_1^2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2\psi_0(\boldsymbol{x}^{(1)})}{\partial x_1\partial x_M} & \frac{\partial^2\psi_1(\boldsymbol{x}^{(1)})}{\partial x_1\partial x_M} & \cdots & \frac{\partial^2\psi_P(\boldsymbol{x}^{(1)})}{\partial x_1\partial x_M} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2\psi_0(\boldsymbol{x}^{(1)})}{\partial x_M^2} & \frac{\partial^2\psi_1(\boldsymbol{x}^{(1)})}{\partial x_M^2} & \cdots & \frac{\partial^2\psi_P(\boldsymbol{x}^{(1)})}{\partial x_M^2} \end{pmatrix} \\ \vdots \end{array}\right] \begin{Bmatrix} u_0 \\ u_1 \\ \vdots \\ u_P \end{Bmatrix} = \left\{\begin{array}{c} \begin{pmatrix} f(\boldsymbol{x}^{(1)}) \\ \frac{\partial f(\boldsymbol{x}^{(1)})}{\partial x_1} \\ \vdots \\ \frac{\partial f(\boldsymbol{x}^{(1)})}{\partial x_M} \\ \frac{\partial^2 f(\boldsymbol{x}^{(1)})}{\partial x_1^2} \\ \vdots \\ \frac{\partial^2 f(\boldsymbol{x}^{(1)})}{\partial x_1\partial x_M} \\ \vdots \\ \frac{\partial^2 f(\boldsymbol{x}^{(1)})}{\partial x_M^2} \end{pmatrix} \\ \vdots \end{array}\right\} \tag{10}$$

Again, the resulting surrogate model is now required to match the function values and their corresponding first- and second-derivatives in a least-squares sense.

This discussed approach is non-intrusive as only the right hand side of the equation needs function, gradient and Hessian values and one can use a black-box approach to obtain them.

III. Proposed Framework

A detailed description of the proposed dynamic training point selection and error estimation framework is provided in this section. The global surrogate model refers to the main surrogate model that is built over the entire domain of interest whereas the local surrogate model refers to the ones that are built over sub-domains of the main surrogate.

III.A. Discussion of steps involved

Figure 1 shows a schematic diagram of the algorithm. The steps involved in the process are

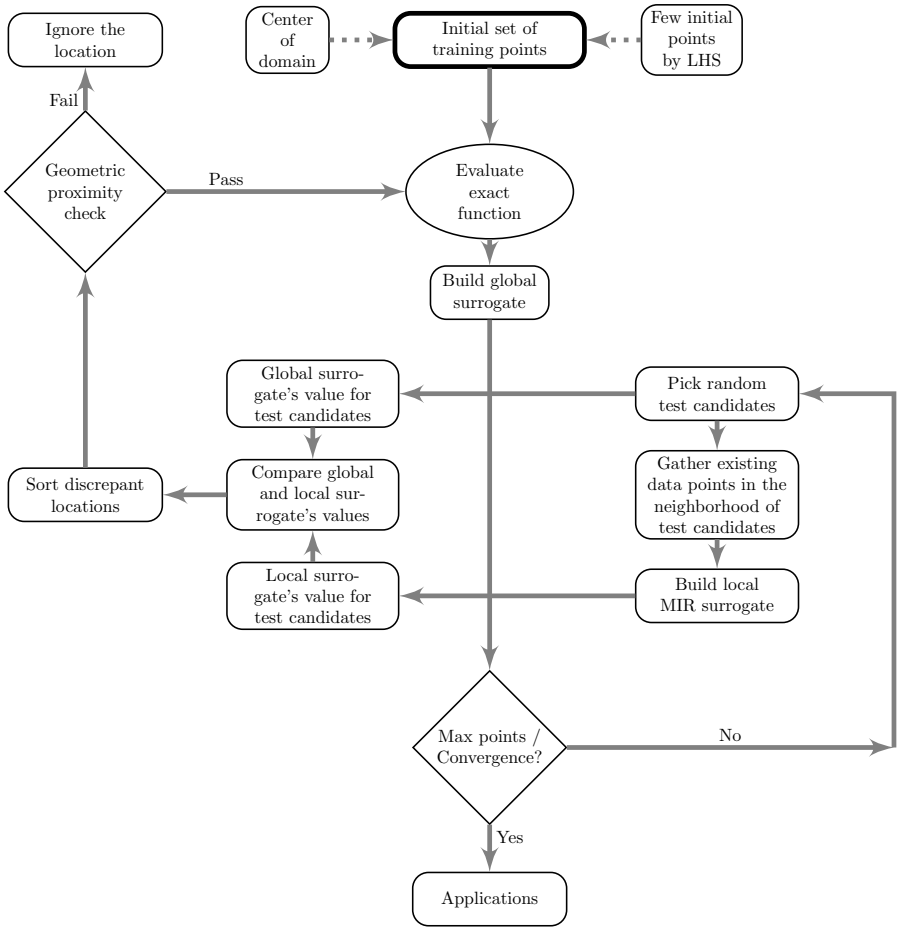

Figure 1. A schematic diagram of the proposed framework using a local surrogate (MIR)

as follows.

1. Initialization

The exact function (also gradients and Hessian, if desired) is evaluated at the center of the domain and a few other points picked by LHS, totaling N_{init} training points. The choice

of N_{init} is left to the user; however, it is recommended to start with a reasonably small number depending on the dimensionality and size of the domain and to let the framework pick additional locations. In this work, for two-dimensional test cases, N_{init} has been set to five for kriging and twelve for PCE, where the latter includes an oversampling factor of two and the required number of data points to build a second-order accurate PCE (setting $p = 2$ and $M = 2$ in Eq. 4). For five-dimensional test cases, N_{init} is set to fifty for kriging and forty-two for PCE.

2. Evaluation of surrogate models

Numerous test candidates, $\boldsymbol{\xi}^{(j)},\ j = 1, \ldots, N_{test}$, are picked throughout the entire domain via LHS and the following is done with the two surrogates:

(a) **Global surrogate:** The global surrogate model, which is built from all available training data, is simply evaluated at all these test candidate locations yielding $\widehat{f}^{(j)}_{global}(\boldsymbol{\xi}),\ j = 1, \ldots, N_{test}$ and the values are stored.

(b) **Local surrogate:** During the first iteration (selection cycle), only one local surrogate is built using all available N_{init} training data points (making it a "second global surrogate"). The local surrogate model is also simply evaluated yielding $\widehat{f}^{(j)}_{local}(\boldsymbol{\xi}),\ j = 1, \ldots, N_{test}$. During subsequent selection cycles, local surrogate models are rebuilt using N_{local} closest existing training points for each test candidate $\boldsymbol{\xi}^{(j)}$ to evaluate $\widehat{f}^{(j)}_{local}(\boldsymbol{\xi})$.

In practical applications, it is intractable to use the exact function to calculate the root mean square error of the global (main) surrogate model. A discrepancy function $\delta(\boldsymbol{\xi})$ is proposed as an approximation to the actual error $\epsilon(\boldsymbol{\xi})$ between the exact function and surrogate model at any given location $\boldsymbol{\xi}$ and is defined as $\delta(\boldsymbol{\xi}) = |\widehat{f}_{global}(\boldsymbol{\xi}) - \widehat{f}_{local}(\boldsymbol{\xi})|$. The underlying assumption is that the local surrogate models provide a more accurate representation of *their corresponding sub-domain* than the global surrogate model since piecewise approximations are generally considered to be more accurate for highly non-linear and non-smooth functions [57, 58]. Though the local surrogate models can be inaccurate in some cases, for example, due to the scarcity of training data, extrapolations in unbounded space, improper tuning of the parameters for the local models, etc., they can still serve as reference models for the global surrogate model. One could also construct multiple local approximations with yet another local surrogate model (e.g. radial basis functions [59]) in addition to the MIR local surrogate for added fidelity, but this avenue is not explored in this work.

A *root mean square discrepancy* (RMSD) for all N_{test} test candidates can be defined as,

$$\text{RMSD} = \sqrt{\frac{1}{N_{test}} \sum_{j=1}^{N_{test}} (\widehat{f}^{(j)}_{global} - \widehat{f}^{(j)}_{local})^2} = \sqrt{\frac{1}{N_{test}} \sum_{j=1}^{N_{test}} (\delta^{(j)})^2}, \tag{11}$$

and can be used as an approximation to the actual root mean square error (RMSE or L_2-norm) of the surrogate model. Similarly, a *maximum absolute discrepancy* (MAD) defined as,

$$\text{MAD} = \mathbf{max}\{|f^{(j)}_{global} - \widehat{f}^{(j)}_{local}|\} \qquad j = 1, \ldots, N_{test}, \tag{12}$$

measures the worst discrepancy between the local and global surrogate models and can be used to emulate the actual maximum absolute error (MAE or L_∞-norm).

Remark 1: The data used for building the local surrogate models is a subset of already available data, $f^{(i)}(\boldsymbol{x}), i = 1, 2 \ldots, N$. Thus, no additional exact function evaluations are needed for constructing the local surrogates.

Remark 2: Although the number of training points used to build a local surrogate, N_{local}, can be as high as the number of points used to build the global surrogate, N, it is recommended to only use an N_{local} that is sufficient to produce a reasonably accurate local surrogate. Using more points to increase the accuracy of the local surrogate model will also increase the computational expenses in building and evaluating it. Thus, the choice of N_{local} is a major heuristic factor in this work. As a rule of thumb, studies similar to the ones shown in Section III.C can be used to assess the performance of the local surrogate, or one can simply specify a certain percentage of the global training data to be used for the local surrogate models. In this work, 5–50 closest existing data points (depending on the dimension of the problem) are used to build the local surrogate models.

Remark 3: The user should choose an appropriate number of test candidates, N_{test}, depending on the dimensionality of the design space. For example, one can adopt heuristics used for Monte Carlo [7] or inexpensive Monte Carlo simulations [35] (IMCS) from the literature. A larger N_{test} facilitates a much wider exploration of the domain but comes with a higher computational cost. In this work 1,000–25,000 test candidates are used. Fortunately, building and evaluating the local surrogates can be executed in parallel. As each *test candidate* is a *potential training point location* during the next selection cycle, these terms will be used synonymously.

3. Selection of training points

Selection is the process that determines whether or not a test candidate becomes a training point location at which the exact function (gradients and Hessian) is to be evaluated. This includes the following two steps:

(a) **Sorting the discrepancies:** Locations are prioritized based on their calculated discrepancies. A sorted discrepancy function vector contains values in the order of decreasing discrepancy and is represented as $\boldsymbol{\delta}_{sort}$.

(b) **Proximity check:** The distance, $\rho^{(j)}$, between each test candidate, $\boldsymbol{\xi}^{(j)}$, and its nearest existing training point, $\boldsymbol{x}^{*(j)}$, is calculated. Mathematically,

$$\rho^{(j)}(\boldsymbol{\xi}^{(j)}, \boldsymbol{x}^{*(j)}) = ||\boldsymbol{\xi}^{(j)} - \boldsymbol{x}^{*(j)}||_2, \quad j = 1, \ldots, N_{test}. \tag{13}$$

The mean value of all these distances measure the average proximity of a potential training point (belonging to the set of test candidates) to an existing training point and is written as,

$$\bar{\rho} = \frac{1}{N_{test}} \sum_{j=1}^{N_{test}} \rho^{(j)}. \tag{14}$$

Now, each test candidate from the ordered set, $\boldsymbol{\delta}_{sort}$, starting with the one with the largest discrepancy is checked for proximity to already existing training points by calculating whether $\rho^{(j)} > c\bar{\rho}$, where c is a *control parameter* that can be specified by the user to relax or strongly emphasize the constraint. In this work c is set to 1.0. If a particular test candidate passes the check, it is added to the actual set of training points and the exact function (gradient and Hessian) is evaluated. After every successful selection of a training point the distances given by Eqns. (13) and (14) are updated.

The proximity check is repeated until N_{cyc} new training points have been selected at this iteration. The newly available training data can now be used in subsequent iterations of the framework to update the surrogate models.

Remark 4: The enforcement of a geometric constraint works under the assumption that the global surrogate model is sufficiently accurate within $c\bar{\rho}$ distance from an existing training data point. In other words, the main surrogate does not warrant any additional training data within this radius, and the exact function can rather be evaluated at a more unexplored location. This approach ensures that the training points are not clustered in one particular region and are sparse in other regions of the domain. As already mentioned in Section I, each function evaluation can be computationally very expensive, especially for high-fidelity physics-based simulations, and it is essential to choose each new training point location effectively.

Remark 5: Due to the enforcement of geometric constraints the framework may not support *fast* optimizations as these constraints can prevent the placement of many training points close to the optimum. Though the framework is recommended for building globally accurate surrogate models, the geometric constraints do not severely restrict the applicability to optimizations as they are controllable.

4. Termination

Steps (2) and (3) are repeated to iteratively update the global and local surrogate models until any one of the following criteria is satisfied.

(a) **Reach desired accuracy:** One can either use the the maximum absolute discrepancy (MAD) and/or root mean square discrepancy (RMSD) to monitor the estimated accuracy of the current surrogate. An excellent agreement between MAD and MAE as well as RMSD and RMSE will be shown in Section IV.A. The process of selecting additional training data can be stopped when a desired accuracy level is reached.

(b) **Exhaust computational budget:** Usually one works with a specified computational budget when applying surrogate models to applications of practical interest. Thus, when a maximum number of exact function (gradients and Hessian) evaluations is reached the user determines to stop the training point selection process.

Remark 6: One can also use the framework for only surrogate model error estimation by skipping Step (3).

