Copyright © 2015 Tech Science Press CMES, vol.106, no.5, pp.323-355, 2015

Differing Mapping using Ensemble of Metamodels for Global Variable-fidelity Metamodeling

Qi Zhou[1], Xinyu Shao[1], Ping Jiang[1,2], Longchao Cao[1], Hui Zhou[1] and Leshi Shu[1]

Abstract: Computational simulation models with different fidelity have been widely used in complex systems design. However, running the most accurate simulation models tends to be very time-consuming and can therefore only be used sporadically, while incorporating less accurate, inexpensive models into the design process may result in inaccurate design alternatives. To make a trade-off between high accuracy and low expense, variable fidelity (VF) metamodeling approaches that aim to integrate information from both low fidelity (LF) and high-fidelity (HF) models have gained increasing popularity. In this paper, a Difference Mapping Framework using Ensemble of Metamodels (DMF-EM) for global VF metamodeling is proposed. In DMF-EM, a tuned model is created to bring the low fidelity model as close as possible to high fidelity model. Then, a VF metamodel is obtained by calibrating the tuned model using scaling function that is used to map the difference between the high fidelity model and the tuned model. Since the nature of the scaling function is not a priori, it is fitted using ensemble of metamodels to decrease the risk of adopting inappropriate metamodels. As a demonstration, the proposed approach is compared to existing methods using several numerical cases and two engineering examples. Results illustrate that the proposed DAD-VFM approach is more accurate and robust, that is needed in metamodel-based engineering design problems.

Keywords: Variable fidelity, ensemble, metamodel, scale function, metamodel-based design optimization.

[1] The State Key Laboratory of Digital Manufacturing Equipment and Technology, School of Mechanical Science and Engineering, Huazhong University of Science & Technology, 430074 Wuhan, PR China.

[2] Corresponding author. Email: jiangping@hust.edu.cn; Tel: +86-027-87557742.

1 Introduction

Computational simulation models have been widely used to explore design alternatives during preliminary design phase. In spite of processing power and storage capability of computer have increased dramatically, the time-consuming, high fidelity engineering simulation codes makes it still impractical to rely exclusively on high fidelity simulation models when exploring the design space for an optimum. Just taking Ford Motor Company as an example, it was reported that it takes the company about 36-160 hours to run one crash simulation for a full passenger car [Crombecq, Laermans, and Dhaene (2011)]. A preferable strategy is to adopt global metamodel (also referred to as "surrogate") which can mimics the original model at a considerably reduced computational cost, replacing the computational cost high fidelity simulations during the optimization. There are a lot of commonly used metamodels, such as Polynomial Response Surface (PRS) models [Jiang and Han (2007)], Kriging models [Li, Wu, and Huang (2014); Jiang, Wang, Zhou, and Zhang (2015)], Neural Networks models [Kerh, Lai, Gunaratnam, and Saunders (2008)], Radial Basis Function(RBF) models [Shokri and Dehghan (2012)], Support vector regression (SVR) models [Xiang, Matsumoto, Wang, and Jiang (2011)],etc. A more detailed overview on various metamodeling techniques can refer to Wang and Shan (2007). These metamodeling techniques play a important role in supporting of design and optimization: (1) engineers can gain insight into the system by employing a cheap-to-run metamodel; (2) it renders a better noise filtering capability than gradient-based method; (3)building metamodel makes it easier to detect simulation errors and identify interesting regions as the entire design space is explored and analyzed; (4) building metamodel makes parallel computing and optimization more simplify because it involves running the same simulation at a lot design alternatives [Shan and Wang (2010); Zadeh, Toropov, and Wood (2009); Panda and Manohar (2008); Munck, Moens, Desmet, and Vandepitte (2009)]. It is important to point out that the quality of the metamodel has a profound impact on the computational cost and convergence characteristics of the metamodel-based design optimization. While the quality of the metamodels largely depends on the sample points at which the computer simulation or physical experiments are conducted. Generally, more sample points offer more information of the system, however, at a higher cost [Shan and Wang (2010)]. Less sample points require lower expense, while lead to inaccurate metamodels even distorted metamodels. Hence, conflict of high accuracy and low expense seems to be inevitable in building metamodels.

To ease this problem, variable-fidelity(VF) metamodeling approaches that use lower fidelity (LF) models and a scaling function to approximate the higher fidelity (HF) models have been widespread concerned [Viana, Simpson, Balabanov, and Toropov

2014]. A HF model is one that is able to accurately describe the physical features of the system but with an unaffordable computational expense, e.g., physical experiment, finite element, computational fluid dynamics, etc. A LF model is one that is able to reflect the most prominent characteristics of the system at a considerably less computationally demanding, e.g., numerical empirical formula. Commonly, the scaling function (also called bridge function) can be approximated using local metamodels, e.g., linear regression, first/second Taylor series [Chang, Haftka, Giles,and KAO (1993); Alexandrov and Lewis (2001); Gano, Renaud, and Sanders (2004)] or global stand-alone metamodel, e.g., Kriging metamodels, SVR metamodels [Gano, Renaud, and Sanders (2005); Qian, Seepersad, Joseph, Allen, and Wu (2006); Forrester, Sóbester, and Keane (2007); Han, Görtz, and Zimmermann (2013); Zheng, Shao, Gao, Jiang, and Li (2013)]. The scaling function approximated using local metamodels for VF metamodeing is easy to implement and can achieve a relative high accuracy within an appropriate trust region size, e.g., Chang, Haftka, Giles,and KAO (1993) used a multiplicative scaling approach to correct the response values of LF models to match the HF models. An application of this metamodel was tested on a wing-box model of a high-speed civil transport using an equivalent-plate analysis model and a refined finite-element model as LF and HF model, respectively. Alexandrov and Lewis (2001) integrated first-order additive and multiplicative scaling modeling method with the convergent techniques of nonlinear programming in engineering analysis and design; and have successfully applied this method to a 3-D aerodynamic wing optimization problem and a 2-D airfoil optimization problem, achieving a threefold savings and twofold savings in computing effort, respectively. Gano, Renaud, and Sanders (2004) stated that these Taylor series or polynomial based scaling methods were effective in reducing computational effort, but they were only suitable for local optimization problems. To circumvent this, Gano, Renaud, and Sanders (2005) put forward an adaptive hybrid scaling method by combining both the multiplicative and additive scaling functions using global Kriging metamodel, and have demonstrated the effectiveness and accuracy of this method in the design of high-lift airfoil. Qian, Seepersad, Joseph, Allen, and Wu (2006) proposed a Bayesian approach to integrate LF model and HF simulation values for engineering design. Forrester, Sóbester, and Keane (2007) demonstrated the application of the co-Kriging (extension to Kriging using the multi-responses) for multi-fidelity design using a generic transonic civil aircraft wing optimization problem. Han, Görtz, and Zimmermann (2013) put forward a gradient-enhanced Kriging to form a generalized corrected based method, which was tested on the design of airfoil. Zheng, Shao, Gao, Jiang, and Li (2013) proposed a hybrid VF global metamodeling method, which a RBF base model and a Kriging linear correction were combined to make full use of LF and HF information. Compared with the local VF metamodeling approach, the most obvious advantage of these global VF metamodeling approaches is that they are able to coping with multiple optimum situations sophisticatedly on the entire domain. However, these practices may increase the risk of adopting an inappropriate metamodel that is used in lieu of the scaling function for VF metamodeling when considering the following two realities: (a) the nature of the scaling function is not a priori and the resource to obtain information describing the relationship between input variables and the response values of scaling function is limited (b) the accuracies of constructed metamodel for scaling function depend on the current training data, and a different metamodel type may become more accurate than the selected one with a new available data set.

To overcome the above mentioned shortcomings and improve the prediction accuracy of VF metamodeling approach, this paper proposed a Difference Mapping Framework using Ensemble of Metamodel (DMF-EM) that aims to take advantage of the prediction ability of each stand-alone metamodel for global VF metamodeling. In DMF-EM, a general difference mapping framework (DMF) is developed to integrate the information from both LF model and HF model, where a linear tuning metamodel is created as a start based on the variable-fidelity data, then, the obtained tuning metamodel is taken as a base metamodel and is mapped to the studied HF model using ensemble of metamodels rather than a stand-alone metamodel. The approximation performance of DMF-EM approach is demonstrated using some mathematical and engineering cases, and a rough comparison of DMF-EM approach and other metamodeling techniques for accuracy and robustness performances are made. The main advantages of DMF-EM for VF metamodeling applications are analyzed and summarized.

The rest of this paper is organized as follows. In Section 2, the background of the stand-alone metamodeling techniques and VF metamodeling approaches are presented. Details of the proposed DMF-EM approach are described in Section 3. Test cases composed of data sampling, performance measures, and the corresponding results and comparisons are provided in Section 4, followed by a conclusion and future work in Section 5.

2 Background

2.1 Stand-alone metamodeling methods

Generally, the relationship between a vector of input variables x and corresponding output values Y can be expressed as:

$$Y = \widehat{f}(\boldsymbol{x}, \beta) + \varepsilon \tag{1}$$

where $\widehat{f}(\cdot)$ is the approximation model, β represents the vector of coefficients, εdenotes a stochastic factor. Metamodeling technologies differ with each other as to their choices of approximation models and stochastic process expressions. In this paper, three related metamodeling methods, including Kriging, Radial Basis Function (RBF) and Support Vector Regression (SVR), are described.

