High Dimensional Model Representation With Principal Component Analysis

Kambiz Haji Hajikolaei
e-mail: khajihaj@sfu.ca

G. Gary Wang
Professor
e-mail: gary_wang@sfu.ca

Product Design and Optimization
Laboratory (PDOL),
School of Mechatronic Systems Engineering,
Simon Fraser University,
250-13450 102 Avenue,
Surrey, BC V3T0A3, Canada

In engineering design, spending excessive amount of time on physical experiments or expensive simulations makes the design costly and lengthy. This issue exacerbates when the design problem has a large number of inputs, or of high dimension. High dimensional model representation (HDMR) is one powerful method in approximating high dimensional, expensive, black-box (HEB) problems. One existing HDMR implementation, random sampling HDMR (RS-HDMR), can build an HDMR model from random sample points with a linear combination of basis functions. The most critical issue in RS-HDMR is that calculating the coefficients for the basis functions includes integrals that are approximated by Monte Carlo summations, which are error prone with limited samples and especially with nonuniform sampling. In this paper, a new approach based on principal component analysis (PCA), called PCA-HDMR, is proposed for finding the coefficients that provide the best linear combination of the bases with minimum error and without using any integral. Several benchmark problems of different dimensionalities and one engineering problem are modeled using the method and the results are compared with RS-HDMR results. In all problems with both uniform and nonuniform sampling, PCA-HDMR built more accurate models than RS-HDMR for a given set of sample points.
[DOI: 10.1115/1.4025491]

Keywords: high dimension, large scale, metamodeling, HDMR, principal component analysis, sampling

1 Introduction

Metamodels are often built in engineering to simplify computationally intensive simulations of physical systems or phenomena. Clear successes have been made in the last two decades by modeling problems of low dimensionality ($d <= 10$, d is the problem dimensionality). Wang and Shan [1] listed roles of metamodeling in design optimization and reviewed the metamodeling techniques. Kriging [2], radial basis function [3], neural network [4], and multivariate adaptive regression splines [5] are common metamodeling techniques.

One major problem with metamodeling is when the dimension of the problem grows, the cost of sampling a sufficient number of points for metamodeling increases exponentially. This difficulty is known as "curse of dimensionality" [6]. Recently, the authors' team reviewed relevant techniques to solve problems with HEB functions [6]. Among these techniques, HDMR is identified as a promising metamodeling approach for HEB problems. HDMR is an approximation method, first introduced by Sobol [7], for representing high dimensional black-box functions. The general form of HDMR for a black-box function f with d input variables $(x_1, x_2, \ldots, x_d)$ is

$$f(x) = f_0 + \sum_{i=1}^{d} f_i(x_i) + \sum_{1 \le i < j \le d} f_{ij}(x_i, x_j) + \cdots + \sum_{1 \le i_1 < \cdots < i_l \le d} f_{i_1 i_2 \cdots i_l}(x_{i_1}, x_{i_2}, \ldots, x_{i_l}) + \cdots + f_{12\cdots d}(x_1, x_2, \ldots, x_d) \quad (1)$$

where f_0 is the zero-th order effect of $f(x)$, $f_i(x_i)$ is the first-order effect associated to variable x_i, independently; $f_{ij}(x_i, x_j)$ is the joint second-order effect associated to variables x_i and x_j; $f_{i_1 i_2 \cdots i_l}(x_{i_1}, x_{i_2}, \cdots x_{i_l})$ is the joint l-th order effect of variables $x_{i_1}, x_{i_2}, \cdots x_{i_l}$; and $f_{12\cdots d}(x_1, x_2, \cdots x_d)$ is the residual d-th order dependence of all variables on $f(x)$.

HDMR expression is a superposition of lower order functions. Usually in practice the first a few low order component functions are sufficient for approximation. Two main types of HDMR are analysis of variance (ANOVA)-HDMR and cut-HDMR [8,9]. Moving least square-HDMR [10], radial basis function-HDMR (RBF-HDMR) [11,12], multicut-HDMR [13], hybrid-HDMR [14], lumping-HDMR [15], indexing-HDMR [16], regularized RS-HDMR [17], and Chebyshev-HDMR [18] are other types of HDMR introduced by researchers.