III.B. Test functions

Eqns. (15), (16) and (17) list the multidimensional exponential, Runge and Rosenbrock test functions, respectively, that are used for evaluation purposes on a hypercube $[-2, 2]^M$;

$$f_1(x_1, \ldots, x_M) = \mathrm{e}^{(x_1 + \ldots + x_M)} \tag{15}$$

$$f_2(x_1, \ldots, x_M) = \frac{1}{1 + x_1^2 + \ldots + x_M^2} \tag{16}$$

$$f_3(x_1, \ldots, x_M) = \sum_{i=1}^{M-1} \left[(1 - x_i)^2 + 100(x_{i+1} - x_i^2)^2\right] \tag{17}$$

The root mean square error (RMSE) between the exact, f, and approximated function values, $\widehat{f}$, calculated on an M-dimensional Cartesian mesh with N_t total nodes is given by,

$$RMSE = \sqrt{\frac{1}{N_t} \sum_{i=1}^{N_t} (f_i - \widehat{f}_i)^2}. \tag{18}$$

The RMSE is calculated on a Cartesian mesh with $10,201$ and $100,000$ nodes for two- and five-dimensional test cases, respectively.

III.C. Choice for local surrogate model

This section provides guidelines on choosing a good local surrogate model that can guide the framework effectively. The authors have used dutch intrapolation and MIR as local surrogate models in their recent works [46, 47].

Figure 2 compares the original kriging and MIR on two-dimensional exponential, Runge, and Rosenbrock test functions as listed in Section III.B, where all training points are selected through LHS. The general observation is that MIR approximates all test functions except Runge better than kriging. The addition of gradient information (labeled FG, continuous lines) yields better results than the approximation with function values alone (labeled F, dotted lines). The reader is referred to Wang *et al.* [39,40] for a detailed comparison of MIR with other surrogate modeling methods and higher-dimensional test functions.

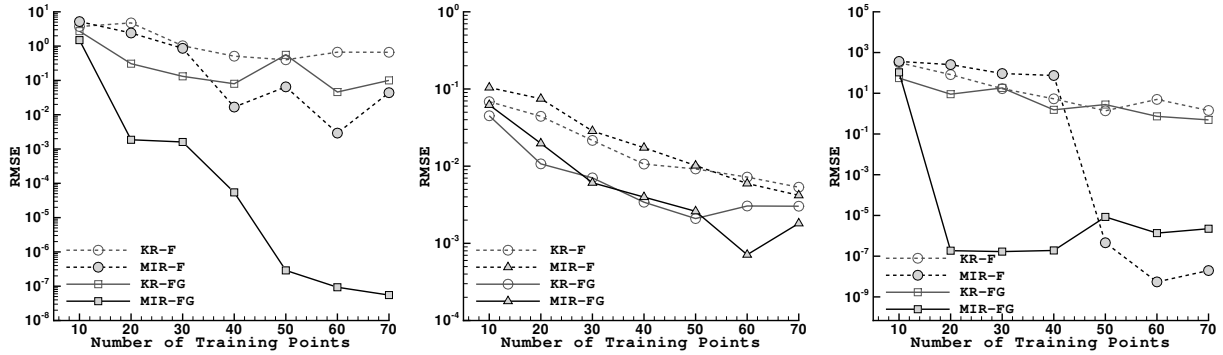

Figure 2. Plots of RMSE of the surrogate versus the number of training points using kriging and MIR for two-dimensional test functions (from left to right: exponential, Runge and Rosenbrock).

Figure 3 shows the effect of different Taylor orders on the accuracy of the MIR approximated function value, $\widehat{f}$. It can be seen that a lower Taylor order, n, such as 1 or 2 produces a less accurate approximation, whereas a higher n leads to an improved approximation. The problem with a higher Taylor order is that it increases the computational time in building the MIR surrogate significantly and there is not an easy way to choose the appropriate Taylor order a priori. However, it is recommended to use a Taylor order that is equal to the number of data points (*i.e.*, $n = N$), or $3N$ if the data points contain gradient information [39, 40].

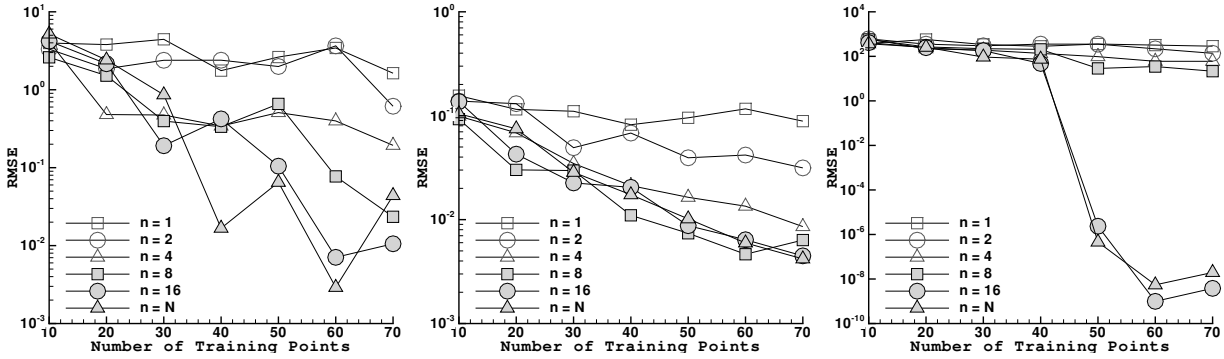

Figure 3. Plots of MIR RMSE versus the number of training points for different Taylor orders, n, on two-dimensional test functions (from left to right: exponential, Runge and Rosenbrock).

Though building the MIR surrogate becomes computationally very expensive with an increasing number of training points [39, 40], one can infer from Figure 2 that MIR is able to produce reasonably good surrogates with only a few training points. Hence, one can

construct cheaper and reasonably accurate local surrogates with MIR and use them to guide the construction of a global surrogate.

It is generally acknowledged that kriging has the capability to represent multi-modal functions and can effectively capture non-smooth functions [22]. Moreover, kriging supports the usage of both high- and low-fidelity training points *i.e.,* one can construct a variable-fidelity surrogate with kriging [19, 31–36]. Due to these reasons kriging is used as the global surrogate model, whereas MIR is used to guide the training point selection process as well as to provide a metric for the accuracy of the global surrogate.

IV. Results for Analytical Test Functions

This section details the results obtained using kriging and polynomial chaos expansions for the analytical test functions listed in Section III.B. The kriging surrogate model employs the Wendland C4 spatial correlation function [60]. In all comparative studies, the training data set is forced to include the center of the domain, while all others are either selected through LHS or dynamically. Whenever plots are shown in a group of three (triplets), the leftmost, middle and rightmost figures, correspond to the exponential (f_1), Runge (f_2), and Rosenbrock (f_3) test functions, respectively.

IV.A. Validation of proposed error estimates

Conventional methods such as cross validation [61, 62] (see Appendix B) are being used for validating the surrogate model. As discussed in Section III.A, the dynamic training point selection framework also provides an approximate error estimate for the global (main) surrogate model through the maximum absolute discrepancy (MAD) and the root mean square discrepancy (RMSD).

IV.A.1. Comparison with actual errors and cross validation

In this section, comparisons of (i) the actual maximum absolute error (MAE or L_∞-norm) and the maximum cross validation error (Max–CVE) with the proposed MAD, and (ii) the actual root mean square error (RMSE or L_2-norm) and mean cross validation error (MCVE) with the proposed RMSD, are provided. Figures 4 and 5 show results for kriging and PCE, respectively. *Leave-one-out cross validation* [61, 62] is employed in this study.

The general observation is that both proposed error measures (MAD and RMSD) feature an excellent agreement with the actual errors (MAE and RMSE) for all tested cases (exponential, Runge and Rosenbrock functions) and surrogate modeling approaches (kriging and PCE), with only a few occasional minor differences. This presented behavior shows great

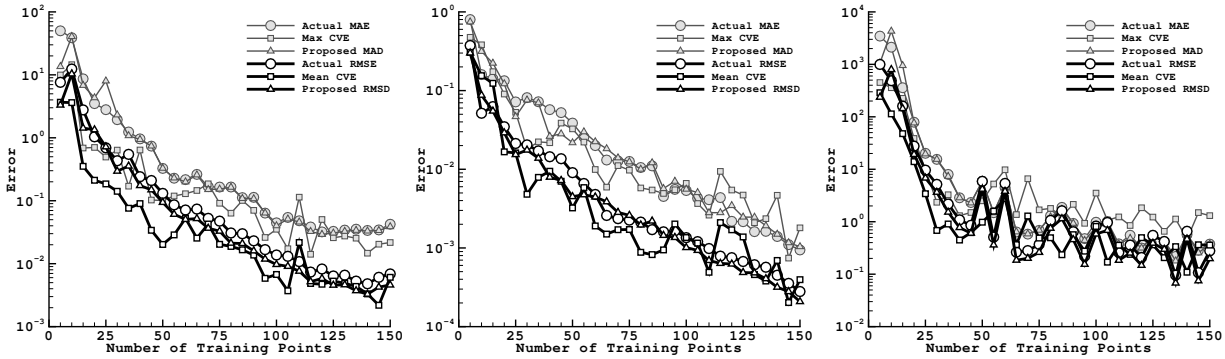

Figure 4. A comparison of the proposed error measures with actual errors and leave-one-out cross validation on two-dimensional test functions using kriging.

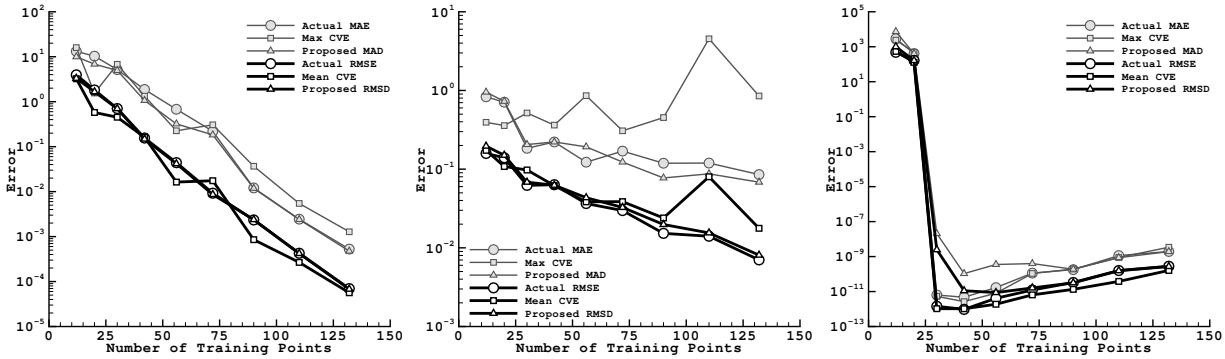

Figure 5. A comparison of the proposed error measures with actual errors and leave-one-out cross validation on two-dimensional test functions using PCE.

promise to validate the surrogate model without warranting exact function evaluations in applications of practical interest. Cross validation exhibits satisfactory tendencies in matching the actual errors but its overall performance is more chaotic. For all test cases a maximum of twenty-five nearest existing training points is used to build local approximations with MIR.

IV.A.2. Comparison with error distributions in the domain

In Figures 4 and 5 quantitative metrics have been used to validate the proposed error estimate for surrogate models. This section is intended to provide an insight into the spatial distribution of *the actual error,* ϵ, and *the proposed discrepancy,* δ. Figures 6, 7 and 8 show contour plots of the distribution of (i) local surrogate model error, ϵ_{local}, (ii) global kriging surrogate model error, ϵ_{global}, and (iii) the proposed discrepancy function, δ, for exponential, Runge and Rosenbrock test functions, respectively.

The main assumption of the framework has been that the local surrogate models provide a more accurate representation of their corresponding sub-domain than the global surrogate model. The local and global surrogate model errors shown in the leftmost and middle contour plots of Figures 6, 7 and 8 reveal that the local surrogate models are more accurate than the global surrogate model. For all the cases the local surrogate models use the closest 25 data

points to predict the function value at a test location. For the Rosenbrock test function (see Figure 8) the local surrogate is indeed a "second global surrogate model" as all the available data (25 data points) is used for its predictions.

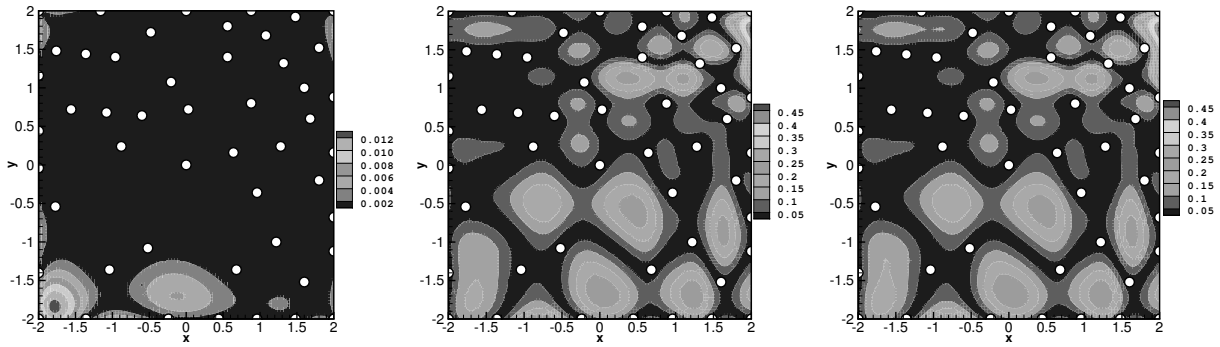

Figure 6. Contour plots for the exponential test function showing the distribution of: the local surrogate model error (left), the global kriging surrogate model error (middle) and the proposed discrepancy function. The global and local surrogate models are built with 50 and 25 training points (white circles), respectively.