2.1.1 Kriging

Kriging metamodels is an interpolative Bayesian metamodeling technique. It was originated from geo-statistical and used by Sacks, Welch, Mitchell, and Wynn (1989) for predicting the unknown response at sample points. Kriging treats the observed response as a combination of a global model and local deviations:

$$\widehat{f}(\boldsymbol{x}) = p(\boldsymbol{x}) + Z(\boldsymbol{x}) \tag{2}$$

where $p(\boldsymbol{x})$ is an known polynomial function, $Z(\boldsymbol{x})$ is the realization of a stochastic process with mean zero and nonzero covariance. The nonzero covariance of $Z(\boldsymbol{x})$ is given by:

$$COV\left(Z\left(x_i\right), Z\left(x_j\right)\right) = \sigma^2 \boldsymbol{R}\left[R\left(x_i, x_j\right)\right] \tag{3}$$

where $\boldsymbol{R}$ is the correlation matrix. $R(x_i, x_j)$ is the correlation function between two sample points x_i and x_j.When the Gaussian correlation function is employed, it can be calculated by:

$$\boldsymbol{R}(\theta) = \exp\left[-\sum_{k=1}^{K} \theta_k \left(x_i^k - x_j^k\right)^2\right] \tag{4}$$

where Kdemotes the dimensions of design space and θ_k are the unknown correlation parameters to be determined. Because Kriging is an interpolative Bayesian metamodeling, the model will have no mean square error (MSE) at all sample points. If the MSE is minimized, the predictor $\widehat{f}(x)$ for unobserved points is expressed as:

$$\widehat{f}(x) = \widehat{\beta} + \boldsymbol{r}^T(x) \boldsymbol{R}^{-1} \left(f - \widehat{\beta} \boldsymbol{p}\right) \tag{5}$$

where f is the column vector of length m that contains the sample data of the responses, and $\boldsymbol{p}$ is a column vector of length m that is filled with ones when $p(\boldsymbol{x})$ is taken as a constant. $r^T(x)$ is the correlation vector between an unobserved point x and the sample points.

$$\boldsymbol{r}^T(x) = \left[R\left(x, x^1\right), R\left(x, x^2\right), \cdots, R\left(x, x^N\right)\right]^T \tag{6}$$

The scalar $\widehat{\beta}$ is estimated using the following equation:

$$\widehat{\beta} = (\boldsymbol{p}^T\boldsymbol{R}^{-1}\boldsymbol{p})^{-1}\boldsymbol{p}^T\boldsymbol{R}^{-1}\boldsymbol{f} \tag{7}$$

The estimated variance of the output model can be calculated by:

$$\widehat{\sigma}^2 = \frac{(\boldsymbol{f}-\widehat{\beta}\boldsymbol{p})^T\boldsymbol{R}^{-1}(\boldsymbol{f}-\widehat{\beta}\boldsymbol{p})}{N} \tag{8}$$

The unknown correlation parameters θ_k are founded using maximum likelihood estimation can be formulated as [Li, Wu, and Huang (2014)]:

$$\begin{aligned} \max\ \ & \Phi(\Theta) = \ -\frac{\left[N\ln(\widehat{\sigma}^2)+\ln|\boldsymbol{R}|\right]}{2} \\ s.t. \ \ & \Theta > 0 \end{aligned} \tag{9}$$

where Θ denotes the vector of θ_k,and both $\widehat{\sigma}$ and $\boldsymbol{R}$ are the function of Θ.

2.1.2 Radial basis function

Radial Basis Function (RBF) is a type of neural network employing a hidden layer of radial units and an output layer of linear units. Let $\boldsymbol{x} = \{x_1, x_2 \cdots x_m\}$ be a set of sampling points generated by the design of experiment (DOE). $\boldsymbol{Y} = \{y_1, y_2 \cdots y_m\}$ are the response function values at data locations. The RBF metamodels can be specified as a line combination of some RBFs with weight coefficients in Eq. (10):

$$\widehat{f}(x) = \sum_{i=1}^{m} \lambda_i \varphi(\|\boldsymbol{x}-\boldsymbol{x}_i\|) \tag{10}$$

where, $\boldsymbol{x}$ is a vector of design variables,m is the number of sample points, x_i is the ith sample point. $\|\boldsymbol{x}-\boldsymbol{x}_i\|$ denotes the Euclidean distance between the design variable and the sample points given by:

$$\|\boldsymbol{x}-\boldsymbol{x}_i\| = \sqrt{(\boldsymbol{x}-\boldsymbol{x}_i)^T(\boldsymbol{x}-\boldsymbol{x}_i)} \tag{11}$$

$\varphi(\cdot)$ is a radial basis function. Commonly used radial basis functions include: (1) bi-harmonic, $\varphi(r) = r$(2) cubic,$\varphi(r) = (r+c)^3$;(3) thin-plate spline,$\varphi(r) = r^2\log(r)$;(4) multiquadric, $\varphi(r) = \sqrt{r^2+c^2}$;(5)inverse-multiquadric,$\varphi(r) = \frac{1}{\sqrt{r^2+c^2}}$;(6) Gaussian, $\varphi(r) = \exp\left(-\alpha r^2\right), \alpha > 0$.$\lambda_i$ are the weight coefficients of the liner combinations can be obtained by:

$$\lambda_i = \left(\Phi^T\Phi+\Lambda\right)^{-1}\Phi^T y_i \tag{12}$$

where,$\Lambda$are all zero except for the regularization parameters along its diagonal. Φ is the design matrix can be expressed as:

$$\Phi = \begin{pmatrix} \varphi(\|x_1 - x_1\|) & \varphi(\|x_1 - x_2\|) & \cdots & \varphi(\|x_1 - x_m\|) \\ \varphi(\|x_2 - x_1\|) & \varphi(\|x_2 - x_2\|) & \cdots & \varphi(\|x_2 - x_m\|) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi(\|x_m - x_1\|) & \varphi(\|x_m - x_2\|) & \cdots & \varphi(\|x_m - x_m\|) \end{pmatrix} \tag{13}$$

2.1.3 Support vector regression

Support Vector Regression (SVR) comes from the theory of support vector machines (SVM), but adds the capability to approximate black box functions. Commonly used SVR is ε-SVR which aims to find a function that has at most ε deviation from the targets of the training inputs [Clarke, Griebsch, and Simpson (2005)]. For the linear regression case, ε-SVR can be depicted as:

$$\widehat{f}(\boldsymbol{x}) = \langle \boldsymbol{w} \cdot \boldsymbol{x} \rangle + b \tag{14}$$

where $\langle \boldsymbol{w} \cdot \boldsymbol{x} \rangle$ is the dot product between w and$x$. Another aims of SVR is to make the $\widehat{f}(\boldsymbol{x})$ to be as flat as possible. Flatness in this sense means a small $\boldsymbol{w}$ in Eq. (14). Hence, we solve the optimization problem described in the following equation:

$$\begin{aligned} \min \quad & \frac{1}{2}|\boldsymbol{w}|^2 \\ s.t. \quad & y_i - \langle \boldsymbol{w} \cdot \boldsymbol{x}_i \rangle - b \le \varepsilon \\ & \langle \boldsymbol{w} \cdot \boldsymbol{x}_i \rangle + b - y_i \le \varepsilon \end{aligned} \tag{15}$$

An assumption made in Eq. (15) is that the prediction error at all sample points are smaller than ε.However, this is not always the case and two slack factors can be incorporated into the original optimization problem to yield a modified formulations [Acar (2010)]:

$$\begin{aligned} \min \quad & \frac{1}{2}|\boldsymbol{w}|^2 + C\sum_{i=1}^{l} \xi_i + \xi_i^* \\ s.t. \quad & y_i - \langle \boldsymbol{w} \cdot \boldsymbol{x}_i \rangle - b \le \varepsilon + \xi_i \\ & \langle \boldsymbol{w} \cdot \boldsymbol{x}_i \rangle + b - y_i \le \varepsilon + \xi_i^* \\ & \xi_i, \xi_i^* \ge 0 \end{aligned} \tag{16}$$

where C is a constant value used to balance the flatness and the degree of the tolerated errors. According to Lagrangian theory and the Karush-Kuhn-Tucker (KKT) condition, the optimization problem can be written in the following dual form:

$$
\begin{aligned}
\max \quad & -\frac{1}{2}\sum_{i,j=1}^{l}(\delta_i-\delta_i^*)(\delta_j-\delta_j^*)\left\langle x_i\cdot x_j\right\rangle \\
& -\varepsilon\sum_{i=1}^{l}(\delta_i+\delta_i^*)+\sum_{i=1}^{l}y_i(\delta_i-\delta_i^*) \\
s.t. \quad & \sum_{i=1}^{l}(\delta_i-\delta_i^*)=0 \\
& (\delta_i-\delta_i^*)\in[0,C]
\end{aligned}
\tag{17}
$$

The weight vector and corresponding linear regression are then obtained through the following expressions:

$$
\boldsymbol{w}=\sum_{i=1}^{l}(\delta_i-\delta_i^*)\boldsymbol{x}_i, \quad \widehat{f}(\boldsymbol{x})=\sum_{i=1}^{l}((\delta_i-\delta_i^*))\left\langle \boldsymbol{x}_i\cdot\boldsymbol{x}_j\right\rangle+b \tag{18}
$$

Another benefit of the above dual form is that nonlinear regression can also be used by replacing the dot product of input vector with kernel functions [Clarke, Griebsch, and Simpson (2005)]. Appling the kernel functions in Eq. (18), we obtain:

$$
\begin{aligned}
\max \quad & -\frac{1}{2}\sum_{i,j=1}^{l}(\delta_i-\delta_i^*)(\delta_j-\delta_j^*)k(\boldsymbol{x}_i\cdot\boldsymbol{x}_j) \\
& -\varepsilon\sum_{i=1}^{l}(\delta_i+\delta_i^*)+\sum_{i=1}^{l}y_i(\delta_i-\delta_i^*) \\
s.t. \quad & \sum_{i=1}^{l}(\delta_i-\delta_i^*)=0 \\
& (\delta_i-\delta_i^*)\in[0,C]
\end{aligned}
\tag{19}
$$

where $k(\boldsymbol{x}_i\cdot\boldsymbol{x}_j)$ denotes the kernel functions. Commonly used kernel functions include: polynomial, Gaussian, Sigmoid and Inhomogeneous polynomial. Then the SVR metamodels for nonlinear regression becomes:

$$
\widehat{f}(x)=\sum_{i=1}^{l}(\delta_i-\delta_i^*)k(\boldsymbol{x}_i\cdot\boldsymbol{x}_j)+b \tag{20}
$$

2.2 Variable fidelity metamodeling technology

The VF metamodeling technology is based on the assumption that, apart from a HF model that is sufficiently accurate but requires expensive computational cost, there is another one that is less accurate but is considerably less computationally demanding [Simpson, Toropov, Balabanov, and Viana (2008)]. Such LF can be obtained by several ways: (1) simplifying the analysis model (e.g. by using a coarse finite element mesh instead of a refined finite element mesh, etc.); (2) simplifying the modeling concept (e.g. by using a two-dimensional (2D) model instead of a three-dimensional (3D) model); (3) simplifying the mathematical or physical description (e.g. by using the Euler non-cohesive equations instead of the Navier-Stokes viscous Newton equations). A LF model must be able to describe most prominent physical features of the design product at a considerably reduced computational cost. In the modeling process, a LF model is tuned (corrected) using the HF model response from a suitable size of design experiments. In this way, the VF metamodels can make use of the advantages of both LF models and HF models, i.e., LF models are used to reduce the computational cost, while HF models are used to guarantee the accuracy.