ANOVA-HDMR is beneficial for statistical purposes but the main drawback in using ANOVA-HDMR is the need to compute lots of integrals. Usually Monte Carlo summations are used for computing the integrals that need numerous function evaluations. Different from ANOVA-HDMR, no integral is included in cut-HDMR component calculations. Cut-HDMR is a superposition of the function values on lines, planes, and hyperplanes. Generally, HDMR is used for two main purposes. The first is to generate an approximation for a black-box function. For this purpose, cut-HDMR is often used due to the simplicity of implementation. However, cut-HDMR requires well-structured sample points as dictated by the method, and is not accurate on hyperplanes that are not sampled in. The second purpose of HDMR is sensitivity analysis in order to identify important variables and correlations [19], for which ANOVA-HDMR is more suitable.

RS-HDMR is a modified version of ANOVA-HDMR [20] and uses orthonormal basis functions to build the approximation of black-box functions. The main advantage of RS-HDMR in comparison with ANOVA-HDMR is that all Monte Carlo summations can be performed using only one set of sample points in

Contributed by the Design Automation Committee of ASME for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received March 13, 2013; final manuscript received September 6, 2013; published online October 17, 2013. Assoc. Editor: Michael Kokkolaras.

RS-HDMR. However, in ANOVA-HDMR different sets of sample points are needed for different integrals of component functions. The other advantage of RS-HDMR is that randomly scattered data can be used for its component calculations but ANOVA-HDMR needs regularized inputs (controlled points) that may not be available for some problems. This advantage remains true when comparing RS-HDMR with cut-HDMR because the latter also requires regularized sampling. Another advantage is that RS-HDMR gives a mapping function between inputs and outputs but plain ANOVA-HDMR and cut-HDMR only give tables of points at the end. Finally, because of using basis functions in constructing RS-HDMR, by changing basis functions different approximation can be built using the same set of data.

Using RS-HDMR has two disadvantages that may affect the accuracy of the approximation. As it will be shown in the following sections, RS-HDMR is an integral-bases method and the coefficients for component functions are obtained using integrals computed from Monte Carlo summations. The Monte Carlo summations are accurate only if the number of sample points is sufficient and the points are distributed uniformly. However, in practice, the sample points may not be distributed uniformly and this may cause an inaccurate approximation model. Noticed in authors' previous work [21], if the density of sample points changes in different sub-regions of the design space, the RS-HDMR model becomes worse than otherwise. In specific, if a sub-region has denser sample points, the approximation model is poor for either the sub-region or outside of the sub-region. Again, this disadvantage comes from the integral-based nature of RS-HDMR. In this paper, the idea of using orthonormal basis functions for building metamodel is used but the coefficients are calculated without using any integrals. Principal component analysis [22] is used for this purpose and the approximation accuracy is compared with RS-HDMR. The proposed method is thus named PCA-HDMR.

The rest of the paper is organized as follows: first, the general form of HDMR is presented followed by RS-HDMR. Next, the proposed method, PCA-HDMR, is introduced and the pros and cons of the new method are discussed. Then, results of approximations for benchmark functions and the engineering problem are presented, and finally conclusions are drawn.

2 High Dimensional Model Representation

HDMR is a family of representations for capturing high dimensional input–output behavior of black-box systems. The general form of HDMR is shown in Eq. (1) that consists of terms for the individual and joint contribution of the input variables to the system output. For obtaining the HDMR terms, first assume that a real, scalar function $f(x)$ is defined on a unit hypercube

$$K^d = \{(x_1, x_2, \ldots, x_d): 0 \le x_i \le 1, \quad i = 1, 2, \ldots, d\} \tag{2}$$

The variables should be rescaled such that $0 \le x_i \le 1$, $i = 1, 2, \ldots, d$. For a general case, let's define μ as a product measure with unit mass and the following density:

$$\begin{aligned} d\mu(x) &= d\mu(x_1, \ldots, x_d) = \prod_{i=1}^{d} d\mu_i(x_i) \\ \int_{K^1} d\mu_i(x_i) &= 1 \\ d\mu(x) &= g(x)dx = \prod_{i=1}^{d} g_i(x_i)dx_i \end{aligned} \tag{3}$$

Different measures μ in Eq. (3) will make different types of HDMR. If the measure μ is the ordinary Lebesgue measure and if the orthogonality conditions are satisfied, the HDMR is called AVOVA-HDMR and the component functions in Eq. (1) can be obtained as below [8]

$$\begin{aligned} f_0 &= \int f(x)dx \\ f_i(x_i) &= \int f(x) \prod_{k \ne i} dx_k - f_o \\ f_{ij}(x_i, x_j) &= \int f(x) \prod_{k \ne i,j} dx_k - f_o - f_i(x_i) - f_j(x_j) \\ &\vdots \end{aligned} \tag{4}$$