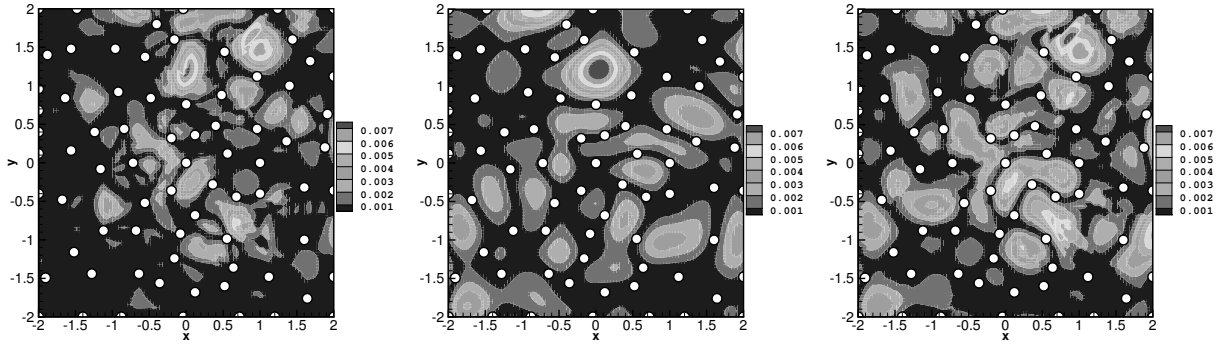

Figure 7. Contour plots for the Runge test function showing the distribution of: the local surrogate model error (left), the global kriging surrogate model error (middle) and the proposed discrepancy function. The global and local surrogate models are built with 75 and 25 training points (white circles), respectively.

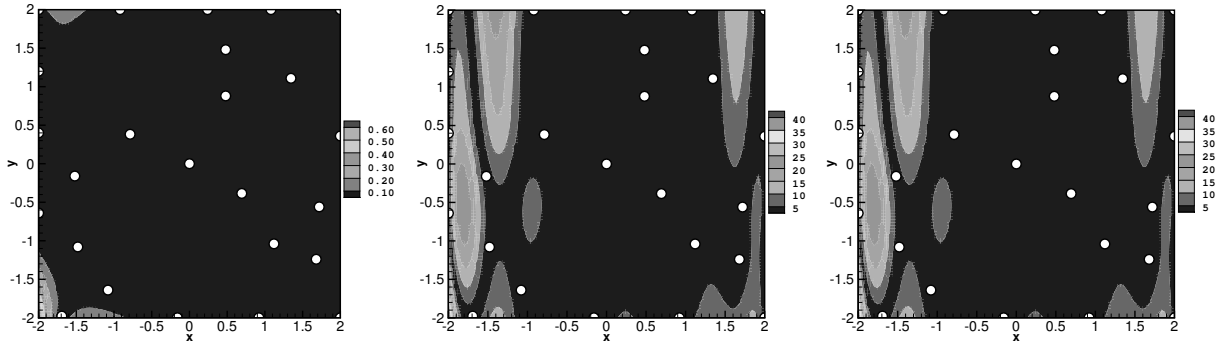

Figure 8. Contour plots for the Rosenbrock test function showing the distribution of: the local surrogate model error (left), the global kriging surrogate model error (middle) and the proposed discrepancy function. The global and local surrogate model(s) are both built with 25 training points (white circles).

The differences between the function predictions of the two surrogate models (local and

global) is proposed as an approximation to the actual error in the global surrogate model as discussed in Section III.A. From Figures 6, 7 and 8 a close agreement can be noticed between the global surrogate model's actual error distribution (middle) and the proposed discrepancy function (right), which explains the excellent trends shown in Figures 4 and 5. Only for the Runge function (see Figure 7) can differences be visually seen between the actual error and the proposed discrepancy function.

Generally, it is possible for the local surrogate models to provide less accurate representation of the domain, especially when only a few training points are available. Therefore, it is advisable to use all the available training data to build local surrogate model(s) during the first few selection cycles. For example, in this work the local surrogate models for the two-dimensional cases use all the available training data until the size of the training data set increases beyond 25 after which it is fixed at 25 for computational efficiency purposes.

As a final remark, if one were to continue the training point selection process, the framework would choose points where the discrepancies shown in the rightmost contour plots of Figures 6, 7 and 8 are large.

IV.B. Number of additional training points per cycle

The proposed dynamic training point selection features a progressive evolution of the training data set for the global surrogate model. The user provides the number of training points (N_{cyc}) to be added at each iteration to the training data set - a factor that determines the rate at which the training data set is evolved. In the case of PCE, the surrogate is built in steps of one polynomial order per cycle and the required number of additional points can be determined from Eq. (4) and the assumed oversampling factor which is two. In the case of kriging, the choice is left to the user as kriging does not mandate any requirements on the minimum number of points needed to build the surrogate. It is recommended to add a moderate number of training points per iteration to facilitate a *better evolution* of the training data set. Adding only a few points per cycle implies more computational burden since the kriging (or any response surface) has to be constructed more often to reach a fixed number of training points.

Figure 9 shows the effect of the number of training points added per iteration on the accuracy of the global surrogate for all three analytical test functions in two dimensions. Since the general monotonic behavior is preserved for all the tested cases (two, five and ten) and the trade-off with accuracy being small, one may accelerate the process by adding more training points per cycle. As an exception, for the two-dimensional Rosenbrock test function, the kriging surrogate model does not present a monotone behavior after reaching about 40 training points. It will be shown later (see Section IV.D.2) that kriging exhibits the same behavior with all the tested training point selection approaches which is due to the choice

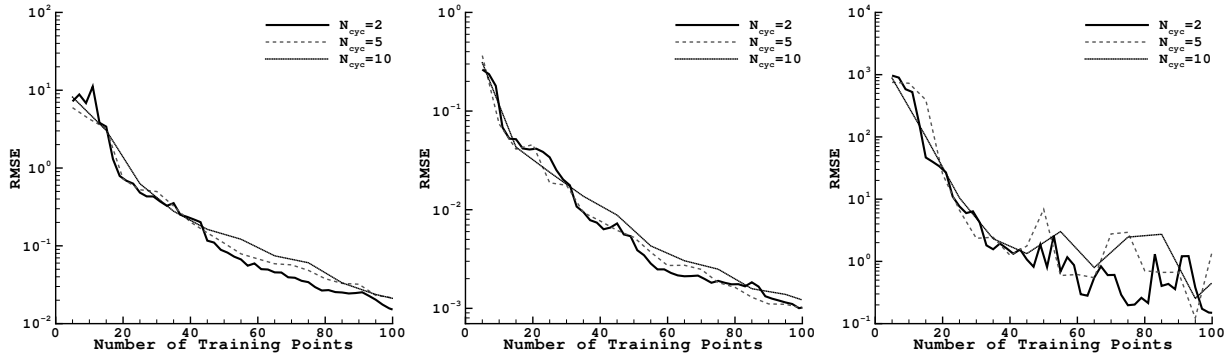

Figure 9. The effect of the number of training points selected per cycle on the RMSE for two-dimensional test functions using kriging.

of the spatial correlation function.

IV.C. Training point distribution

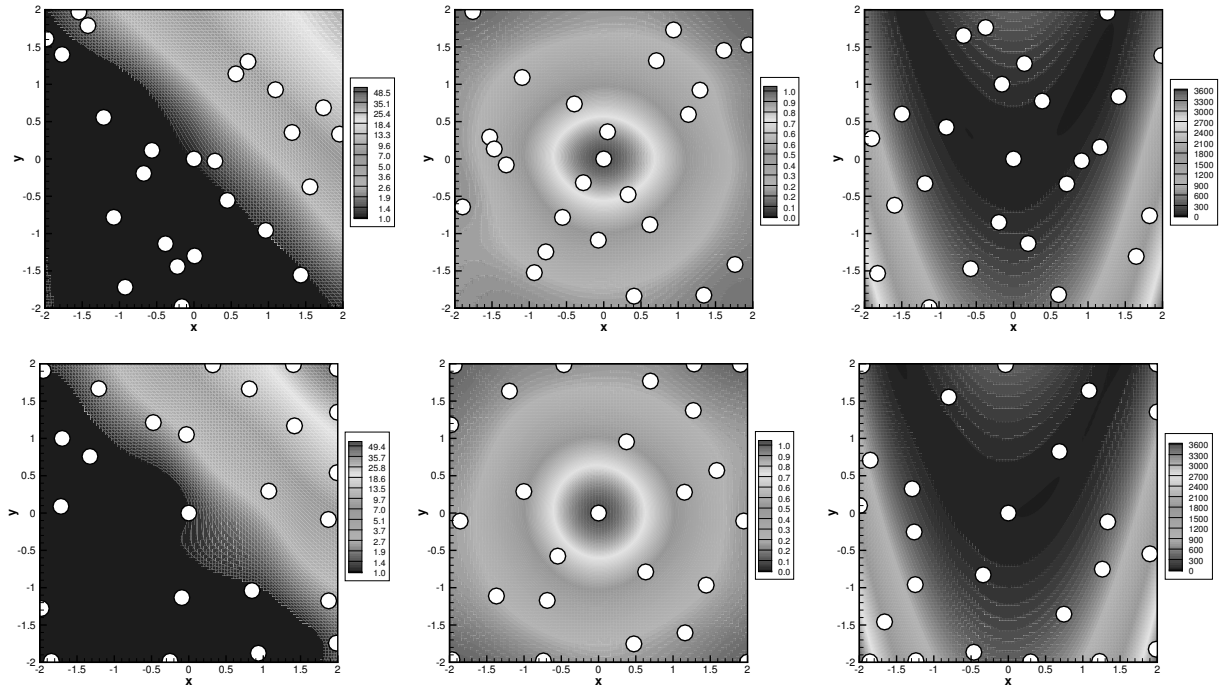

Figure 10. Training point distributions ($N = 25$) for two-dimensional test functions using LHS (top) and the dynamic method (bottom).

Figure 10 shows typical training point distributions for all three test functions using LHS as well as the proposed dynamic method using kriging. In this example, five points were selected per iteration until reaching the final training data set with 25 points. One can clearly observe that the dynamic strategy concentrates training points in regions where the curvature is high, near the peaks and bounds, and in regions where the points are sparse.

For example, the top row of Figure 10 (LHS sampling) has larger unsampled regions in

the domain than the bottom row (dynamic selection), where the training points are more spread throughout the domain. Such a strategy can save a lot of computational time when applied to high-fidelity simulations by reducing the number of required function evaluations to produce a globally accurate surrogate model. On the other hand, LHS tends to miss important locations and tends to affect the matrix conditioning through too closely spaced points.

IV.D. Accuracy of dynamically enhanced kriging surrogate model

A quantitative comparison of the accuracy of the dynamic training point selection enhanced kriging surrogate model will be provided in the following paragraphs by means of RMSE comparisons.

IV.D.1. Comparison with low-discrepancy sequences using kriging

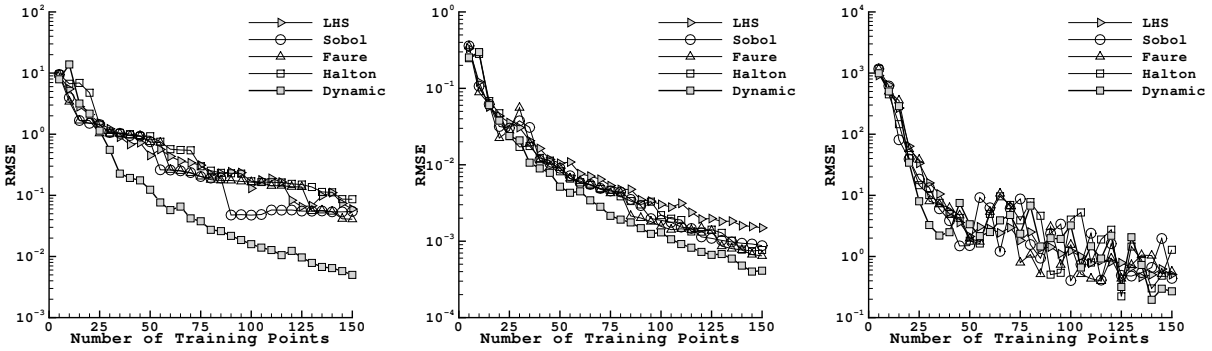

Figure 11. A comparison of dynamic training point selection with some quasi-Monte Carlo sequences and LHS on two-dimensional test functions using kriging.

The dynamic training point selection is compared with some quasi-Monte Carlo sequences such as Sobol, Faure, and Halton. LHS is also included in the comparisons to show its performance compared to other strategies. From Figure 11, it can be seen that the dynamic training point selection is better at producing accurate surrogate models than the other approaches. As discussed above, for the Rosenbrock test function all training point selection strategies suffer from poor convergence with kriging due to the choice of the spatial correlation function.

IV.D.2. Comparison with kriging MSE

The kriging surrogate model provides an error estimate through the expected mean squared error (MSE). One can use the MSE to guide the training point selection *i.e.,* add training points at locations where the MSE is large. Figure 12 shows a comparison of kriging surrogate models built with LHS, built by minimizing MSE and the proposed framework for training

point selection. For all three test functions, the proposed framework produces more accurate surrogate models.