Generally, the VF metamodel based on the interaction of HF model and LF model can be expressed as follows:

$$\widehat{F}(\boldsymbol{x},\boldsymbol{a}) \equiv \widehat{F}(f_l(\boldsymbol{x}),\boldsymbol{a}) \approx F(\boldsymbol{x}) \tag{21}$$

where x is the design vector, $\widehat{F}(\boldsymbol{x},\boldsymbol{a})$ denotes the VF metamodel that is used to replace the actual HF model,$f_l(\boldsymbol{x})$ represents the response of the LF model and $F(\boldsymbol{x})$ represents the true response of the HF metamodel , $\boldsymbol{a}$ is a vector of tuning parameters used for minimizing the discrepancy between the LF and HF models. From the above definition, the VF metamodel $\widehat{F}(\boldsymbol{x},\boldsymbol{a})$ tends to approach the high accuracy of the HF model but at a considerably less computational effort.

Three types of the LF model tuning have been suggested by Toropov and Markine (1996) (1) linear and multiplicative metamodel with two tuning parameters; (2) multiplicative and additive correction functions; (3) use of model inputs as tuning parameters. More details can refer to the literature, e.g., Viana, Simpson, Balabanov, and Toropov (2014), Toropov and Markine (1996), Zadeh, Toropov, and Wood (2009).

3 Proposed approach

The goal of the proposed approach is to obtain a good estimate of the output response by integrating the information from both LF model and HF model. Based on the variable-fidelity data, a linear tuning metamodel is created as a start. Then, the obtained tuning metamodel is taken as a base metamodel and is mapped to the studied HF model using scaling function. The scaling function is fitted using ensemble of metamodels rather than a stand-alone metamodel. Finally, The obtained mapped base metamodel can be used to mimic the behavior of the computational expensive HF model if its accuracy achieved.

The formulation of the proposed VF metamodeling approach can be stated as follows:

$$\widehat{y}_v(\boldsymbol{x}) = \widehat{y}_{l,tuned}(\boldsymbol{x}) + c(\boldsymbol{x}) = a_0 + a_1 y_l(\boldsymbol{x}) + c(\boldsymbol{x}) \tag{22}$$

where a_0 and a_1 are two running parameters that defined the tuned LF metamodel $\widehat{y}_{l,tuned}(\boldsymbol{x}) = a_0 + a_1 y_l(\boldsymbol{x})$, $c(\boldsymbol{x})$ denotes the scaling function which is used to map the difference between the tuned LF metamodel $\widehat{y}_{l,tuned}(\boldsymbol{x})$ and the true HF model $y_h(\boldsymbol{x})$,$y_l(\boldsymbol{x})$ can be either the original LF model or its metamolel $\widehat{y}_l(\boldsymbol{x})$, however, for simplicity of the notation and presenting a more generalized description, the LF metamodel $\widehat{y}_l(\boldsymbol{x})$ is used throughout this paper.

The remainder of this section describes the proposed VF metamodeling approach in more details as follows: Section 3.1 gives the procedure for tuning the LF metamodel, Section 3.2 gives the procedure for difference mapping using scaling function fitted with ensemble of metamodels and Section 3.3 presents a step-by-step description of the proposed approach.

3.1 Tuning the LF metamodel

Two tuning parameters a_0 and a_1 are adopted to help bring the LF model as close as possible to the HF model. Several approaches were proposed to determine these two tuning parameters in previous work, e.g., cross validation [Hastie and Tibshirani, (2001)] and maximum likelihood estimation [Gano, Renaud, and Sanders (2004)]. In this work, the least square (LS) method according to its convenience and easily application ability, together with bounds constraints are selected to identified the tuning parameters a_0 and a_1. The optimization formulation is as follows:

$$\begin{aligned} &\min : L(a_0, a_1) = \sum_{i=1}^{m} \left[y_h(x_i) - (a_0 + a_1 \widehat{y}_l(x_i)) \right]^2 \\ &s.t. \quad l_0 \le a_0 \le u_0,\ l_1 \le a_1 \le u_1 \end{aligned} \tag{23}$$

where $L(a_0,a_1)$ stands for the loss function in a square sense; $x_i (i = 1,\ldots,m)$ are the sample points of HF model. The bounds (l_0,u_0), (l_1,u_1) posed on tuning parameters a_0 and a_1 represent the prior knowledge of the global constant bias and multiplicative scaling between LF and HF models. This is helpful to avoid the overfitting issue within regular linear when there is no enough data. Different types of tuning parameters a_0 and a_1 are available, e.g., constant terms, linear terms, and quadratic terms. To simplify the modeling procedure, the tuning parameters a_0 and a_1 are assumed to be unknown but fixed as constant terms in this paper. Note that using a_1 will help better preserve the "profile" of a LF model and using a_0 will help satisfy the assumption of "zero-mean" priors posed on the bias function $c(\boldsymbol{x})$, especially if a global bias exists between LF and HF models.

When the optimum a_0^* and a_1^* are obtained by solving Eq. (3), a tuned based metamodel $y_{l,tuned}$ can be expressed as follows:

$$y_{l,tuned} = a_0^* + a_1^* \widehat{y}_l(\boldsymbol{x}) \tag{24}$$

where $\widehat{y}_l(\boldsymbol{x})$ denotes the LF metamodel, $y_{l,tuned}$ is the tuned LF metamodel which is used as a base metamodel in the scaling process.

3.2 Difference mapping using ensemble of metamodels

The obtained tuned LF metamodel in Eq. (24) is not enough to approximate the true HF model when data are far from sufficient to explore the behavior of the HF model performance [Xiong, Chen, and Tsui (2008)]. A scaling function should better be adopted to account for the remaining discrepancy between the HF simulations data and the scaled LF metamodel. Suppose that the HF sampling set is $\boldsymbol{X}_H = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$, which consists of $m$evaluated experiments in a K-dimentional design space, and its corresponding HF response is $\boldsymbol{Y}_h = \{y_1, y_2, \ldots, y_m\}$.The remaining discrepancy $\boldsymbol{C} = \{c_1, c_2, \ldots, c_m\}$ between the true HF response and the obtained base metamodel in the tuning process can be represented as:

$$c(x_i) = f_h(x_i) - (a_0^* + a_1^* \widehat{y}_l(x_i)), \;\; i = 1, 2, \ldots m \tag{25}$$

Since the expression or internal structure for the HF sampling set $\boldsymbol{X}_H$ and remaining discrepancy $\boldsymbol{C}$ is unknown, different stand-alone metamodels are used to fit the scaling functions, e.g., Xiong, Chen, and Tsui (2008) constructed a common Kriging metamodel for the scaling function. Zheng, Shao, Gao, Jiang, and Qiu (2014) put forward a SVR-based linear scaling function to accomplish the difference mapping. Main drawbacks of these practices are that the nature of the scaling function is not a priori and the accuracies of the constructed stand-alone metamodel largely depend on the current training sampling set and/or the problem properties at hand, so it cannot be guaranteed that the selected metamodel will always perform the best as another training data available or problems are changed. Therefore, as an alternative to using a stand-alone metamodel to fit the scaling function, it would be beneficial to in lieu of the scaling function using ensemble of metamodel, which can take advantage of the prediction ability of each separate stand-alone metamodel. In this paper, three different stand-alone metamodeling techniques are considered here, they include: Kriging, RBF and SVR. The ensemble of metamodel can be constructed by using a weighted average of different metamodels, hence, the metamodel for remaining discrepancy is defined as:

$$\widehat{c}(\boldsymbol{x}) = \sum_{j=1}^{3} w_j \widehat{c}_j(\boldsymbol{x}) \tag{26}$$

where, $\widehat{c}(\boldsymbol{x})$ is the predicted remaining discrepancy obtained from the ensemble metamodel, $\widehat{c}_j(\boldsymbol{x})(j = 1,2,3)$ are the predicted remaining discrepancy obtained using the Kriging, RBF and SVR metamodels, respectively. For the Kriging metamodels, $\widehat{c_1}(\boldsymbol{x})$ is the best linear unbiased predictors which minimize the MSE:

$$MSE[\widehat{c_1}(\boldsymbol{x})] = E\left[\boldsymbol{\lambda}^T(\boldsymbol{x})C_1 - C_1(\boldsymbol{x})\right]^2 \tag{27}$$

With the unbiasedness constraints as follows:

$$E\left[\boldsymbol{\lambda}^T(\boldsymbol{x})C_1\right] = E(C_1(\boldsymbol{x})) \tag{28}$$

Once the correlation functions $\boldsymbol{R}(\cdot)$ and $\boldsymbol{r}^T(x) = \left\{R(x,x^1), R(x,x^2), \ldots, R(x,x^n)\right\}$ have been selected and the vector of parameter Θ obtained by solving Eq. (9), $\widehat{c_1}(\boldsymbol{x})$ can be obtained after several deductions:

$$\begin{aligned} &\widehat{c_1}(\boldsymbol{x}) = \widehat{\beta} + (\boldsymbol{r}^*)^T(\boldsymbol{R}^*)^{-1}(\boldsymbol{Y}_h - (a_0^* + a_1^*\boldsymbol{Y}_l) - \widehat{\beta}\boldsymbol{p}) \\ &\widehat{\beta} = (\boldsymbol{p}^T\boldsymbol{R}^{-1}\boldsymbol{p})^{-1}\boldsymbol{p}^T\boldsymbol{R}^{-1}(\boldsymbol{Y}_h - (a_0^* + a_1^*\boldsymbol{Y}_l)) \end{aligned} \tag{29}$$

For the RBF metamodels, when substituting the optimum tuning parameters a_0^* and a_1^* in Eq. (12), the optimal weight coefficients λ_i^* can be obtained by:

$$\lambda_i^* = \left(\Phi^T\Phi + \Lambda\right)^{-1}\Phi^T\left(f_h(x_i) - (a_0^* + a_1^*\widehat{y}_l(x_i))\right) \tag{30}$$

In this study, popular Gaussian function is selected as the radial basis function. With the optimal weight coefficients of the liner combinations λ_i^* are obtained by Eq. (30), $\widehat{c_2}(\boldsymbol{x})$ can be obtained by the following expression:

$$\widehat{c_2}(\boldsymbol{x}) = \sum_{i=1}^{m} \lambda_i^* \varphi(\|\boldsymbol{x} - \boldsymbol{x}_i\|) \tag{31}$$