The orthogonality condition is [9]

$$\int f_{i_1 \ldots i_s}(x_{i_1}, x_{i_2}, \ldots, x_{i_s}) \cdot f_{j_1 \ldots j_p}(x_{j_1}, x_{j_2}, \ldots, x_{j_p})dx = 0 \tag{5}$$

for cases that at least one index is different between $\{i_1 \ldots i_s\}$ and $\{j_1 \ldots j_p\}$, which can be derived from a requirement of HDMR component functions

$$\int_0^1 f_{i_1 \ldots i_l}(x_{i_1}, x_{i_2}, \ldots, x_{i_l})dx_k = 0, \quad k = i_1, \ldots, i_l \tag{6}$$

The integrals in Eq. (4) are obtained using Monte Carlo approximation. The number of function evaluations used for Lth order ANOVA-HDMR is [8]

$$N \times \left(\sum_{i=0}^{L} \frac{d!}{(d-i)!i!} \right) \tag{7}$$

where N is the number of sample points used in each Monte Carlo summation. The main drawback of ANOVA-HDMR is the computation of the integrals or equivalently the corresponding Monte Carlo summations. For each integration, a different set of sample points is needed in appropriate order. Therefore, numerous function evaluations are needed in a controlled manner.

Under this condition, RS-HDMR is proposed by Alis and Rabitz [20] as a modified version of HDMR that provides a mapping between input variables and system output using only one set of sample points (i.e., N function evaluations) randomly scattered in the space. The RS-HDMR is built as a linear combination of basis functions

$$f(x) = c_0 + \sum_{i=1}^{d} \sum_{k=1}^{s} c_i^k \phi_i^k(x_i) + \sum_{i<j}^{d} \sum_{k=1}^{s'} c_{ij}^k \phi_{ij}^k(x_i, x_j) + \ldots \tag{8}$$

where $\{\phi_i^k(x_i)\}_{k=1}^{s}$ is a family of linearly independent bases for univariate functions of x_i on [0 1] and $\{\phi_{ij}^k(x_i, x_j)\}_{k=1}^{s'}$ is similarly defined as a family of linearly independent bases for bivariate functions of x_i and x_j on $[0\ 1]^2$. Similarly, a set of basis functions can be defined for any higher orders of correlations with the following condition satisfied (similar to Eq. (6))

$$\int_{[0,1]} \phi_{i_1 i_2 \ldots i_l}^k(x_{i_1}, \ldots, x_{i_l})dx_m = 0, \quad k = 1, 2, \ldots, s'', \quad m = 1, 2, \ldots, l \tag{9}$$

Although the basis function can be defined in different ways, Alis and Rabitz [20] suggested the product of univariate basis functions for higher correlations

$$\phi^k_{i_1 i_2 \ldots i_l}(x_{i_1}, \ldots, x_{i_l}) = \phi^{k_1}_{i_1}(x_{i_1})\phi^{k_2}_{i_2}(x_{i_2})\ldots\phi^{k_l}_{i_l}(x_{i_l}) \qquad k \equiv (k_1, k_2, \ldots, k_l) \tag{10}$$

The coefficients related to RS-HDMR expansion can be calculated by

$$c_0 = \int_{[0,1]^d} f(x)dx \approx \frac{1}{N}\sum_{r=1}^{N} f\left(x_1^{(r)}, x_2^{(r)}, \ldots, x_d^{(r)}\right)$$

$$c_i^k = \int_{[0,1]^d} f(x)\phi_i^k(x_i)dx \approx \frac{1}{N}\sum_{r=1}^{N} f\left(x_1^{(r)}, x_2^{(r)}, \ldots, x_d^{(r)}\right)\phi_i^k\left(x_i^{(r)}\right)$$

$$c_{ij}^k = \int_{[0,1]^d} f(x)\phi_{ij}^k(x_i, x_j)dx \approx \frac{1}{N}\sum_{r=1}^{N} f\left(x_1^{(r)}, x_2^{(r)}, \ldots, x_d^{(r)}\right)\phi_{ij}^k\left(x_i^{(r)}, x_j^{(r)}\right) \tag{11}$$

The complete proof on how to obtain the coefficients can be found in Ref. [20]. The sensitivity analysis can be performed easily using the RS-HDMR coefficients [23]. Different sets of bases can be used for RS-HDMR approximation that may lead to different approximation. Li et al. [24] compared three types of basis functions (orthonormal polynomials, Cubic B spline, and polynomials) with direct Monte Carlo integrations and concluded that orthonormal polynomials provide the best accuracy. In another research, Li et al. [25] used product of lower order functions to build higher order component functions. Note that if the basis function shapes are similar to the black-box function, the model will be more accurate than otherwise.