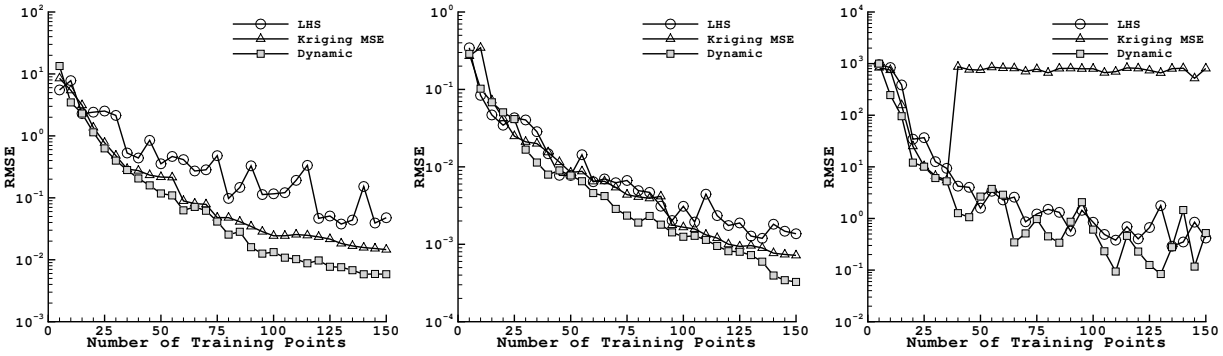

Figure 12. A comparison of dynamic training point selection with minimizing the MSE method on two-dimensional test functions using kriging. LHS is also included to show its relative performance.

For the Rosenbrock function f_3, the MSE method features a poor convergence after about 40 training points and is even outperformed by LHS. In order to alleviate the non-smooth convergence behavior of kriging for the Rosenbrock test function (using LHS, Sobol, Faure, Halton, MSE, and the proposed framework) other spatial correlation functions have been employed and the results are displayed in Figure 13. By comparing the rightmost Figure 12

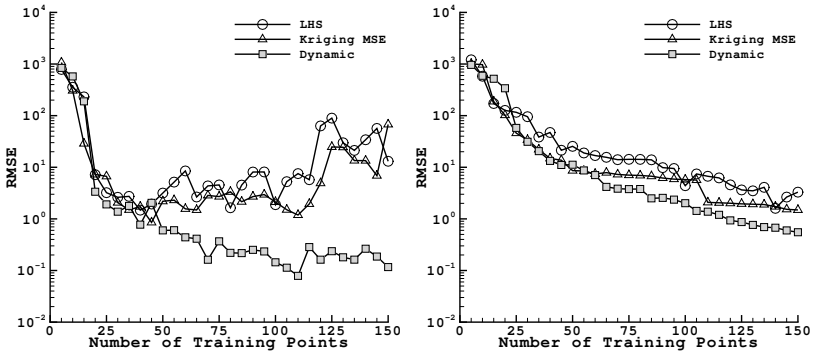

Figure 13. A comparison of accuracies of the kriging model built with Gaussian (left) and spline (right) spatial correlation functions for two-dimensional Rosenbrock test function.

(using Wendland C4) with Figure 13 (using Gaussian and spline) it can be noted that spline spatial correlation functions feature a much smoother convergence when compared to Wendland C4 and Gaussian spatial correlation functions. However, since a derivative-enhanced kriging model is employed in this work, Wendland C4 spatial correlation function [60] is used as default for the remainder of this article to ensure differentiability. Also, for further comparisons only LHS will be considered to reduce the complexity in presenting the results.

IV.D.3. Dynamic versus LHS in 2D using kriging

In order to account for the inherent randomness in LHS all two-dimensional results are averaged over ten separate runs and the mean results are presented with bounds referring

to the best and worst case.

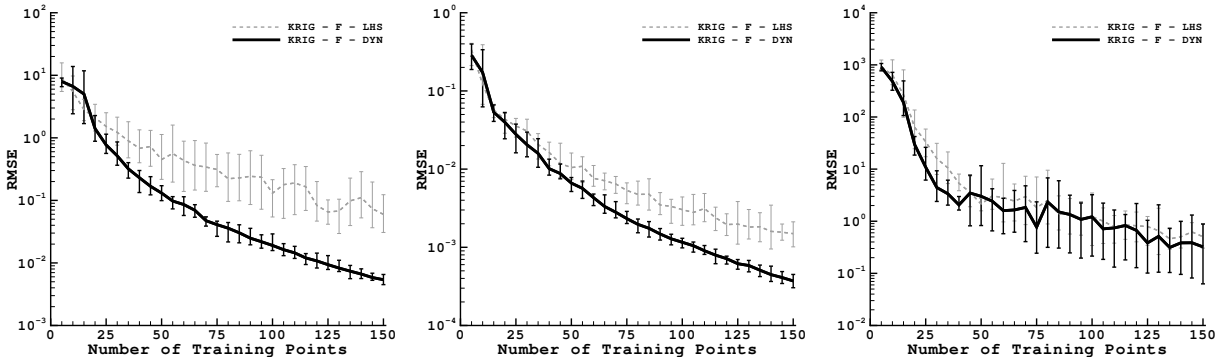

Figure 14. Plots comparing training point selections using LHS and the dynamic method on two-dimensional test functions (with function values only) with kriging. Five training points are selected per cycle.

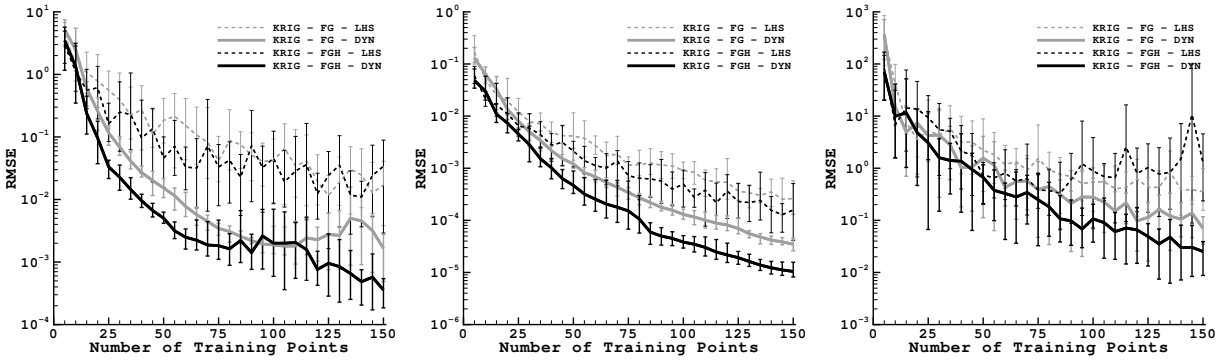

Figure 15. Plots comparing training point selections using LHS and the dynamic method on two-dimensional test functions (with higher-order derivative information) with kriging. Five training points are selected per cycle.

From Figure 14 one can infer that the dynamic method (shown with continuous lines) performs better than LHS (shown with dashed lines), both in terms of monotonicity and accuracy. The dynamic training point selection improves the accuracy of the surrogate by roughly an order of magnitude when compared to LHS, for the first two test functions. For the Rosenbrock test function (f_3), though not as distinct as for the other test functions, the lower bounds show that the dynamic method is still more accurate than LHS.

The training point selection in the presence of derivative information is shown separately in Figure 15. It can be noted that the addition of gradient and Hessian information helps to improve the accuracy of the response surface for both LHS and the dynamic training point selection. The dynamic training point selection improves the accuracy even more by placing the higher-order derivative information in the most viable locations. This behavior can be observed across all test functions in Figure 15. Also, it can be seen that the LHS Hessian cases (FGH) of f_1 and f_3 (Figure 15 left and right, respectively), are not more accurate than the gradient cases (FG) as one would expect. But with the dynamic method, additional auxiliary information always improves the accuracy of the surrogate model.

The smaller bounds on the RMSE for the dynamic method in the figures indicate that it is less random than LHS implying that only one run is sufficient whereas most random sample point selection methods require multiple runs to ensure that "bad luck" does not impede the results. In most instances, the upper bounds of the dynamic method (corresponding to its worst approximation) are still better than the lower bounds of LHS (corresponding to its best approximation). Hence one can expect the dynamic training point selection method to produce more accurate surrogates for a fixed computational budget. The superiority of the dynamic method over other quasi-Monte Carlo sequences has already been demonstrated in Figure 11.

IV.D.4. Dynamic versus LHS in 5D using kriging

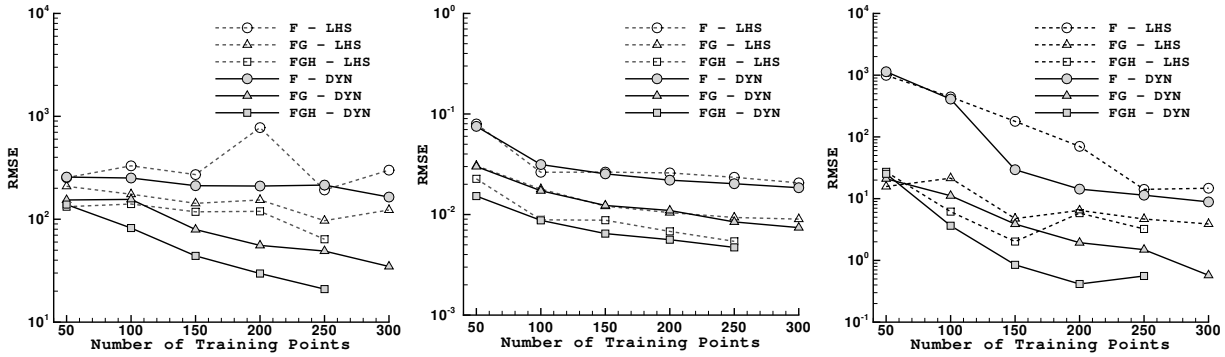

Figure 16. Plots comparing training point selections using LHS and the dynamic method on five-dimensional test functions with kriging. Fifty training points are selected per cycle.

Figure 16 shows the results for five-dimensional test functions. Due to the computational resources needed to build five-dimensional surrogate models, error bounds are not included. It can be seen that the dynamic method (shown as continuous lines) helps improve the accuracy of the kriging surrogate compared to LHS (shown as dashed lines). The advantage of including derivative information is also more evident for these higher-dimensional test cases. If the proposed dynamic training point selection (or any response-based method) is used, the surrogate is built progressively starting from an initial (small) number of training points, to achieve better accuracy for a fixed computational budget. The cost of building the surrogate model multiple times can generally be neglected in comparison with the computational cost of high-fidelity physics-based simulations. If the surrogate construction should become too expensive, one can accelerate the process by selecting a larger number of additional training points per cycle for a minimal trade-off with accuracy as demonstrated in Figure 9.

IV.E. Accuracy of of dynamically enhanced polynomial chaos expansion

Results pertaining to the polynomial chaos expansions are discussed in this section. An oversampling factor of two is enforced to improve the matrix conditioning. The dynamic training point selection is initiated by building a second-order accurate PCE including a training point at the center of the domain. As the order of expansion increases, the required number of training points are picked dynamically.

IV.E.1. Dynamic versus LHS in 2D using PCE

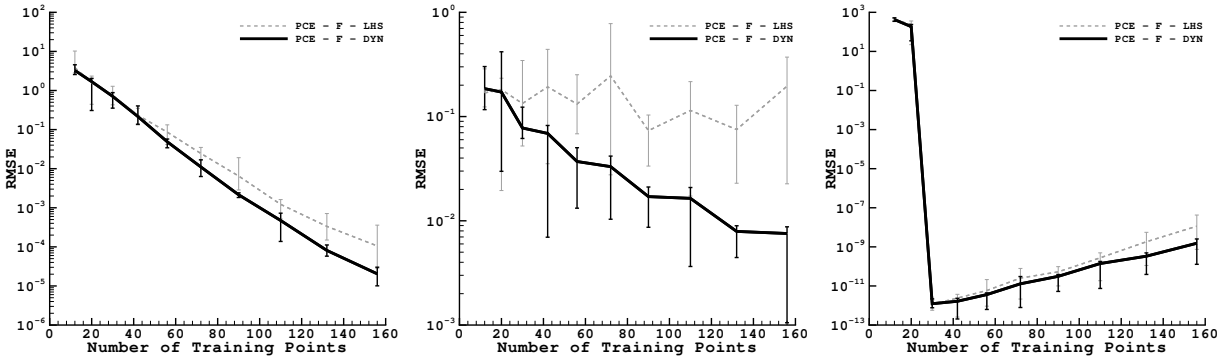

Figure 17. Plots comparing training point selections using LHS and the dynamic method on two-dimensional test functions (function values only) with PCE. The order of expansion, p, ranges from 2 to 11.

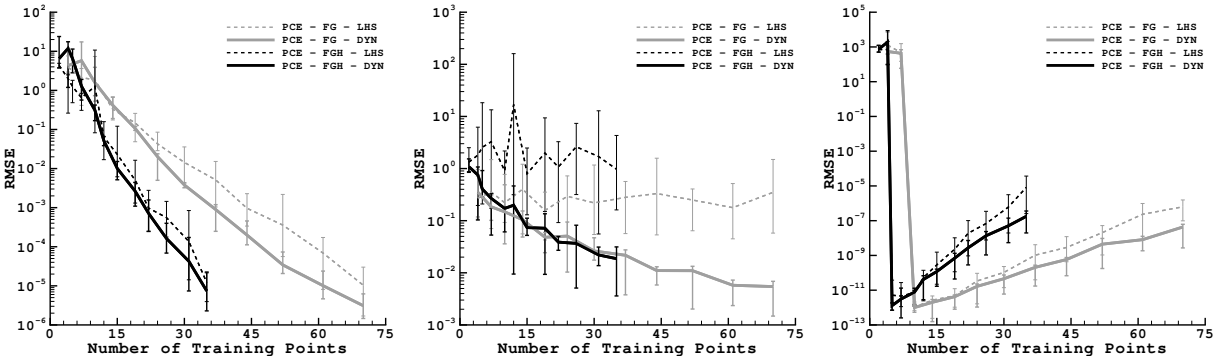

Figure 18. Plots comparing training point selections using LHS and the dynamic method on two-dimensional test functions (with higher-order derivative information) with PCE. The order of expansion, p, ranges from 2 to 12.