To construct SVR metamodels $\widehat{c_3}(\boldsymbol{x})$, the Gaussian kernel function is adopted as its advantages of fewer parameters to be set and an excellent overall performance. The following quadratic programming should be solved to obtain the vector $\boldsymbol{\delta}$ and $\boldsymbol{\delta}^*$:

$$\begin{aligned}
\max \quad & -\frac{1}{2}\sum_{i,j=1}^{l}(\delta_i-\delta_i^*)(\delta_j-\delta_j^*)\exp(\frac{\|x_i-x_j\|}{2\sigma^2}) \\
& -\varepsilon\sum_{i=1}^{l}(\delta_i+\delta_i^*)+\sum_{i=1}^{l}(f_h(x_i)-(a_0^*+a_1^*\widehat{y}_l(x_i)))(\delta_i-\delta_i^*) \\
s.t. \quad & \sum_{i=1}^{m}(\delta_i-\delta_i^*)=0 \\
& (\delta_i-\delta_i^*)\in[0,C] \\
& i,j=1,2,\ldots m
\end{aligned} \tag{32}$$

When the optimal $\boldsymbol{\delta}$ and $\boldsymbol{\delta}^*$ are obtained, $\widehat{c_3}(\boldsymbol{x})$ can be obtained through:

$$\widehat{c_3}(\boldsymbol{x})=\sum_{i=1}^{m}(\delta_i-\delta_i^*)\exp(\frac{\|x_i-x_j\|}{2\sigma^2})+b \quad ; \; i,j=1,2,\ldots m \tag{33}$$

In Eq. (26), $w_j(j=1,2,3)$ are weight factors that determine the relative contribution of the three metamodels in ensemble. To obtain unbiased response estimations, a constraint is posed on weight factors as follows:

$$\sum_{j=1}^{3} w_j = 1 \tag{34}$$

The weight factors can be determined such that the metamodel with high accuracy have large weight factors and vice versa [Acar 2010; Zhou, Ma, and Li (2011)]. In this paper, the weight factors are determined by minimizing the generalized mean square Leave–one-out (LOO) errors ($GMSE_{LOO}$) of the ensemble metamodels. LOO method is one of the cross validation methods that can be used for assessing the accuracy of a metamodel. The basic processes of obtaining the $GMSE_{LOO}$ can be divided into four steps. Step 1: for a given HF sampling set $\boldsymbol{X}_H=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_m\}$, remove one of the sampling points out. Step 2: use the remaining sampling points $\boldsymbol{X}_{H,-i}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_{i-1},\boldsymbol{x}_{i+1},\ldots,\boldsymbol{x}_m\}$ to build the ensemble of metamodel, which is used to predict the response value for the omitted sample. Step 3: calculate the square difference between the predicted values and the accurate value for the omitted sample. Step 4: repeat Step 1 to Step 3 until all sample points in the given sampling set are considered. With this method, no additional simulation runs are needed for error calculations.

The $GMSE_{LOO}$ for the ensemble of metamodel can be expressed as:

$$GMSE_{LOO}=\sum_{i=1}^{m}(\widehat{y}_{en,-i}(x_i)-y(x_i))^2/m \tag{35}$$

where $y(x_i)$ denotes the actual response value at x_i, $\widehat{y}_{en,-i}(x_i)$ denotes the prediction of the response value for x_i using the ensemble of metamodel created based on the current sampling set with the sample point x_i moved out.

Combining the weighted-sum formulations of the ensemble metamodels in Eq. (26) and error prediction metric in Eq. (35), a constrained minimization problem is established to select the optimum weight factors. It can be expressed as follows:

$$\begin{aligned}
&find \;\; w_j \\
&\min \;\; GMSE_{LOO}(w_j) = \sum_{i=1}^{m} \Big(\sum_{j=1}^{3} w_j \widehat{c}_j(x_i) - c(x_i)\Big)^2 / m \\
&s.t. \;\; \sum_{i=1}^{3} w_j = 1
\end{aligned} \tag{36}$$

Since the optimization problem in Eq. (36) is not necessary convex, evolutionary algorithms are preferred to be selected to solve this problem. In this study, genetic algorithms [Coello (2000)] are used and the constraint is processed using penalty function.

With the obtained optimal weight factors, the formulation of the proposed VF metamodeling approach is becomes:

$$\widehat{y}_v(\boldsymbol{x}) = \widehat{y}_{l,tuned}(\boldsymbol{x}) + c(\boldsymbol{x}) = a_0^* + a_1^* \widehat{y}_l(\boldsymbol{x}) + \sum_{j=1}^{3} w_j^* \widehat{c}_j(\boldsymbol{x}) \tag{37}$$

3.3 Steps for the proposed DMF-EM approach

The flowchart of the proposed DMF-EM is demonstrated in Fig. 1. The details steps are as follows:

Step 1: Generate two sampling sets $\boldsymbol{X}_L$ and $\boldsymbol{X}_H$. One set ($\boldsymbol{X}_L$) with a large number of sample points is used to obtain response values for LF model, the other one ($\boldsymbol{X}_H$) with a significantly smaller set is used to evaluate the HF model.

Step 2: Run LF simulation model at the $\boldsymbol{X}_L$ to obtain the LF response values.

Step 3: Run HF simulation model at the $\boldsymbol{X}_H$ to obtain the HF response values.

Step 4: Build metamodel for the LF model based on the sampled LF data.

Step 5: Tune the LF metamodel to pull the LF metamodel to the HF real response values using Eq. (23).

Step 6: Construct the scaling function model using ensemble of metamodels. This step can be divided into several steps to demonstrate the approach in more details. The details of each step are given next and the entire procedures are demonstrated in the dashed box of Fig. 1.

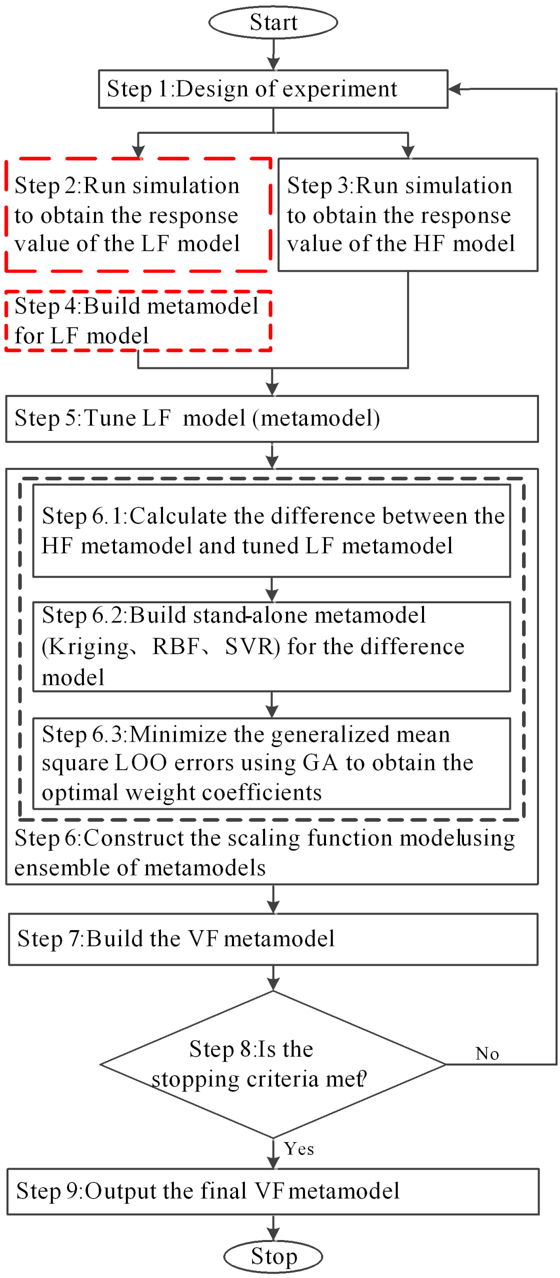


Figure 1: Flowchart for the proposed approach.

Step 6.1: Calculate the discrepancies between the HF model and tuned LF metamodel for sample points using Eq. (25).

Step 6.2: Build three stand-alone metamodels (Kriging, RBF, SVR) for the difference model.

Step 6.3: Minimize the generalized mean square LOO errors to find the optimum weight factors by solving the optimization problem in Eq. (36).

Step 7: Build the VF metamodel using Eq. (37).

Step 8: Check the stopping criterion. Check whether the desired level of accuracy is achieved or not. If yes, continue to Step 9, otherwise, go back to Step 1. In this paper, Root mean square error (RMSE) is used as the accuracy metric.

Step 9: Output the final VF metamodel.

It should be point out that apart from either of these circumstances: (a) running LF simulation is expensive (b) the relationships between the input variables and corresponding output values cannot be express explicitly in LF model, Step 2 and Step 4 can be omitted, i.e., the LF model can be directly used for VF modeling without fitting a metamodel to replace it.

4 Examples and results

To test the performance of the proposed modeling method, overall seven example problems are considered. In the first five, the responses of the LF models and HF models are described by analytic functions that are well-known numerical test problems with two or more design parameters. The last two problems are engineering cases, where in the first engineering case the output responses of LF model and HF model are obtained using the empirical formulas, whereas in the other one the true responses of the LF model and HF model are obtained from finite element analysis (FEA), which aims to demonstrate that the proposed modeling method is applicable to complex engineering problems. For comparison, we model all tests with other three different variable-fidelity metamodeling methods: (1) variable-fidelity metamodeling combining different mapping framework with Kriging (DFM-Kriging), (2) variable-fidelity metamodeling combining different mapping framework with RBF (DFM-RBF), (3) variable-fidelity metamodeling combining different mapping framework with SVR (DFM-SVR) [Zheng, Shao, Gao, Jiang, and Qiu (2014)].

4.1 Numerical examples and results discussion

4.1.1 Numerical examples description

The numerical test problems are described by the following analytic functions. In all tests, y_h represents the HF metamodel, which is needed to be approximated and

y_l represents the LF one.