Using Monte Carlo approximation instead of the corresponding integrals causes some errors that may lead to loss of accuracy in the RS-HDMR model [26]. Several attempts are made to improve the accuracy of Monte Carlo approximations. In Ref. [27] a correlation method is used for reducing the Monte Carlo summation error. The error can be decreased by either increasing the sample size or decreasing the variance of the function. The latter approach models the difference between the approximation model and the black-box function and adaptively modifies the model coefficients [27]. Li and Rabitz [28] used the ratio control variate method to reduce the Monte Carlo integration error, and therefore to improve the RS-HDMR accuracy.

Although using the mentioned methods, RS-HDMR accuracy can be improved but the fundamental error of using Monte Carlo summation to approximate integrals still exists. This error increases if the sampling is not performed uniformly in the space. In case of nonuniform sampling or having some subregions with different density of sample points, the RS-HDMR model will have a large error in both the dense and sparse subregions [21]. Therefore, using the existing RS-HDMR method for modeling a black-box function with existing nonuniform sample points may lead to poor approximation. In this paper, the RS-HDMR coefficients are calculated using another method without using any integration. Thus, the errors from Monte Carlo summation disappear and models are more accurate, especially in nonuniform sampling cases. The proposed method is explained in Sec. 3.

3 PCA-HDMR

PCA is a technique to analyze data by transforming multivariable data to a set of new orthogonal variables so that the importance of the variables is revealed. PCA was originated in 1901 [29], but the term principal component was formally used in 1933 [30]. PCA has four main goals [22]: (1) extracting the most important part of data, (2) decreasing the size of data, (3) obtaining the structure of data, and (4) simplifying data description. These goals are obtained by introducing new variables called *principal components*, which are linear combinations of existing variables. The first principal component accounts for the largest variation in the data. The second principal component is computed in a direction that has the second largest variation and is also orthogonal to the first component. The other components are specified similarly. Higher order components must be orthogonal to all the lower order components. New values of the data in new coordinates are called *factor scores*. PCA can be performed using singular value decomposition (SVD). Suppose that a data matrix X is of $N \times d$; then the SVD of the matrix can be shown as

$$X = P\Delta Q^{\mathrm{T}} \tag{12}$$

In which P is an $N \times L$ matrix called left singular vectors; Q is a $d \times L$ matrix called right singular vectors. L is the number of eigenvalues and Δ is the diagonal matrix of singular values. Factor scores (F) can be obtained using the multiplication of matrices P and Δ

$$F = P\Delta \tag{13}$$

Q gives the coefficients that can be used for linear transformation between the previous and new variables, and can be interpreted as the projection matrix between the raw data and factor scores [22]. In this paper, PCA properties are used as an application along with HDMR to identify the best linear combination of basis functions to build the approximation with minimum variation from the black-box function.

One salient feature of PCA is that the new components are ordered by the amount of variations in the observations. In other words, the component with the maximum possible variation is the first component with the largest singular value and the one with minimum possible variation is the last component with the smallest singular value. Also, all the components are orthogonal to each other. In fact, SVD provides a rotation of the coordinates in a way that the variation of data is maximum along the first and minimum along the last coordinates. Geometrically, for an example, Fig. 1 shows a 2D data set including 20 sample points, shown as black dots. The original coordinates are shown by solid lines. After performing the SVD, the coordinates are rotated using the Q matrix. The dashed axis shows the first new coordinate (with maximum