Figure 17 compares PCE with LHS (shown as dashed lines) with that of PCE with dynamic training point selection (shown as continuous lines) for all three test functions, along with error bounds similar to the results presented for kriging. As one can see, the convergence behavior of PCE with LHS is chaotic compared to that of the dynamic strategy which is monotonic instead. Although the rational Runge function (f_2) is known to pose difficulties for PCE [63], the dynamic method shows good convergence whereas LHS shows very poor results. Note that recently a rational polynomial chaos expansion scheme has been developed by Sheshadri *et al.* [63] to address the difficulty of PCE with such functions.

The fourth-order polynomial Rosenbrock function (f_3) is captured exactly using only a few training points, however, it suffers from polynomial over-fitting [64] as the order of expansion increases further. Overall, the dynamic training point selection consistently produces more accurate surrogates compared to LHS. Figure 18 shows the improved performance of the dynamic method for choosing training points that contain gradient (labeled FG) and Hessian information (labeled FGH). In particular, for the Runge function, the advantage of the proposed framework is very pronounced.

IV.E.2. Dynamic versus LHS in 5D using PCE

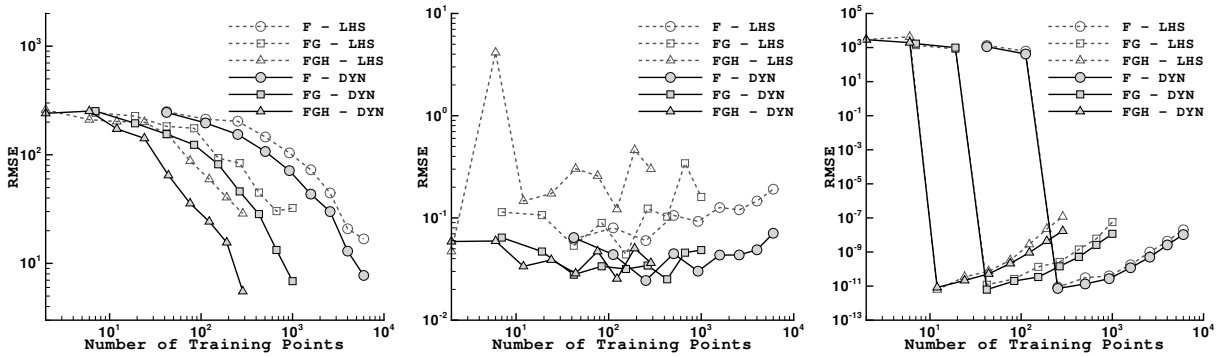

Figure 19. Plots comparing training point selections using LHS and the dynamic method on five-dimensional test functions. The order of expansion, p, ranges from 2 to 10.

Figure 19 shows the five-dimensional results. It is again evident that the dynamic method is more consistent in producing a good PCE surrogate compared to LHS which tends to show random fluctuations. When the function values alone are used, about 6000 points are needed for a tenth order polynomial. If gradients are used, only about 1000 data points are needed with function and gradient information and only about 200 points are required with function, gradient and Hessian information to roughly obtain the same level of accuracy.

IV.F. Some observations on using higher-order derivatives

The following discussion is aimed at providing guidelines for choosing an appropriate combination of training point selection method and incorporation of higher-order derivative information. Generally, the use of derivative information provides improved surrogate models for both kriging and polynomial chaos, irrespective of whether the training points are chosen dynamically or at random. As discussed in Section I, efficient gradient and Hessian calculation methods are available for high-fidelity physics-based simulations and they provide the means to reduce the effects of the "curse of dimensionality".

Figures 20 and 21 take into account the computational time for calculating the gradient and Hessian and plot the model accuracy versus the number of equivalent exact function

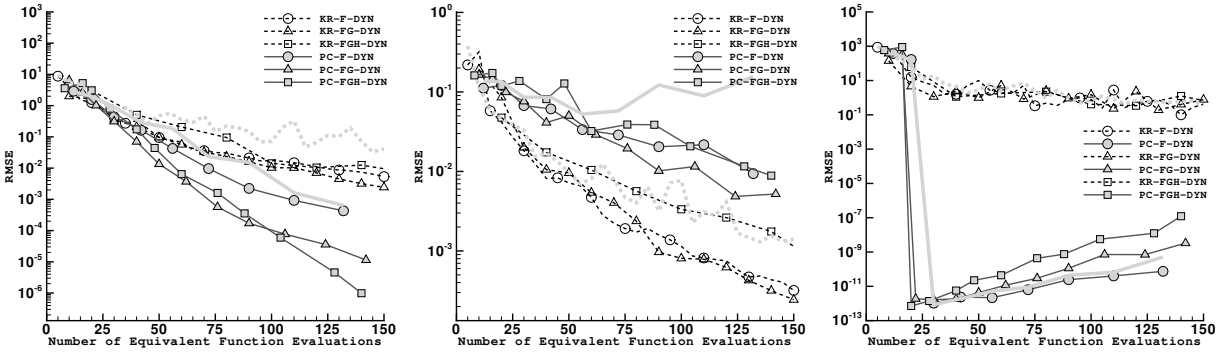

Figure 20. RMSE versus the number of equivalent function evaluations for two-dimensional test functions using kriging and PCE with dynamic training point selection. Reference LHS results with function values only are shown with dashed and continuous gray lines corresponding to kriging and PCE, respectively.

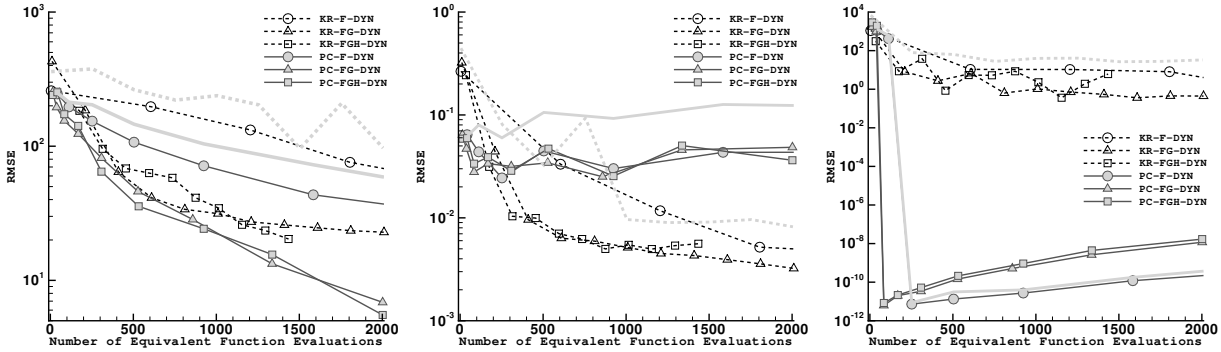

Figure 21. RMSE versus the number of equivalent function evaluations for five-dimensional test functions using kriging and PCE with dynamic training point selection. Reference LHS results with function values only are shown with dashed and continuous gray lines corresponding to kriging and PCE, respectively.

evaluations for two- and five-dimensional test functions, respectively. For example, the computational time for evaluating N data points with function values only (F), is the same as evaluating $\frac{N}{2}$ points with function and gradient values (FG), which in turn is the same as $\frac{N}{M+2}$ points with function, gradient and Hessian information (FGH), using the adjoint method as discussed in Section I.C, where M is the number of dimensions. From these figures it can be inferred that the gradient enhanced surrogates are computationally more efficient than the others. The Hessian enhancement does not yield convincing results, however, one could choose to use the Hessian information only in specific locations, for example, where data points are sparsely distributed rather than to add Hessian information to all training points as done here.

In summary, LHS and some low-discrepancy sequences have been shown to produce less accurate results than the proposed dynamic method. Additionally, the computational advantage of building gradient enhanced surrogate models is shown in Figures 20 and 21. Hence, based on these observations, the dynamic training point selection in conjunction with the use of gradient information for the construction of surrogate models offers the best of both worlds.

V. Aerodynamic Database of Drag and Lift Coefficients

In this section, the benefits of the dynamic training point selection is demonstrated for an aerodynamic test case. Kriging and polynomial chaos are applied to construct two-dimensional aerodynamic databases of drag and lift coefficients.

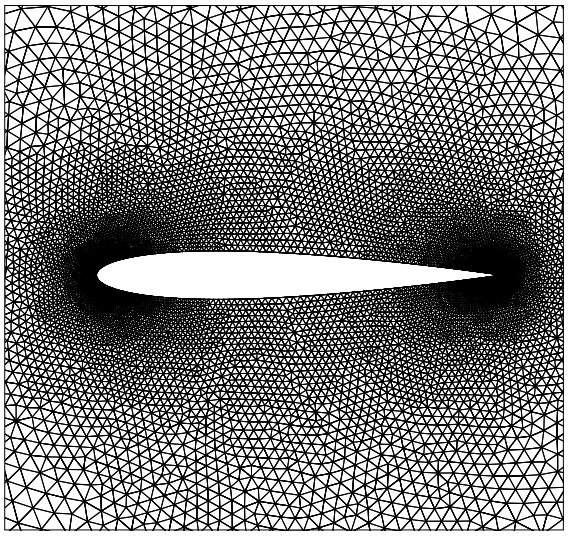

Figure 22. Computational mesh around NACA0012 with 19,548 triangular elements.

The steady inviscid flow around a NACA 0012 airfoil governed by the Euler equations is solved by using a second-order accurate finite-volume approach [65, 66]. The computational mesh with 19,548 triangular elements is shown in Figure 22. The variations of the drag and lift coefficients with changes in Mach number ($0.5 \leq M_\infty \leq 1.5$) and angle of attack ($0^\circ \leq \alpha \leq 5^\circ$) are studied. An "exact" database is obtained from Euler flow solves on a Cartesian mesh of $N_t = 51 \times 51 = 2601$ equispaced nodes and is used to validate both kriging and PCE surrogate models.

Figure 23 shows the gradients obtained using a discrete-adjoint approach (demonstrated to be accurate to machine precision [65, 66]) for two typical angles of attack, $\alpha = 1^\circ$ and 4°. It can be seen that they are quite noisy (due to the transonic behavior of the flux limiters) and are hence counterproductive in the construction of the surrogate models and will not be used here.

V.A. Validation of proposed error estimates

V.A.1. Comparison with actual errors and cross validation

In Figures 24 and 25, the proposed error estimates (MAD and RMSD) are compared with the actual errors (MAE and RMSE) as well as leave-one-out cross validations for kriging and PCE, respectively. The excellent agreement observed for analytical test functions cannot be

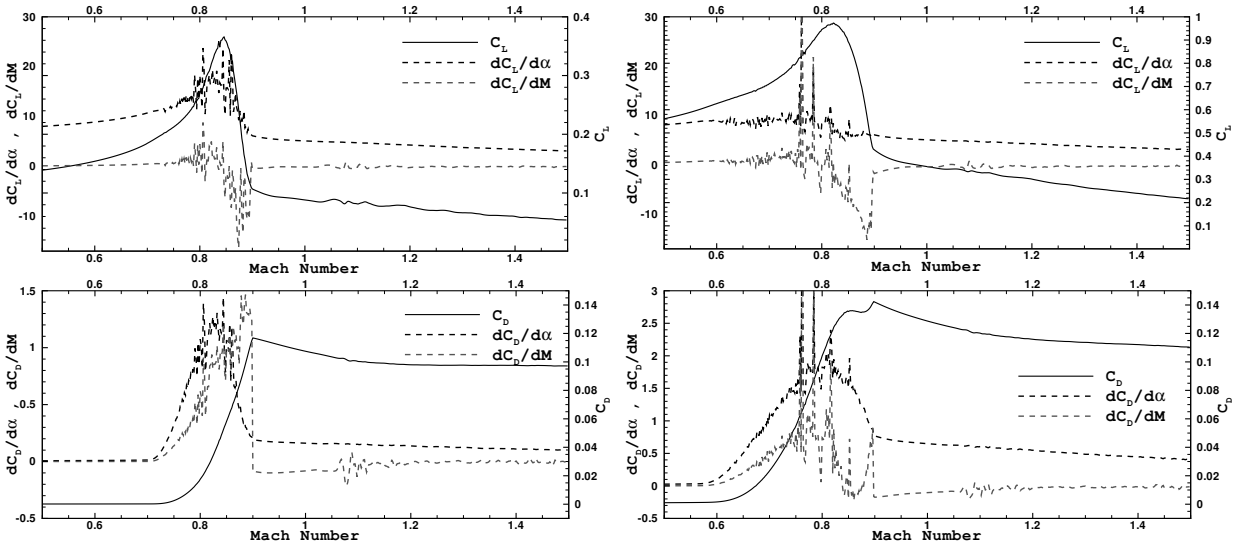

Figure 23. Noisy gradients for $\alpha = 1^\circ$ (left) and $\alpha = 4^\circ$ (right).

seen here. This is due to the reason that the kriging excels MIR in approximating the non-smooth drag and lift functionals, thereby making our initial assumption of a more accurate local surrogate model invalid. However, when compared to cross validation the proposed error measures are still much better.