- Six-hump Camel-back function (SC):

$$
\begin{aligned}
&f(x_1,x_2) = 4x_1^2 - 2.1x_1^4 + x_1^6/3 + x_1x_2 - 4x_2^2 + 4x_2^4;\\
&y_h = f(x_1,x_2);\\
&y_l = f(0.7x_1, 0.7x_1) + x_1x_2 + \sin(x_1) + 2x_2 + \sin(x_2) + x_1;\\
&x_1 \in [-2,\ 2], x_2 \in [-2,\ 2]
\end{aligned} \tag{38}
$$

- Himmelblau function(HM) :

$$
\begin{aligned}
&f(x_1,x_2) = (x_1^2 + x_2 - 11)^2 + (x_2^2 + x_1 - 7)^2;\\
&y_h = f(x_1,x_2);\\
&y_l = f(0.5x_1, 0.8x_2) + x_2^3 - (x_1+1)^2;\\
&x_1 \in [-3,3], x_2 \in [-3,3];
\end{aligned} \tag{39}
$$

- 2-dimensional Rosenbrock function (2D Rosenbrock) :

$$
\begin{aligned}
&f(x_1,x_2) = 100(x_1^2 - x_2)^2 + (x_1-1)^2;\\
&y_h = f(x_1,x_2);\\
&y_l = 70(x_1^2 - x_2)^2 + 0.7(x_1-1)^2;\\
&x_1 \in [-2.048, 2.048], x_2 \in [-2.048, 2.048];
\end{aligned} \tag{40}
$$

- 3-dimensional Rosenbrock function (3D Rosenbrock) :

$$
\begin{aligned}
&f(x_1,x_2,x_3) = \sum_{i=1}^{2} [100(x_i^2 - x_{i+1})^2 + (x_i-1)^2];\\
&y_h = f(x_1,x_2,x_3);\\
&y_l = 90(x_1^2 - x_2)^2 + 1.1(x_3-1)^2 + 100(x_2^2 - x_3)^2 + (x_2-1)^2;\\
&x_i \in [-2.048, 2.048],\ i = 1,2,3;
\end{aligned} \tag{41}
$$

- Dixon & Price function (DP):

$$
\begin{aligned}
&f(x_1,\ldots,x_4) = (x_1-1)^2 + \sum_{i=2}^{n} i(2x_i^2 - x_{i-1})^2;\\
&y_h = f(x_1,\ldots,x_4);\\
&y_l = (x_1-1)^2 + \sum_{i=2}^{n} i(2x_i^2 - 0.75x_{i-1})^2;\\
&x_i \in [-10,\ 10], i = 1,\ldots,4;\\
&n = 4
\end{aligned} \tag{42}
$$

4.1.2 Data sampling

The goal of the data sampling is to minimize the influence of errors in physical experiments to the response functions, while allowing the designers to build metamodels more efficiently [Zadeh, Toropov, and Wood (2009).]. In this work, the optimal Latin Hypercube Sampling (OLHS), proposed by Jin, Chen, and Simpson (2001), is adopted to generate sample points. Two different sample sizes, small size (m=10) and large size (m=50), are generated. It should be pointed out that the term large sample size for HF model is still in the scope of small size compared with that in LF models.

4.1.3 Accuracy and robustness measures

Three different accuracy metrics are adopted to assess the accuracies of each metamodel: (1) relative root mean square error (RRMSE) (2) relative maximum absolute error (RMAE) (3) coefficient of multiple determinations (R^2). RRMSE and R^2 reveal the global accuracy of the metamodel, while the RMAE reflects the local accuracy of the metamodel. The lower the value of RRMSE/RMAE, the more accurate of the metamodel, while larger value of R^2 indicates a more accurate metamodel. Expressions of these three accuracy metrics are defined as follows:

$$
\begin{aligned}
&RRMSE = \frac{1}{STD}\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i-\hat{y}_i)^2} \\
&RMAE = \frac{1}{STD}\max|y_i-\hat{y}_i|, i=1,\ldots,N \\
&R^2 = \frac{(N\sum_{i=1}^{N} y_i\widehat{y}_i - \sum_{i=1}^{N} y_i \sum_{i=1}^{N}\widehat{y}_i)^2}{(N\sum_{i=1}^{N}\widehat{y}_i^2-(\sum_{i=1}^{N}\widehat{y}_i)^2)(N\sum_{i=1}^{N}y_i^2)-(\sum_{i=1}^{N}y_i)^2)} \\
&STD = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(y_i-\bar{y}_i)^2}
\end{aligned}
\tag{43}
$$

where N represents the total number of test points; y_i is the actual response value at test points; $\hat{y}_i$ is the predicted response at test points; $\bar{y}_i$ and STD are the mean and standard deviation of all observed response values, respectively.

Apart from the above accuracy measures, robustness performance of each VF metamodeling approach is taken into consideration. The robustness of a metamodel refers to its capability of achieving similar accuracies for different problems and different sample size [Gao, Xiao, Shao, Jiang, Nie, and Qiu (2012)]. In this work, the standard deviation of the above mentioned error metrics calculated according to Eq. (43), are selected to measure the robustness performance of different VF metamodeling approaches.

4.1.4 Results and discussion

To calculate the three accuracy metrics, additional 1600 randomly validation points are used in each numerical case. The accuracy results of different metamodeling approaches are summarized in Tab. 1. In Tab. 1, the best results for each model type are printed in bold, while the worst results for each model type are printed in italic. Two intuitive conclusions can be drawn from Tab. 1, i.e., (1) The metamodels constructed by DMF-EM nerve be the worst one among the four VF metamodeling approaches for all the numerical cases at two different sample sizes, while other three VF metamodel are the worst at least for one case, which indicates that the proposed DMF-EM metamodeling can decrease the risk of adopting an inappropriate metamodel. (2) For most test cases, metamodels constructed by DMF-EM are demonstrated to be the best in terms of both local and global accuracy, especially at small sample size.

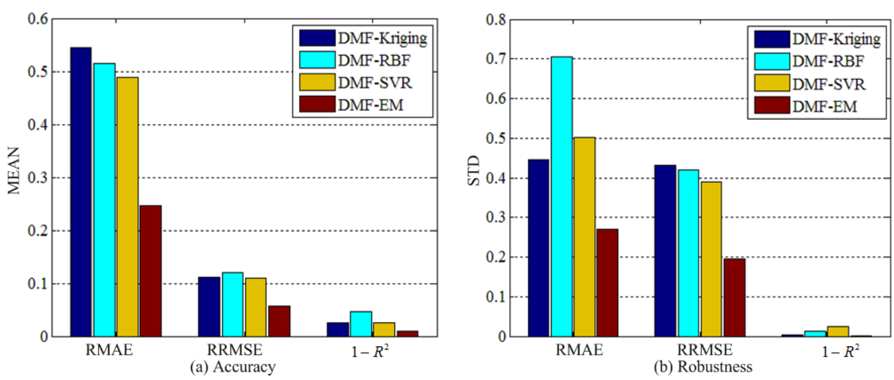


Figure 2: Overall performance for different VF metamodeling approaches (a) accuracy (b) robustness.

Fig. 2 demonstrates the mean and standard deviation of the three accuracy metrics for all the metamodels under small and large sample sizes. As seen in Fig. 2 (a), it can be observed that the accuracy of DMF-EM is the best in all the three accuracy metrics. DMF-Kriging performs slightly better than DMF-RBF and DMF-SVR in RRMSE and R^2, while its RMAE value is the largest. The average value of RRMSE and $1-R^2$ is the largest for DMF-RBF, which indicates the poorest global performance among four VF metamodeling approaches. As illustrated in Fig. 2 (b),

Table 1: Summary of the accuracy results for different sample size.

Tests	Accuracy criteria	Sample size (m=10)				Sample size (m=50)			
		DMF-Kriging	DMF-RBF	DMF-SVR	DMF-EM	DMF-Kriging	DMF-RBF	DMF-SVR	DMF-EM
SC	RMAE	1.3411	***2.5313***	1.7077	**0.5868**	***0.4242***	0.1160	0.3289	**0.2258**
	RRMSE	0.3850	***0.6582***	0.5089	**0.1794**	***0.0562***	0.0345	0.0477	**0.0325**
	R^2	0.8684	***0.6066***	0.8269	**0.9693**	***0.9973***	0.9988	0.9978	**0.999**
HM	RMAE	***1.1437***	1.1073	0.9918	**0.9313**	0.2979	0.1981	***0.3187***	**0.1966**
	RRMSE	***0.3148***	0.2692	0.2445	**0.2398**	0.0321	0.0253	***0.0431***	**0.0206**
	R^2	***0.9033***	0.9413	0.9426	**0.9435**	0.9990	0.9994	***0.9981***	**0.9996**
2D Rosenbrock	RMAE	***1.2015***	0.5710	0.9633	**8.3E-07**	***0.4421***	0.0292	0.1425	**2.4E-08**
	RRMSE	***0.1700***	0.0974	0.1488	**1.1E-07**	***0.0396***	0.0027	0.0132	**2.1E-09**
	R^2	***0.9783***	0.9920	0.985	**1.0000**	***0.9990***	1.0000	0.9999	**1.0000**
3D Rosenbrock	RMAE	0.2602	0.2713	***0.2689***	**0.2594**	***0.2275***	0.2003	**0.0933**	0.1808
	RRMSE	0.0600	0.0544	***0.0636***	**0.0587**	***0.0365***	0.0337	**0.0139**	0.0289
	R^2	0.9982	0.9983	***0.9982***	**0.9982**	***0.9991***	0.9991	**0.9998**	0.9994
DP	RMAE	0.0593	***0.0666***	0.0593	**0.0582**	0.0490	***0.0635***	**0.0057**	0.0272
	RRMSE	0.0132	***0.0141***	0.0130	**0.0126**	0.0075	***0.0131***	**0.0007**	0.0056
	R^2	0.9998	***0.9998***	0.9998	**0.9998**	0.9999	***0.9998***	**1.0000**	1.0000

the robustness of the DMF-EM is also the best in all the three accuracy metrics. The robust of DMF-SVR and DMF-Kriging are close. DMF-Kriging is slightly better than DMF-SVR in RMSE and R^2, while DMF-SVR is a little better than DMF-Kriging in RRMSE. DMF-RBF presents the poorest robustness in RMAE, but the robustness of its global performance is somewhat in between the DMF-SVR and DMF-Kriging. In summary, the performance of DMF-EM is the best in terms of both average accuracy and robustness for all the test numerical cases at small and large sample sizes.