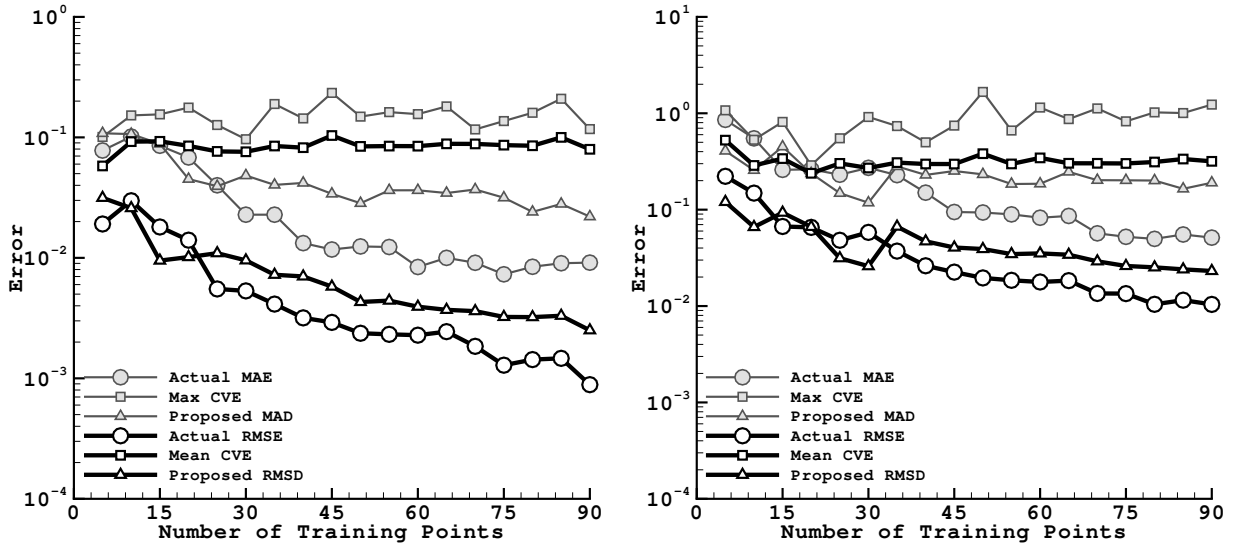

Figure 24. A comparison of proposed error measures with actual errors and leave-one-out cross validation for drag (left) and lift coefficients (right) using kriging.

V.A.2. Comparison with error distributions in the domain

Figures 26 and 27 show the distribution of local (left) and global surrogate model error (middle), along with the proposed discrepancy function (right), for drag and lift coefficients,

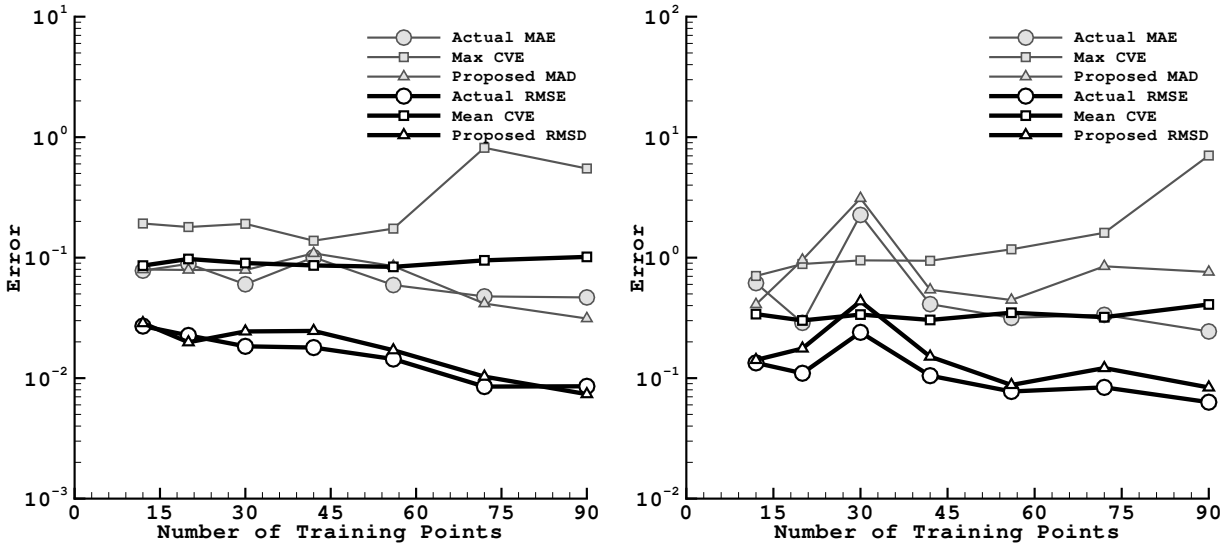

Figure 25. A comparison of proposed error measures with actual errors and leave-one-out cross validation for drag (left) and lift coefficients (right) using PCE.

respectively, using kriging as the global surrogate model. The regions where the actual errors are high in the global kriging model (*i.e.,* the transonic regime) are predicted reasonably well by the discrepancy function, though the magnitudes are quite different. It can also be seen that the MIR local surrogate models are less accurate than the global kriging model for the drag and lift coefficients. Another observation that one can make is that the dynamic training point selection method is adding a majority of the training points in the transonic region. The kriging MSE minimization approach would not exhibit this behavior since MSE is a measure of space filling. If one also prefers a better spread of training points, the *control parameter* described in Section III.A can be set to a higher value such as 1.1 or 1.2, that will enforce a more strict geometric constraint, or the vice versa when using the framework for optimizations (not studied here).

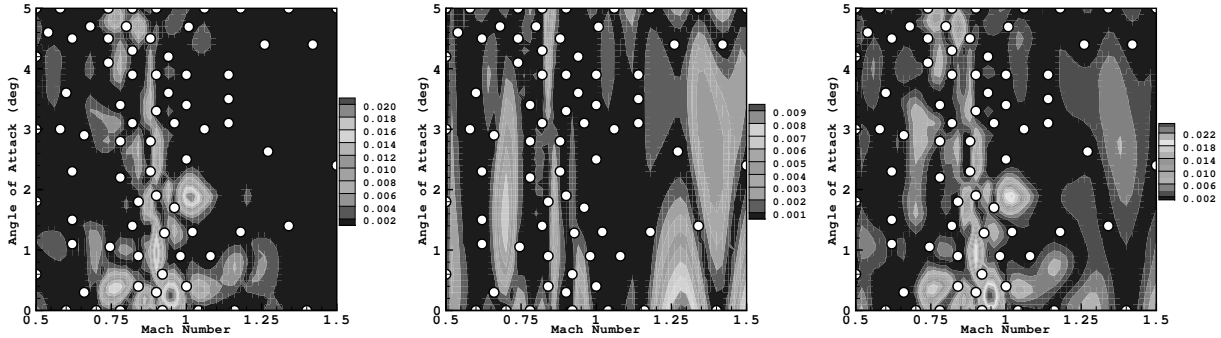

Figure 26. Contour plots for the drag coefficient showing the distribution of: the local surrogate model error (left), the global kriging surrogate model error (middle) and the proposed discrepancy function. The global and local surrogate models are built with 75 and 25 training points (white circles), respectively.

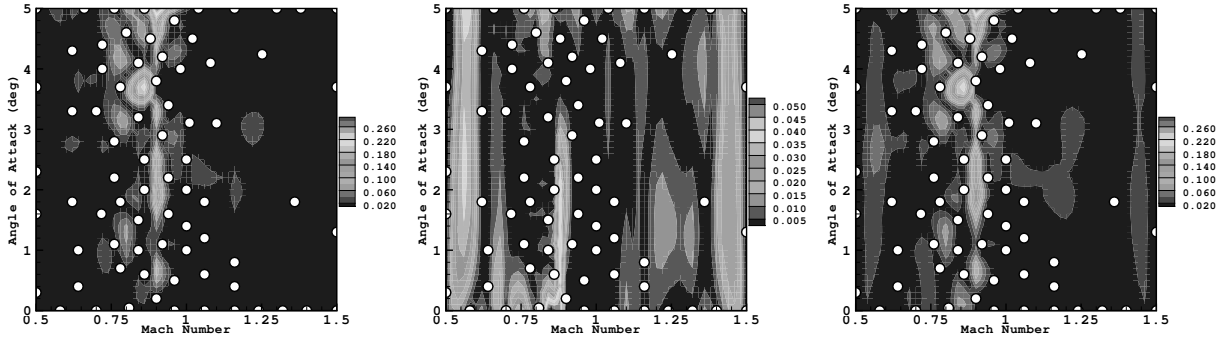

Figure 27. Contour plots for the lift coefficient showing the distribution of: the local surrogate model error (left), the global kriging surrogate model error (middle) and the proposed discrepancy function. The global and local surrogate models are built with 75 and 25 training points (white circles), respectively.

V.B. Construction of aerodynamic database

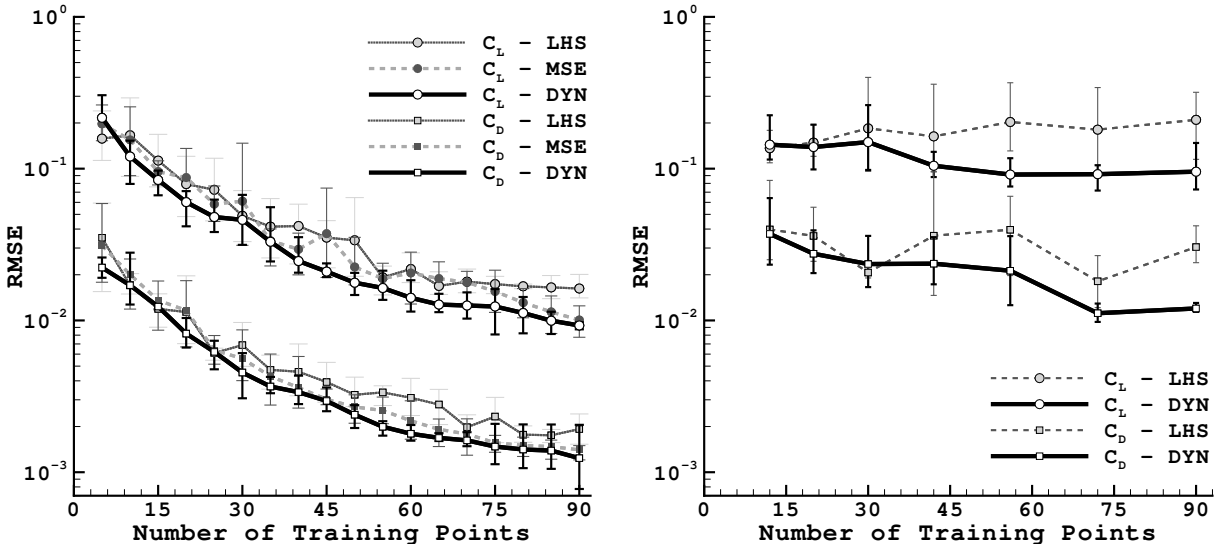

Figure 28. Plot of RMSE versus the number of training points for kriging (left) and PCE (right).

Figure 28 compares the performance of the proposed dynamic training point selection with other DoE methods for kriging (left) and polynomial chaos (right), respectively. Here, ten separate runs are averaged and the mean, best and worst trends are shown. It can be seen that the dynamic training point selection performs better than LHS for both surrogate models and for both functionals. In Figure 28 (left), which corresponds to kriging, minimization of MSE is also compared with the other two methods. Though not as distinct as seen with analytical test functions, it can be observed that the dynamic method is still consistently more monotonic and accurate. From Figure 28 (right), it can be noticed that the accuracy of PCE tends to deteriorate with increasing order of the expansion when LHS is used. However, the dynamic training point selection prevents the early onset of such behavior.

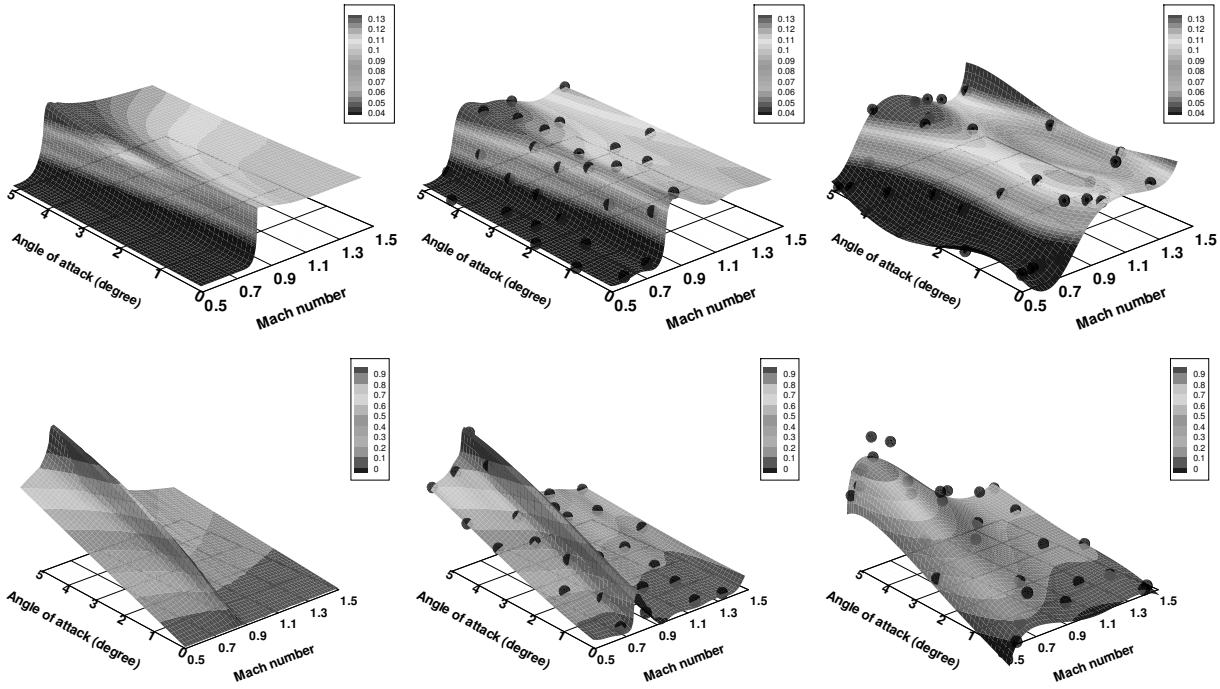

Figure 29. Contours of exact database (left), kriging (middle) and PCE (right) for drag (top) and lift coefficients (bottom) with 30 training points chosen with dynamic training point selection.

Figure 29 shows the exact and surrogate based contours for both the drag and lift coefficients. It can be inferred that the kriging model shown in the middle column of Figure 29 is in good agreement with the exact model when compared to PCE which features high over- and undershoots in the domain as shown in the right column of Figure 29. In addition, the kriging is able to capture the transonic behavior very well, demonstrating its ability to model non-smooth functions. Thus, for this test case the kriging proves to be a much better surrogate model than PCE.

Kriging supports the usage of both high- and low-fidelity training points [19,31–36]. The general idea is to combine trends from low-fidelity data (e.g., coarser meshes, less sophisticated models) with interpolations of high-fidelity data (e.g., finer meshes, better models, experimental data). This approach can help to reduce the time taken to build an accurate surrogate model since the low-fidelity data can be obtained much faster. In this work a simple cokriging with cross-covariances between the low- and high-fidelity data is used [19,35]. The low-fidelity data is calculated on a mesh with only 4,433 triangular elements (not shown), which is roughly four times cheaper to solve than the mesh shown in Figure 22 on which the high-fidelity data is calculated.

Figure 30 shows the variable-fidelity kriging contours for the drag and lift databases. Here, 15 high-fidelity and 60 low-fidelity training points were used in the construction of the kriging surrogate and the training point locations are shown as spheres, where the dark spheres are high-fidelity training points picked dynamically and the smaller gray spheres

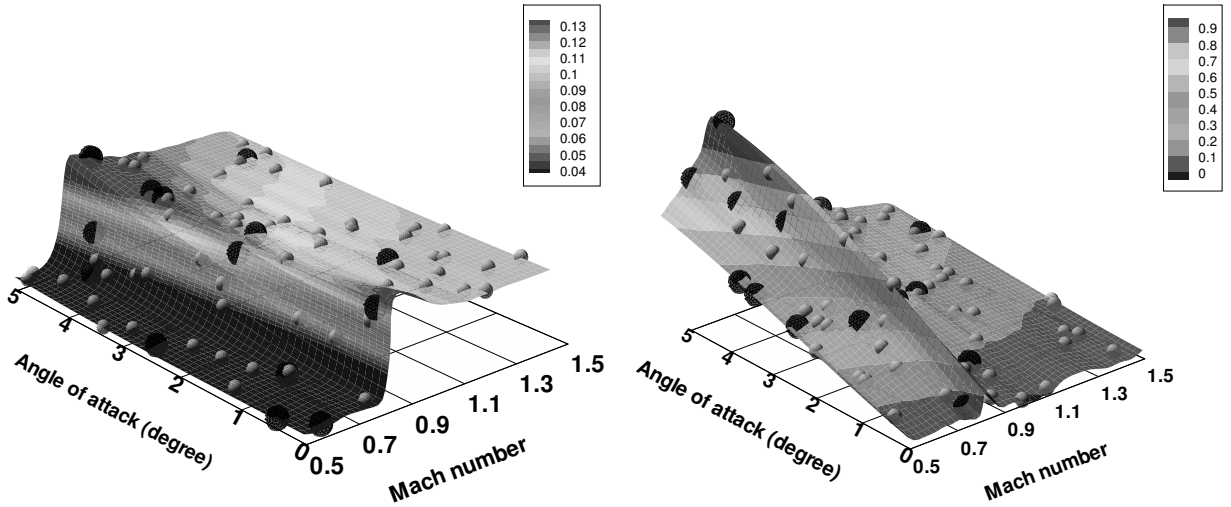

Figure 30. Kriging contour plots demonstrating the use of variable-fidelity data for drag (left) and lift (right) coefficients.

are low-fidelity training points picked via LHS. The overall computational cost is roughly equivalent to the construction of the kriging surrogate model using 30 high-fidelity training points. One can notice an improvement in capturing the contour levels using the variable-fidelity kriging by comparing Figures 29 and 30 as well as from Table 1.

Table 1. RMSE comparisons for different kriging models

RMSE	High-fidelity (30 high-fidelity points)	Variable-fidelity (15 high-fidelity and 60 low-fidelity points)
Drag Coefficient	0.39×10^{-2}	0.31×10^{-2}
Lift Coefficient	0.35×10^{-1}	0.18×10^{-1}

VI. Conclusions

A dynamic training point selection framework has been proposed and applied to two different surrogate models: kriging and polynomial chaos expansions. It provides a better choice of training point locations for both surrogate models since it inherits the characteristics of both domain- and response-based training point selection methods, *i.e.,* a geometric criterion is used to spread out the points and the points are chosen based on discrepancies between local and global surrogate approximated response values. Comparisons with latin hypercube sampling and other domain-based training point selection approaches show that more monotone convergence behavior and better accuracies are achieved. The dynamic method does not require extra exact function evaluations and the extra computational overhead of

building and evaluating local surrogate models is negligible compared to most high-fidelity physics-based function evaluations.

The framework also addresses the question of training point selection in the presence of higher-order derivative information: the local surrogate models use the available derivative information in approximating the function values and influence the training point selection via the discrepancy function. As a result, it is shown that the dynamic method ably chooses better locations to evaluate the gradient and Hessian information than LHS. In this process, PCE has been newly enhanced to incorporate Hessian information. The computational advantage of using higher-order derivative information for the construction of surrogate models in conjunction with the proposed training point selection is also discussed. It is shown that the addition of gradient information offers the highest computational advantage in the construction of surrogates over its counterparts: function values only, and function, gradient and Hessian values.

Lastly, the framework also introduces a local as well as global measure of surrogate model accuracy. The proposed maximum absolute discrepancy (MAD) and root mean square discrepancy (RMSD) show great promise for measuring surrogate model error in applications of practical interest where it is intractable to calculate the actual errors (MAE or RMSE). This is demonstrated by a good agreement between the proposed measures and actual errors for a variety of analytical test functions and an aerodynamic test case for both surrogate models.

Acknowledgments

This work has been supported in part by the University of Dayton Office for Graduate Academic Affairs through the Graduate Student Summer Fellowship Program. The authors acknowledge Wataru Yamazaki for his kriging model, Qiqi Wang for his MIR model and Karthik Mani for his flow solver. The authors are also very grateful to anonymous reviewers for their invaluable suggestions without which this article would not have been the same.

References

[1]Fang, K. T., Lin, D., and Winker, P., "Uniform design: Theory and application," *Technometrics*, Vol. 42, No. 3, 2000, pp. 237–248.

[2]Metropolis, N. and Ulam, S., "The Monte Carlo method," *Journal of the American Statistical Association*, Vol. 44, 1949, pp. 335–341.

[3]McKay, M. D., Conover, W. J., and Beckman, R. J., "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, Vol. 21, No. 2, 1979, pp. 239–245.

[4]Xiu, D., *Numerical Methods for Stochastic Computations: A Spectral Method Approach*, Princeton University Press.

[5]Maître, O. P. L. and Knio, O. M., *Spectral Methods for Uncertainty Quantification, Scientific Computation*, Springer.

[6]Wong, T. T., Luk, W. S., and Heng, P. A., "Sampling with Hammersley and Halton points," *J. Graph. Tools*, Vol. 2, No. 2, 1997, pp. 9–24.

[7]Sobol, I., *A primer for the Monte Carlo Method*, CRC Press, 1994.

[8]Rosenbaum, B. and Schulz, V., "Comparing sampling strategies for aerodynamic Kriging surrogate models," *Journal of Applied Mathematics and Mechanics*, Vol. 92, 2012, pp. 852–868.

[9]Mehmani, A., Chowdhury, S., Zhang, J., and Messac, A., "Regional Error Estimation of Surrogates (REES)," AIAA Paper, 2012-5707, 2012.

[10]Alexandrov, N. M., Dennis, J. E., Lewis, R. M., and Torczon, V., "A Trust-Region Framework for Managing the Use of Approximation Models in Optimization," *Structural Optimization*, Vol. 15, 1998, pp. 16–23.

[11]Roderick, O., Anitescu, M., and Fischer, P., "Polynomial Regression Approaches Using Derivative Information for Uncertainty Quantification," *J. of Nuclear Science and Engineering*, Vol. 164, No.2, 2010, pp. 122–139.

[12]Li, Y., Anitescu, M., Roderick, O., and Hickernell, F., "Orthogonal Bases for Polynomial Regression with Derivative Information in Uncertainty Quantification," *International Journal for Uncertainty Quantification*, Vol. 1, No.4, 2011, pp. 297–320.

[13]Cheng, H. and Sandu, A., "Collocation least-squares polynomial chaos method." SCS/ACM, 2010.

[14]Keane, A. and Nair, P., *Computational Approaches for Aerospace Design*, John Wiley & Sons, 2005.

[15]Arora, J. S., *Optimization of Structural and Mechanical Systems*, World Scientific Publishing Co. Pte. Ltd., 2007.

[16]Forrester, A., Sobester, A., and Keane, A., *Engineering Design via Surrogate Modelling: A Practical Guide*, John Wiley & Sons, 2008.

[17]Queipo, N., Haftka, R., Shyy, W., Goel, T., Vaidhyanathan, R., and Tucker, P., "Surrogate-based Analysis and Optimization," *Progress in Aerospace Sciences*, Vol. 41, No. 1, 2005, pp. 1–28.

[18]Bozdogan, H., "Akaike's Information Criterion and Recent Developments in Information Complexity," *J. Math. Psychol.*, Vol. 44, No. 1, 2000, pp. 62–91.

[19]Yamazaki, W. and Mavripilis, D. J., "Derivative-Enhanced Variable Fidelity Surrogate Modeling for Aerodynamic Functions," *AIAA Journal*, Vol. 51, No. 1, 2013, pp. 126–137.

[20]Chung, H. S. and Alonso, J. J., "Using Gradients to Construct Cokriging Approximation Models for High-Dimensional Design Optimization Problems," AIAA Paper, 2002-0317, 2002.

[21]Laurenceau, J. and Sagaut, P., "Building Efficient Response Surfaces of Aerodynamic Functions with Kriging and Cokriging," *AIAA Journal*, Vol. 46, No. 2, 2008, pp. 498–507.

[22]Yamazaki, W., Rumpfkeil, M. P., and Mavriplis, D. J., "Design Optimization Utilizing Gradient/Hessian Enhanced Surrogate Model," AIAA Paper, 2010-4363, 2010.

[23]Errico, R. M., "What is an adjoint model?" *Bulletin of the American Meteorological Society*, Vol. 8(11), 1997, pp. 2577–2591.

[24]Mani, K. and Mavriplis, D. J., "Unsteady Discrete Adjoint Formulation for Two-Dimensional Flow Problems with Deforming Meshes," *AIAA Journal*, Vol. 46 No. 6, 2008, pp. 1351–1364.

[25]Sherman, L. L., Taylor III, A. C., Green, L. L., and Newman, P. A., "First- and second-order aerodynamic sensitivity derivatives via automatic differentiation with incremental iterative methods," *Journal of Computational Physics*, Vol. 129, 1996, pp. 307 – 331.

[26]Taylor III, A. C., Green, L. L., Newman, P. A., and Putko, M., "Some Advanced Concepts in Discrete Aerodynamic Sensitivity Analysis," *AIAA Journal*, Vol. 41 No. 7, 2003, pp. 1224–1229.

[27]Ghate, D. and Giles, M. B., "Inexpensive Monte Carlo uncertainty analysis," in "Recent Trends in Aerospace Design and Optimization, Tata McGraw-Hill, New Delhi," , 2006, pp. 203–210.

[28]Chalot, F., Dinh, Q., Herbin, E., Martin, L., Ravachol, M., and Roge, G., "Estimation of the impact of geometrical uncertainties on aerodynamic coefficients using CFD," AIAA Paper, 2068-2008, April, 2008.

[29]Rumpfkeil, M. P. and Mavriplis, D. J., "Efficient Hessian Calculations using Automatic Differentiation and the Adjoint Method," AIAA Paper, 2010-1268, January, 2010.

[30]Rumpfkeil, M. P. and Mavriplis, D. J., "Efficient Hessian Calculations using Automatic Differentiation and the Adjoint Method with Applications," *AIAA Journal*, Vol. 48, No. 10, 2010, pp. 2406–2417.

[31]Han, Z. H., Goertz, S., and Zimmermann, R., "Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function," *Aerospace Science and Technology*.

[32]Han, Z. H., Zimmermann, R., and Goertz, S., "Alternative Cokriging Method for Variable-Fidelity Surrogate Modeling," *AIAA Journal*, Vol. 50, No. 5, 2012, pp. 1205–1210.

[33]Han, Z. H., Zimmermann, R., and Goertz, S., "On Improving Efficiency and Accuracy of Variable-Fidelity Surrogate Modeling in Aero-data for Loads Context," CEAS 2009 European Air and Space Conference, 2009.

[34]Han, Z. H., Zimmermann, R., and Goertz, S., "A New Cokriging Method for Variable-Fidelity Surrogate Modeling of Aerodynamic Data," AIAA Paper, 2010-1225, 2010.