A deeper demonstration on the accuracy and robustness of each metamodeling performance under different sample sizes (small and large) are presented in Fig. 3. It is observed in Fig. 3 that for both small and large sample sets, the average accuracy and robustness of DMF-EM are much better than other three VF metamodeling methods in three accuracy metrics expect its slightly lower accuracy than DMF-RBF and DMF-SVR in RMAE under large sample set. With the small sample set, DMF-Kriging shows a better performance than DMF-RBF and DMF-SVR. However, this conclusion is as opposed to the observation under large sample set that the DMF-Kriging is the worst in terms of both average accuracy and robustness. The performance of DMF-RBF and DMF-SVR is similar under different sample sizes.

Through the comparison of numerical cases results under small and large sample sizes, it can be easily found that when the sample set is increased from small size to large size, the average accuracy and robustness of all the VF metamodeling are increased significantly, especially for DMF-RBF and DMF-SVR. A more important observation is that the average accuracy and robustness of DMF-EM are high in spite of the utilization of small sample set.

Besides the 3D surface of the actual LF and HF models, the contour of difference between them is plotted in Fig. 4. The metamodel for SC obtained using different VF modeling methods under small sample size are also presented in Fig. 4. It is illustrated that DMF-EM reflects the behavior of the actual SC function most accuracy in the whole design space. Though, other three VF metamodeling methods can capture the general trend, there exist large errors in some local regions.

4.2 Engineering case 1: design of a curved stepped beam

The first engineering example originally from Balabanov and Venter (2004) is a design of a curved stepped beam, which is the lower left portion of a 10-ton crane hood, shown in the Fig.5 (a). In this work, the curved beam, with a force $P = 20,000lb$ $(88.97kN)$ acting at the tip of the beam is divided into five sections depicted in Fig.5 (b). Section numbering started from the fixed end of each section. The design variables are the height of each segment $(h_i, i = 1, 2, \ldots, 5)$ and

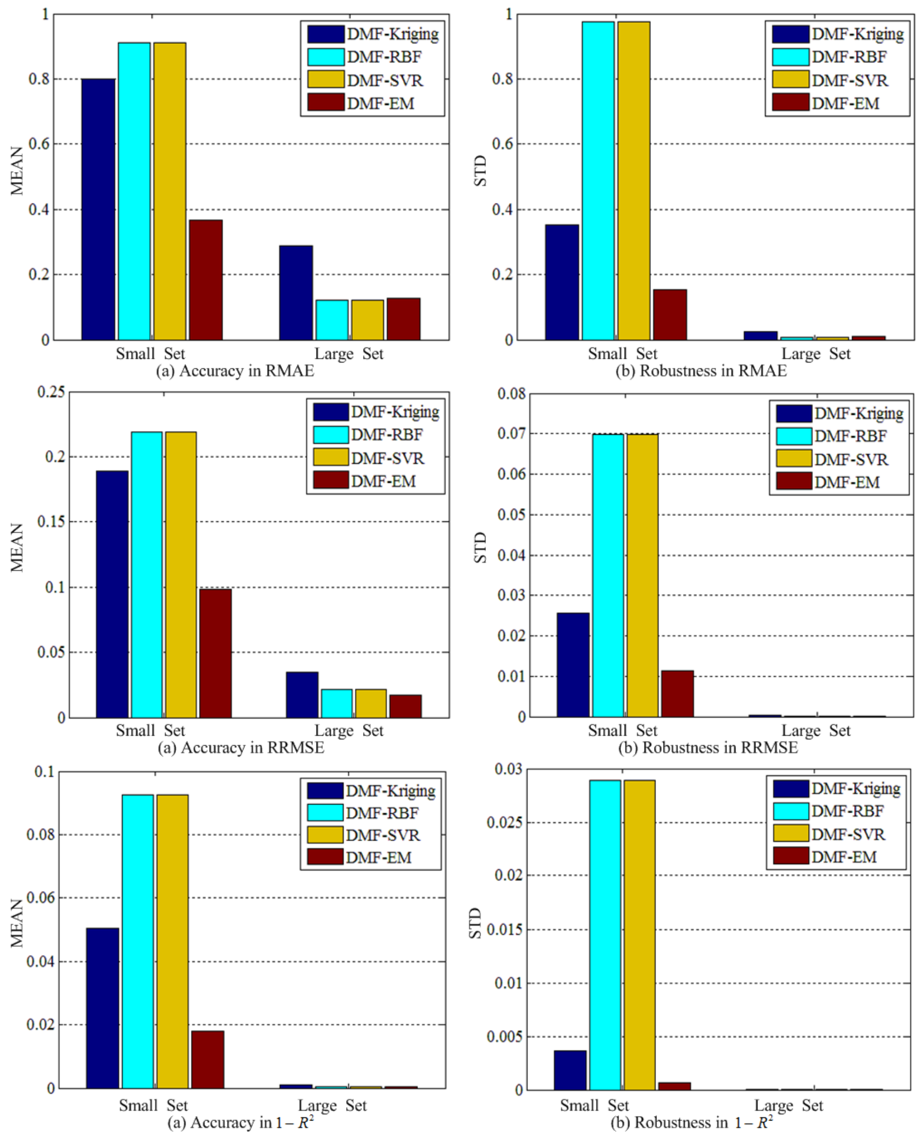


Figure 3: Performances under different sample sizes. (a) Accuracy in RMAE (b) Robustness in RMAE (c) Accuracy in RRMSE (d) Robustness in RRMSE (e) Accuracy in R^2 (f) Robustness in R^2.

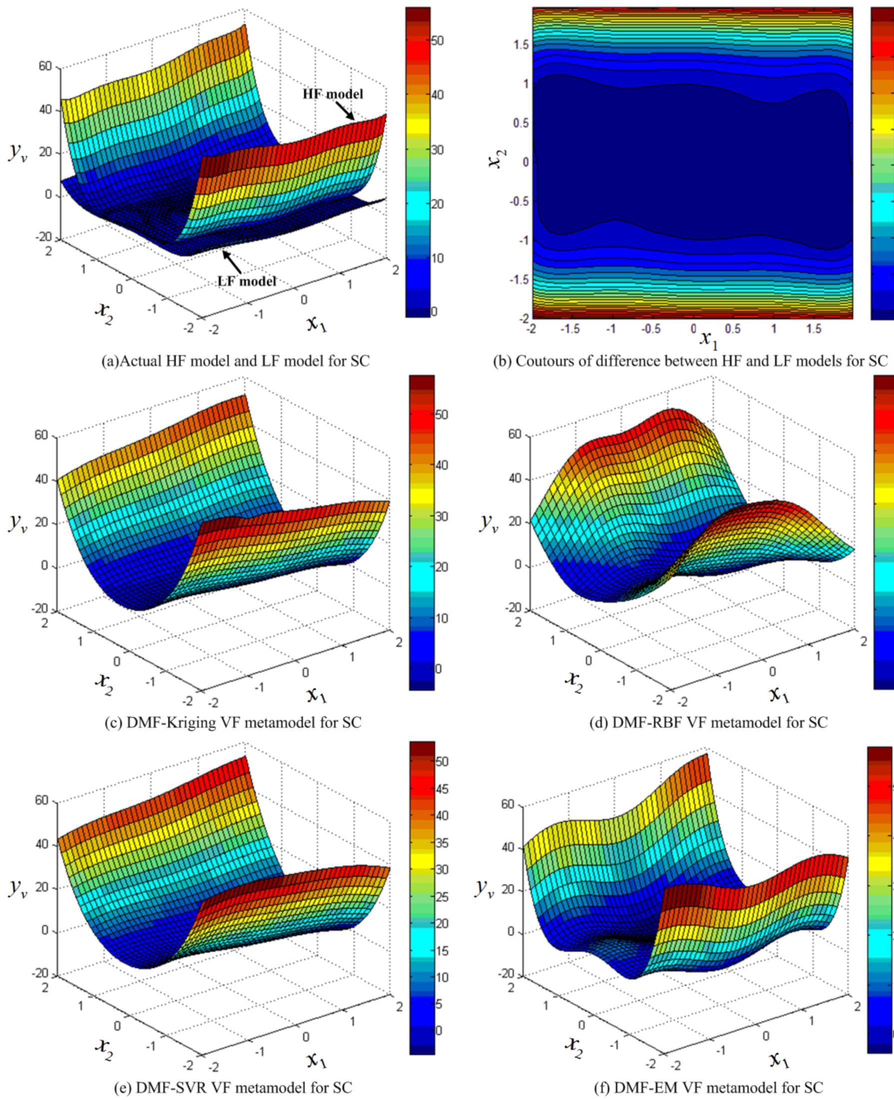
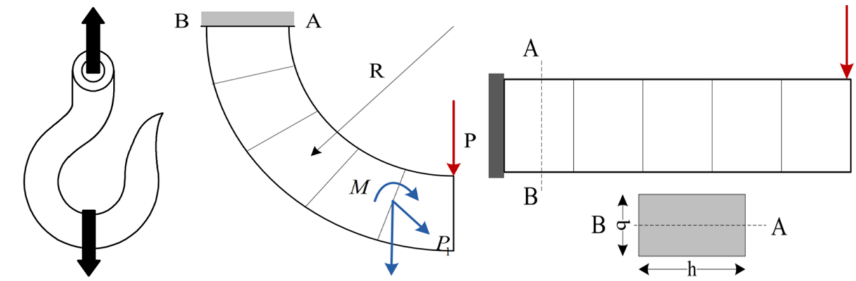


Figure 4: The actual model and difference metamodels for SC function under small sample set.



(a) The 10-ton crane (b) The curved beam divided into 5 sections taken as HF (c) The straight beam divided into 5 sections taken as LF

Figure 5: Schematic plot of the curved stepped beam.

the width of each segment ($b_i, i = 1, 2, \ldots, 5$). The ranges for the variables are $0.5 \leq h_i \leq 8; 0.5 \leq b_i \leq 2.25$, $i = 1, 2, \ldots, 5$. The radius of the middle layer of the curved beam, R, is $4.5in(11.43cm)$.

Two different fidelity models are used to calculate the maximum stress in each section. In HF model (shown in Fig.5 (b).), the maximum stress of each section is calculated using the formula for bending stress in curved beams, shown as follows:

$$\sigma_i = \frac{P_i}{b_i h_i} + \frac{M_i}{b_i h_i R}\left(1 + \frac{1}{Z}\frac{h_i/2}{R - h_i/2}\right) \quad , \text{ where } Z = -1 + \frac{R}{h_i}\ln\left(\frac{R + h_i/2}{R - h_i/2}\right) \tag{44}$$

where P_i and M_i are normal forces and bending couples for each section, which are formed by resolving the force P.

A straight beam with its length is equal to be R is adopted as the LF model, shown in Fig.5 (c). The straight beam is divided into the same number of sections as the HF model and the maximum stress of each section is calculated using the formula for bending stress in straight beams. It is expressed as:

$$\sigma_i = \frac{6M_i h_i}{b_i h_i^3} \tag{45}$$

Notice that, although the two fidelity models (Fig.5 (b) and Fig.5 (c)) are quite different, we felt that the LF model can correctly capture the general trends for changes in maximum stress of each beam section as variations of design variables. In this engineering case, 50 sample points are generated to obtain HF response values, while the LF model is directly used for VF modeling without fitting a metamodel to replace it. Additional 100 validation points are randomly selected to calculate the three accuracy metrics. The comparison results are summarized in Tab.3, where the best results for each model type are printed in bold, while the worst results for each model type are printed in italic. It is clear from Tab.3 that the proposed DMF-EM approach provides the most accurate VF metamodel compared to other three approaches. The accuracy of DMF-Kriging is somewhere in between DMF-RBF and DMF-SVR. Because the analogous types of maximum stress formulas in each beam section, the accuracy comparison results tend to be fairly uniform form σ_1 to σ_5.

Table 2: Accuracy comparison of the design of curved stepped beam.

Maximum Stresses	Accuracy criteria	DMF-Kriging	DMF-RBF	DMF-SVR	DMF-EM
σ_1	RMAE	1.1686	1.1751	***1.4497***	**1.102**
	RRMSE	0.2696	0.2736	***0.3335***	**0.2693**
	R^2	0.9388	***0.9386***	0.943	**0.9523**
σ_2	RMAE	1.8667	1.209	***2.538***	**1.0611**
	RRMSE	0.3173	0.3393	***0.4542***	**0.2553**
	R^2	0.9421	***0.9308***	0.9605	**0.9568**
σ_3	RMAE	2.1939	2.1858	***3.0598***	**1.1132**
	RRMSE	0.338	0.3439	***0.5089***	**0.2711**
	R^2	0.9354	***0.9316***	0.9488	**0.9496**
σ_4	RMAE	2.1557	2.1474	***3.0088***	**1.173**
	RRMSE	0.3354	0.3412	***0.5046***	**0.2692**
	R^2	0.9364	***0.9327***	0.9497	**0.9532**
σ_5	RMAE	1.5169	1.5059	***2.0858***	**1.0896**
	RRMSE	0.2984	0.3034	***0.4173***	**0.2622**
	R^2	0.9459	***0.9431***	0.9598	**0.9532**

4.3 Engineering case 2: design of a long cylinder pressure vessel for compressed natural gas

In this section, the implementation of the DMF-EM is demonstrated on a long cylinder pressure vessel design optimization problem. The geometry, model parameters and loading force of the long cylinder pressure vessel are illustrated in Fig. 6. The objective is to minimize the total consumption of the manufacturing material. Five continuous design variables are included: the height of the end part h_1, the inside diameter of the end part r_1, the thickness of the end part t_1, the inside diameter of the body part r_2 and the thickness of the body part t_2. The range of the design variables are listed in Tab. 3. Other geometric parameters are predefined and fixed during the optimization. The optimization is constrained by two design constraints, maximum allowable stress and minimum volume. The cylinder pressure vessel subjected a uniformly distributed load $P = 23MPa$.The Young's modulus and Poisson's ratio are $E = 207GPa$ and $u = 0.3$, respectively. The maximum all allowable stress and the minimum volume are $\sigma_{aw} = 250MPa$ and $V_{low} = 0.63m^2$, respectively.

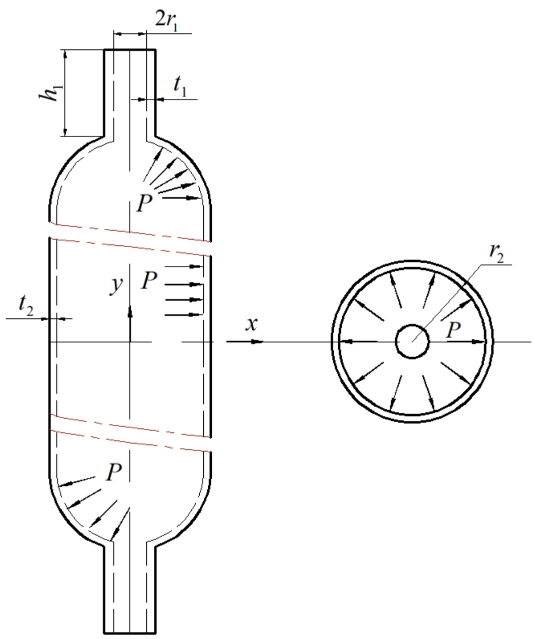


Figure 6: Schematic plot of the cylinder pressure vessel.

Table 3: Ranges of the design variables.

Design variables	Range (mm)
the height of the end part h_1	280-320
the inside diameter of the end part r_1	40-50
the thickness of the end part t_1	19-27
the inside diameter of the body part r_2	165-205
the thickness of the body part t_2	13-23

The mathematical formulation of the problem is as follows:

$$\begin{aligned} &\min : f \\ &s.t. : \sigma_s \leq \sigma_{al} \\ &\qquad V \geq V_{low} \end{aligned} \tag{46}$$

where V is the volume of the cylinder pressure vessel f is the total consumption of the manufacturing material. The quantities V and f are calculated using the following equations:

$$V = \pi r_2^2(6000 - h_1 - \sqrt{r_2^2 - t_1^2}) + 2\pi h_1 r_1^2 + \pi(r_1^2 + r_2^2)\sqrt{r_2^2 - r_1^2} + \frac{1}{3}\pi(\sqrt{r_2^2 - r_1^2})^3 \tag{47}$$

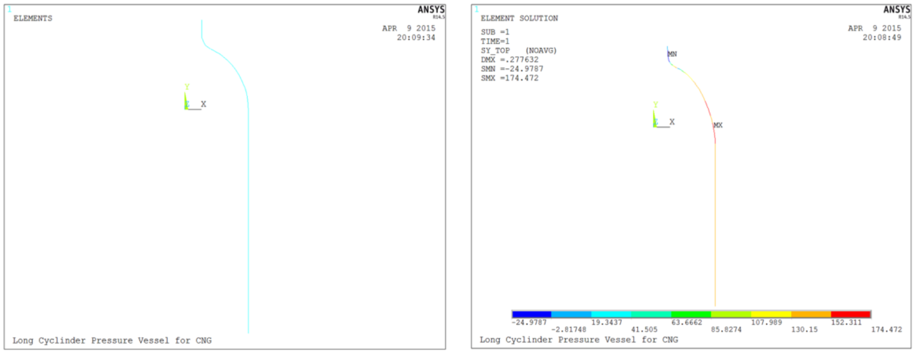
$$\begin{aligned} f =& \pi[(t_2 + r_2)^2 - r_2^2](6000 - h_1 - \sqrt{r_2^2 - r_1^2}) + 2\pi h_1[(t_1 + r_1)^2 - r_1^2] \\ &+ \frac{1}{3}\pi\{3[(t_1 + r_1)^2 + (t_2 + r_2)^2]\sqrt{(t_2 + r_2)^2 - (t_1 + r_1)^2} \\ &+ [\sqrt{(t_2 + r_2)^2 - (t_1 + r_1)^2}]^3 - 3(r_2^2 + r_1^2)\sqrt{r_2^2 - r_1^2} - (\sqrt{r_2^2 - r_1^2})^3\} \end{aligned} \tag{48}$$

From the above formula, it is found that the maximum von Mises stress of the pressure vessel cannot be obtained directly from an explicit function. Therefore, the proposed DMF-EM is adopted to fit the relationship between the stress response and the design variables. In this paper, ANSYS14.5 is used as a simulation tool for the stress response. Fig. 7 demonstrates the 1-D finite element model, which is used for LF model. Correspondingly, the axial symmetry 3-D finite element model with Hexahedral meshes is selected as the HF model, which is depicted in Fig. 8.

In this engineering example, 60 sample points are simulated for LF metamodel and the total number of sample points for developing the HF model is limited to 15. Additional 30 validation points are selected randomly to compare the accuracy performance of four different VF metamodeling methods. The accuracy comparison results are listed in Tab. 4 and the actual simulation values and the corresponding predicted values obtained using different VF metamodel are demonstrated in Fig. 9. It is observed that the proposed DMF-EM provides the most accurate metamodel. Based on the DFM-EM, the optimal design is as follows: the height of the end part $h_1 = 284.1mm$, the inside diameter of the end part $r_1 = 40.0mm$, the thickness of the end part $t_1 = 20.0mm$, the inside diameter of the body part $r_2 = 185.8mm$, the thickness of the body part $t_2 = 18.2mm$.

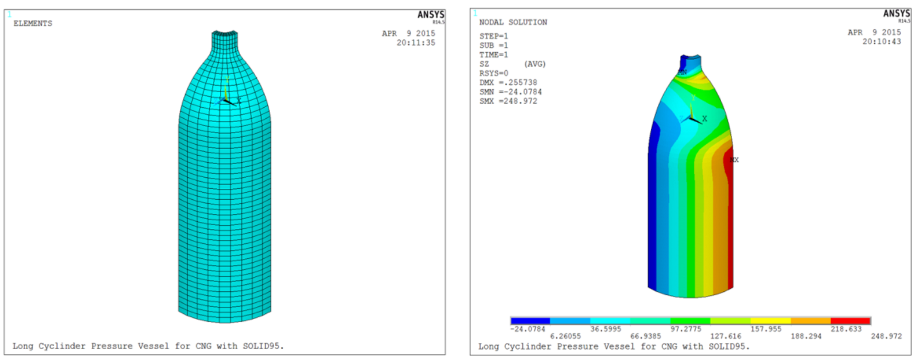
5 Conclusion

A DFM-EM variable fidelity modeling is proposed to approach the HF model when available HF sampling resource is limited. In DFM-EM, a linear tuning model is



(a) Grid model (b) Simulation analysis

Figure 7: 1-D LF model of the cylinder pressure vessel (a) grid model (b) simulation analysis.



(a) Grid model (b) Simulation analysis

Figure 8: 3-D HF model of the cylinder pressure vessel (a) grid model (b) simulation analysis.