[35]Yamazaki, W., "Uncertainty Quantification via Variable Fidelity Kriging Model," *Japan Society of Aeronautical Space Sciences*, Vol. 60, 2012, pp. 80–88.

[36]Yamazaki, W. and Mavriplis, D. J., "Derivative-Enhanced Variable Fidelity Surrogate Modeling for Aerodynamic Functions," AIAA Paper, 2011-1172, 2011.

[37]Mavripilis, D. J., "Aerodynamic Drag Prediction Using Unstructured Mesh Solvers, VKI Lecture Notes, CFD-Based Aircraft Drag Prediction and Reduction, Von Karman Institute for Fluid Dynamics, Rhode-Saint-Genese, Belgium," , 2003.

[38]Yamazaki, W., Mouton, S., and Carrier, G., "Efficient Design Optimization by Physics-Based Direct Manipulation Free-Form Deformation," AIAA Paper, 2008-5953, 2008.

[39]Wang, Q., Moin, P., and Iaccarino, G., "A rational interpolation scheme with super-polynomial rate of convergence," *SIAM Journal of Numerical Analysis*, Vol. 47, No. 6, 2010, pp. 4073–4097.

[40]Wang, Q., Moin, P., and Iaccarino, G., "A High-Order Multi-Variate Approximation Scheme for Arbitrary Data Sets," *Journal of Computational Physics*, Vol. 229, No. 18, 2010, pp. 6343–6361.

[41]Cressie, N., "The Origins of Kriging," *Mathematical Geology*, Vol. 22, No. 3, 1990, pp. 239–252.

[42]Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P., "Design and Analysis of Computer Experiments," *Statistical Science*, Vol. (4), 1989, pp. 409–423.

[43]Jeong, S., Murayama, M., and Yamamoto, K., "Efficient Optimization Design Method Using Kriging Model," *Journal of Aircraft*, Vol. 42, No. 2, 2005, pp. 413–420.

[44]Martin, J. D. and Simpson, T. W., "Use of Kriging Models to Approximate Deterministic Computer Models," *AIAA Journal*, Vol. 43, No.4, 2005, pp. 853–863,.

[45]Rumpfkeil, M. P., Yamazaki, W., and Mavriplis, D. J., "Uncertainty Analysis Utilizing Gradient and Hessian Information," Sixth International Conference on Computational Fluid Dynamics, ICCFD6, St. Petersburg, Russia, July 12-16, 2010.

[46]Rumpfkeil, M. P., Yamazaki, W., and Mavriplis, D. J., "A Dynamic Sampling Method for Kriging and Cokriging Surrogate Models," AIAA Paper, 2011-883, 2011.

[47]Boopathy, K. and Rumpfkeil, M. P., "A Multivariate Interpolation and Regression Enhanced Kriging Surrogate Model," AIAA Paper, 2013-2964.

[48]Kraaijpoel, D. A., *Seismic ray fields and ray field maps: theory and algorithms*, Ph.D. thesis, Universiteit Utrecht, The Netherlands, 2003.

[49]Kraaijpoel, D. and van Leeuwen, T., "Raising the order of multivariate approximation schemes using supplementary derivative data," *Procedia Computer Science*, Vol. 1(1), 2010, pp. 307–316.

[50]Ghanem, R. and Spanos, P. D., *Stochastic Finite Elements: A Spectral Approach (2nd edition)*, New York: Springer, 1991.

[51]Elred, M. S., Webster, C. G., and Constantine, P. G., "Evaluation of Non-Intrusive Approaches for Wiener-Askey Generalized Polynomial Chaos," AIAA Paper, 2008-1892, 2008.

[52]Atkinson, K., *An Introduction to Numerical Analysis*, John Wiley, New York, 1989.

[53]Xiu, D. and Hesthaven, J., "Higher-order collocation methods for differential equations with random inputs," *SIAM Journal on Scientific Computing*, Vol. 27, 2005, pp. 1118–1130.

[54]Downing, D. J., Gardner, R. H., and Hoffman, F. O., "An examination of response-surface methodologies for uncertainty analysis in assessment models," *Technometrics*, Vol. 27, 1985, pp. 151–163.

[55]Isukapalli, S., Roy, A., and Georgopoulos, P., "Efficient sensitivity/uncertainty analysis using the combined stochastic response surface method and automated differentiation: Application to environmental and biological systems," *Risk Analysis*, Vol. 20, 2000, pp. 591–602.

[56]Jaynes, E. and Bretthorst, G., *Probability Theory: The Logic of Science*, University Press Cambridge, Cambridge, UK,, 2003.

[57]Dyn, N., Light, W., and Cheney, E., "Interpolation by piecewise-linear radial basis functions, I," *Journal of Approximation Theory*, Vol. 59, No. 2, 1989, pp. 202 – 223, doi:http://dx.doi.org/10.1016/0021-9045(89)90152-4.

[58]Desai, A., Witteveen, J. A. S., and Sarkar, S., "Uncertainty Quantification of a Nonlinear Aeroelastic System Using Polynomial Chaos Expansion With Constant Phase Interpolation," *Journal of Vibration and Acoustics*, Vol. 135, No. 5.

[59]Buhmann, M., *Radial Basis Functions (1st edn)*, Cambridge University Press: Cambridge, 2005.

[60]Wendland, H., *Scattered Data Approximation (1st edn)*, Cambridge University Press: Cambridge, 2005.

[61]Lachenbruch, P. A. and Mickey, M. R., "Estimation of error rates in discriminant analysis," *Technometrics*, Vol. 10, 1968, pp. 1–11.

[62]Efron, B., "Estimating the error rate of a prediction rule: Some improvements on cross-validation," *Journal of the American Statistical Association*, Vol. 78, 1983, pp. 316–331.

[63]Sheshadri, P., Constantine, P., Gonnet, P., and Parks, G. T., "Sparse Robust Rational Interpolation for Parameter-dependent Aerospace Models," AIAA Paper, 2013-1680, 2013.

[64]Hawkins, D. M., "The Problem of Overfitting," *J. Chem. Inf. Comput. Sci.*, Vol. 44, 2004, pp. 1–12.

[65]Mani, K. and Mavriplis, D. J., "An Unsteady Discrete Adjoint Formulation for Two-Dimensional Flow Problems with Deforming Meshes," AIAA Paper, 2007-60, 2007.

[66]Mani, K. and Mavriplis, D. J., "Discrete Adjoint Based Time-Step Adaptation and Error Reduction in Unsteady Flow Problems," AIAA Paper, 2007-3944, 2007.

[67]Jones, D. R., Schonlau, M., and Welch, W. J., "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, Vol. 13, 1998, pp. 455–492.

A. Training Point Selection Review

The location and number of training points used to construct any surrogate model has a significant effect on its overall accuracy. Training point selection approaches can be broadly classified into domain-based and response-based approaches [15]. In domain-based approaches, training points are chosen based on information available from the domain (e.g., distance between two training points), whereas in response-based approaches, the training points are chosen based on information provided by the surrogate model (e.g., mean squared error approach). The latter approach was developed to enhance the efficiency of the selection process by using information from the existing metamodel. For example, in the response-based approach the user could monitor the progress of the model and choose to stop or extend the selection process. Domain-based selection is based on space-filling concepts that try to fill the domain with training points. It is, in general, not possible for the user to select the number of training points a priori to ensure a given accuracy, due to the non-linearity of most functions of interest. Following is a brief outline of some important domain- and response-based approaches found in the literature.

A.A. Domain-based approaches

A.A.1. Monte Carlo

Monte Carlo (MC) techniques [2] are the simplest of all training point selection methods. Here, a random number generator is used to select training point locations in the domain. A major drawback of MC is the fact that for a small amount of training points large areas of the domain may be left unexplored while others may be sampled densely [7, 14, 16].

A.A.2. Latin hypercube sampling

Latin hypercube sampling (LHS) was proposed by McKay *et al.* [3] for designing computer experiments as an alternative to MC techniques. The basic idea is to divide the range of each design variable into N bins of equal probability, which yields N^M bins in the domain, where M is the dimension of the problem. Subsequently, N training points are generated for each design variable such that no two values lie in the same bin (as shown in the left of

Figure 31). In the middle of Figure 31, one can notice the random fluctuations in the root mean square error (RMSE) of a generic surrogate model built with training points chosen by LHS. In spite of increasing the number of training points, the RMSE does not necessarily decrease, partly because all these points are picked at random. Thus, a superior strategy for training point selection is required to ensure that the RMSE will reduce when one increases the number of training points and this has been the major motivation for this research.

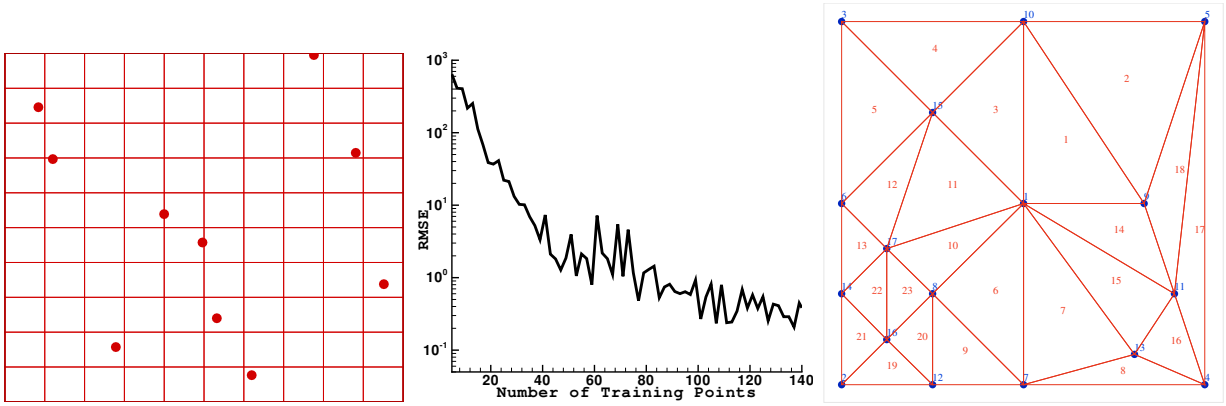

Figure 31. Left: An example of LHS for a two-dimensional domain. Middle: A typical convergence history of a generic surrogate model using LHS or MC. Right: A Delaunay triangulation example.

A.A.3. Delaunay triangulation

Delaunay triangulation is a geometrical method of training point selection, where the domain is divided into hyper-triangles and the training points are chosen at some geometrically significant locations such as the centers of the hyper-triangles and midpoints of the edges as shown in Figure 31 (rightmost). A major drawback of the Delaunay triangulation is that it does not scale well to higher dimensions [46].

A.A.4. Quadratures

Quadrature rules determine the nodes (training points) based on the underlying distribution, e.g., the roots of corresponding orthogonal polynomials. For example, Gauss-Hermite and Gauss-Legendre quadratures have their nodes distributed at the roots of Hermite and Legendre polynomials, respectively. Multi-dimensional quadratures are easily obtained using tensor products of the corresponding one-dimensional quadrature. Though they have been shown to provide optimal convergence [4, 5] for certain applications, they become computationally intractable for higher dimensions as they need $(P+1)^M$ function evaluations in M-dimensions. Though efforts have led to the development of sparse grid quadratures (e.g., Smolyak sparse grids) these still suffer from high computational cost requirements in higher dimensions. Moreover, sparse grids tend to need smoother functions for good results as they

involve extrapolations.

A.B. Response-based approaches

Response-based methods are adaptive (or dynamic) methods that are built progressively starting from an initial number of training points.

A.B.1. Sequential response surface

Jones *et al.* [67] proposed a sequential response surface methodology which starts with a small number of training points and adds additional training points at locations where the standard error is high. During this process, the training point set is updated, the metamodel is reconstructed, and the process of choosing new additional training points is continued until the expected improvement from new training points has become sufficiently small.

A.B.2. Trust-region method

Alexandrov *et al.* [10] proposed a metamodel management framework using a trust-region method for updating metamodels according to the improvement of the objective function during an optimization procedure.

A.B.3. Regional error estimation

Mehmani *et al.* [9] developed a new methodology to quantify the surrogate error in different regions of the domain, which is called the regional error estimation of surrogates (REES) method. The REES method provides a model independent error measure that does not require any additional function evaluations. It works roughly as follows: after segregating the domain into sub-spaces (or regions) variation of the error with sample points (VESP) regression models are constructed to predict the accuracy of the surrogate in each subspace. These regression models are trained by the errors (the mean and the maximum error) evaluated for the intermediate surrogates in an iterative process. At each iteration, the intermediate surrogate is constructed using different subsets of training points and tested over the remaining points. Their results indicate that the REES measure is capable of evaluating the regional performance of a surrogate with reasonable accuracy and that one could use this error estimate to guide a surrogate-building process.

B. Cross Validation

Cross validation [61, 62] is a popular method to estimate a surrogate model's accuracy which does not mandate any additional function evaluations.

B.A. k-fold cross validation

In $k-$fold cross validation the data set is divided into k disjoint subsets and the surrogate model is constructed using the union of $k-1$ subsets and points from the left out subset are used to find the *cross validation error* (CVE). The entire procedure is repeated k times, each time with a different subset left out for validation. Thus, the entire data is used for training as well as validation. The average error across all k trials can be computed as *mean cross validation error* (MCVE) and one can also compute the *maximum cross validation error*.

B.B. Leave-one-out cross validation

If the number of subsets, k, is equal to the total number of training points, N, then the approach is termed *leave-one-out cross validation*. In this case the surrogate model has to be reconstructed N times using $N-1$ training points each time.