Table 4: Accuracy comparison of the CNG pressure vessel.

	DMF-Kriging	DMF-RBF	DMF-SVR	DMF-EM
RMAE	***0.5486***	0.4001	0.3899	**0.3544**
RRMSE	0.2119	***0.2454***	0.1882	**0.1879**
R^2	0.9535	***0.9520***	0.9633	**0.9635**

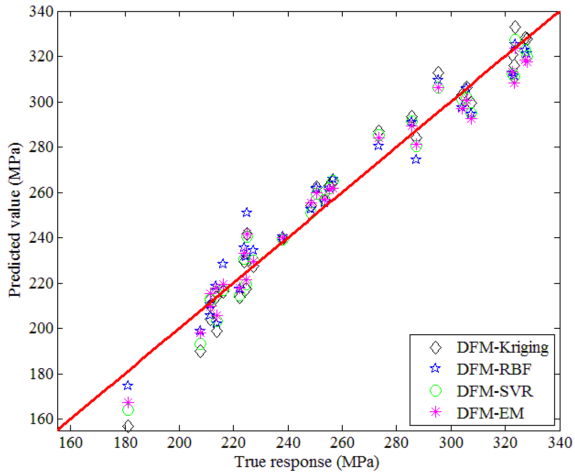


Figure 9: True response and predicted value of the validation points for different VF modeling methods, where the straight represents that the true response equal to predicted value.

created to pull the LF model to the true response of the HF model, then, the difference between the HF models and tuned model is fitted using the ensemble of metamodels, which is a hybrid of Kriging, RBF and LSSVR. Using the ensemble of metamodels in lieu of the scaling function can eliminate the need to determine a priori that which metamodel types should be used. Several numerical cases and two engineering design problems are employed to compare the accuracy and robustness performance of the proposed DFM-EM with those other three VF metamodeling methods. The comparison results illustrate that (1) for the same number of simulation evaluations and in terms of local/overall accuracy and robustness, the proposed VF metamodeling approach significantly outperforms the other three VF metamodeling methods, especially under the small sample size. (2) arbitrarily selecting metamodels to approximate scaling function increases the risk of adopting an unsatisfactory model (3) the proposed metamodeling approach accords with "insurance policy" mode that it will never be the worst, though it is not always be the best, which can decrease the risk of adopting an inappropriate modeling approach. It is also observed that the DFM-Kriging, DFM-RBF and DFM-SVR have comparable ability of applying problems of varying difficulty.

It should be pointed out that the proposed DFM-EM method may require additional computational cost to obtain the optimum tuning parameters and weight coefficients. However, this additional computational cost is more than likely offset by the savings in calls required for high fidelity models. Practically speaking, most engineering simulations usually yield multiple responses for each execution of the simulation model. Extending the DFM-EM to solve engineering design with multiple input and output parameters will be investigated in our future work. Overall, as a novel variable-fidelity modeling technique, DFM-EM exhibits great capability for metamodel-based engineering design and optimization problems.

Acknowledgement: This research has been supported by the National Natural Science Foundation of China (NSFC) under Grant No. 51505163, No. 51121002 and No. 51323009, National Basic Research Program (973 Program) of China under grant No. 2014CB046703, and the Fundamental Research Funds for the Central Universities, HUST: Grant No. 2014TS040. The authors also would like to thank the anonymous referees for their valuable comments.

References

Acar, E. (2010): Various approaches for constructing an ensemble of metamodels using local measures. *Structural and Multidisciplinary Optimization*, vol. 42, no. 6, pp. 879-896.

Alexandrov, N. M.; Lewis, R. M. (2001): An overview of first-order model management for engineering optimization. *Optimization and Engineering,* vol. 2, no. 4, pp. 413-430.

Chang, K. J.; Haftka, R. T.; Giles, G. L.; KAO, I. J. (1993): Sensitivity-based scaling for approximating structural response. *Journal of Aircraft*, vol. 30, no. 2, pp. 283-288.

Clarke, S. M.; Griebsch, J. H.; Simpson, T. W. (2005): Analysis of support vector regression for approximation of complex engineering analyses. *Journal of mechanical design*, vol. 127, no. 6, pp. 1077-1087.

Coello Coello, C. A. (2000): Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry,* vol. 41, no. 2, pp. 113-127.

Crombecq, K.; Laermans, E.; Dhaene, T. (2011): Efficient space-filling and noncollapsing sequential design strategies for simulation-based modeling. *European Journal of Operational Research,* vol. 214, no. 3, pp. 683-696.

De Munck, M.; Moens, D.; Desmet, W.; Vandepitte, D. (2009): An efficient response surface based optimisation method for non-deterministic harmonic and transient dynamic analysis. *CMES: Computer Modeling in Engineering & Sciences,* vol. 47, no. 2, pp. 119-166.

Gano, S. E., Renaud, J. E., & Sanders, B. (2004, August). Variable fidelity optimization using a kriging based scaling function. *In Proc. 10th AIAA/ISSMO*

Multidisciplinary Analysis and Optimization Conf., Albany, New York.

Gano, S. E.; Renaud, J. E.; Sanders, B. (2005): Hybrid variable fidelity optimization by using a kriging-based scaling function. *AIAA Journal,* vil. 43, no. 11, pp. 2422-2433.

Gao, L.; Xiao, M.; Shao, X.; Jiang, P.; Nie, L.; Qiu, H. (2012): Analysis of gene expression programming for approximation in engineering design. *Structural and Multidisciplinary Optimization,* vol. 46, no. 3, pp. 399-413.

Han, Z. H.; Görtz, S.; Zimmermann, R. (2013): Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function. *Aerospace Science and Technology,* vol. 25, no. 1, pp. 177-189.

Hastie, T.; Tibshirani, R.; Friedman, J. (2001): *The elements of statistical learning.* NY Springer.

Jiang, C.; Han, X. (2007): A new uncertain optimization method based on intervals and an approximation management model. *CMES: Computer Modeling in Engineering & Sciences,* vol. 22, no. 2, pp. 97-118.

Jiang, P.; Wang, J.; Zhou, Q.; Zhang, X. (2015): An Enhanced Analytical Target Cascading and Kriging Model Combined Approach for Multidisciplinary Design Optimization. *Mathematical Problems in Engineering,* 2015.

Jin, R.; Chen, W.; Simpson, T. W. (2001): Comparative studies of metamodelling techniques under multiple modelling criteria. *Structural and Multidisciplinary Optimization*, vol. 23, no. 1, pp. 1-13.

Kerh, T.; Lai, J. S.; Gunaratnam, D.; Saunders, R. (2008): Evaluation of seismic design values in the Taiwan building code by using artificial neural network. *CMES: Computer Modeling in Engineering & Sciences*, vol. 26, no. 1, pp. 1-12.

Li, Y.; Wu, Y.; Huang, Z. (2014): An Incremental Kriging Method for Sequential Optimal Experimental Design. *CMES: Computer Modeling in Engineering & Sciences,* vol. 97, no. 4, pp. 323-357.

Panda, S. S.; Manohar, C. S. (2008): Applications of meta-models in finite element based reliability analysis of engineering structures. *CMES: Computer Modeling in Engineering & Sciences,* vol. 28, no. 3, pp. 161-184.

Qian, Z.; Seepersad, C. C.; Joseph, V. R.; Allen, J. K.; Wu, C. J. (2006): Building surrogate models based on detailed and approximate simulations. *Journal of Mechanical Design*, vol. 128, no. 4, pp. 668-677.

Sacks, J.; Welch, W. J.; Mitchell, T. J.; Wynn, H. P. (1989): Design and analysis of computer experiments. *Statistical science*, pp. 409-423.

Shan, S.; Wang, G. G. (2010): Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. *Structural and Multidisciplinary Optimization*, vol. 41, no. 2, pp. 219-241.

Shokri, A.; Dehghan, M. (2012): A Meshless Method Using Radial Basis Functions for the Numerical Solution of Two—Dimensional Complex Ginzburg—Landau Equation. *CMES: Computer Modeling in Engineering & Sciences,* vol. 84, no. 4, pp. 333-358.

Simpson, T. W.; Toropov, V.; Balabanov, V.; Viana, F. A. (2008): Design and analysis of computer experiments in multidisciplinary design optimization: a review of how far we have come or not. *In 12th AIAA/ISSMO multidisciplinary analysis and optimization conference*, vol. 5, pp. 10-12.

Toropov, V. V.; Markine, V. L. (1996): The use of simplified numerical models as mid-range approximations. *In Proceedings of the 6-th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Part vol. 2, pp. 4-6.

Viana, F. A.; Simpson, T. W.; Balabanov, V.; Toropov, V. (2014): Metamodeling in Multidisciplinary Design Optimization: How Far Have We Really Come? *AIAA Journal,* vol. 52, no. 4, pp. 670-690.

Wang, G. G.; Shan, S. (2007): Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical Design,* vol. 129, no. 4, pp. 370-380.

Xiang, J.; Matsumoto, T.; Wang, Y.; Jiang, Z. (2011): A hybrid of interval wavelets and wavelet finite element model for damage detection in structures. *CMES: Computer Modeling in Engineering & Sciences,* vol. 81, no. 3, pp. 269-294.

Xiong, Y.; Chen, W.; Tsui, K. L. (2008): A new variable-fidelity optimization framework based on model fusion and objective-oriented sequential sampling. *Journal of Mechanical Design,* vol. 130, no. 11, 111401.

Zadeh, P. M.; Toropov, V. V.; Wood, A. S. (2009): Metamodel-based collaborative optimization framework. *Structural and Multidisciplinary Optimization,* vol. 38, no. 2, pp. 103-115.

Zheng, J.; Shao, X.; Gao, L.; Jiang, P.; Li, Z. (2013): A hybrid variable-fidelity global approximation modelling method combining tuned radial basis function base and kriging correction. *Journal of Engineering Design,* vol. 24, no. 8, pp. 604-622.

Zheng, J.; Shao, X.; Gao, L.; Jiang, P.; Qiu, H. (2014): A prior-knowledge input LSSVR metamodeling method with tuning based on cellular particle swarm optimization for engineering design. *Expert Systems with Applications,* vol. 41, no. 5, pp. 2111-2125.

Zhou, X. J.; Ma, Y. Z.; Li, X. F. (2011): Ensemble of surrogates with recursive arithmetic average. *Structural and Multidisciplinary Optimization,* vol. 44, no. 5, pp. 651-671.