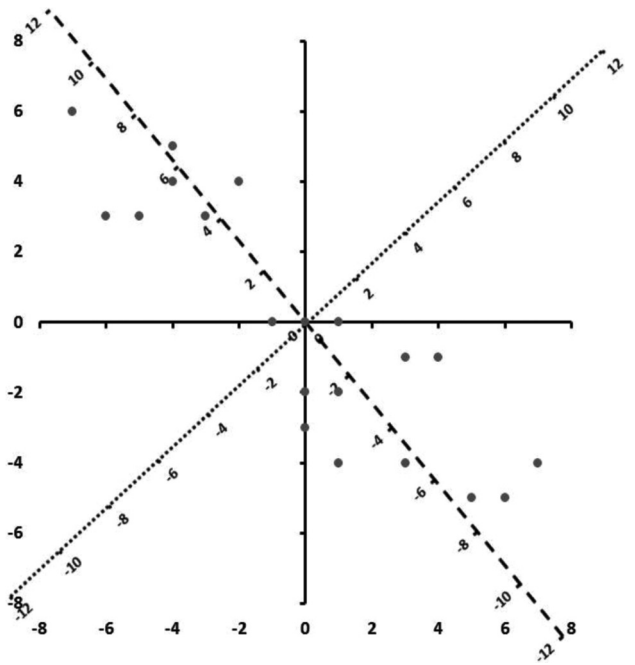


Fig. 1 The geometric representation of PCA

variation) and the dotted axis shows the last one (with minimum variation). It can be seen in the figure that data are located along the dashed axis. Thus, if someone has to model the system with only one coordinate, the dashed coordinate has the most influence on the variation and the dotted one can be removed with the least possible error. In the special case that all the sample points are located on the dashed axis, the system can be modeled using only one independent variable without any error.

In addition, it is noted that the transformation in PCA is performed with linear combinations of existing variables. Recall that the RS-HDMR approximation is a *linear combination* of the orthogonal basis functions. The proposed approach, called PCA-HDMR, then tries to build such linear combinations of the basis function so that components of HDMR are re-ordered according to their importance. Then, the last component, which should have the least variation, is set to be zero to find the most efficient set of HDMR model coefficients. Also, other components include information of the black-box function and can be used in metamodeling. This will be further explained in this section 3.

For a RS-HDMR in Eq. (8), suppose that the approximation is truncated at the L-th order. Therefore, the HDMR structure is

$$f(x) = c_0 + \sum_{i=1}^{d}\sum_{k=1}^{s} c_i^k \phi_i^k(x_i) + \sum_{i<j}^{d}\sum_{k=1}^{s'} c_{ij}^k \phi_{ij}^k(x_i,x_j) + \cdots + \sum_{i_1<i_2\ldots<i_L}^{d}\sum_{k=1}^{s''} c_{i_1 i_2 \ldots i_L}^k \phi_{i_1 i_2 \ldots i_L}^k(x_{i_1},\ldots,x_{i_L}) \tag{14}$$

The original coordinates are the input variables $[x_1,\ldots,x_d]$. A new set of coordinates are defined as a gathering of all the existing basis functions in the approximation with different input variables. The black-box function f is added to the new coordinate set as well. The new coordinate set Φ can be represented as

$$\begin{aligned}
\Phi &= [\Phi_1, \Phi_2, \ldots, \Phi_L, f] \\
\Phi_1 &= \left[\phi_1^1(x_1), \ldots, \phi_d^s(x_d)\right] \\
\Phi_2 &= \left[\phi_{12}^1(x_1,x_2), \ldots, \phi_{(d-1)d}^{s'}(x_{d-1},x_d)\right] \\
&\vdots \\
\Phi_L &= \left[\phi_{1\ldots L}^1(x_1,\ldots,x_L), \ldots, \phi_{(d-L+1)\ldots d}^{s''}(x_{d-L+1},\ldots,x_d)\right]
\end{aligned} \tag{15}$$

Φ_L shows all combinations of the L-th order bases and input variables. However, if there is *a priori* knowledge that some of the combinations of variables do not exist, then they can be removed from both the HDMR structure and the new coordinate set Φ. In this paper, the function is considered to be completely black-box without any a priori knowledge. Therefore, the approximations are performed using all the possible combinations of the bases and input variables until the specified truncating order.

For simplicity of representation, the bases are numbered from the first to the last one as shown in Eq. (16). For example, the first $\psi_{i,i=1..s\times d}$ are from Φ_1, followed by the terms in Φ_2, and so on.

$$\hat{\Psi} = [\psi_1, \psi_2, \ldots, \psi_m, \tilde{f}] \tag{16}$$

where $\psi_1, \psi_2, \ldots, \psi_m$ are all the combinations of basis functions used in the approximation and $\tilde{f}$ is rescaled value of f over the existing sample points that can be calculated as

$$\tilde{f} = 2 \times \frac{[f - \min(f)]}{[\max(f) - \min(f)]} - 1 \tag{17}$$

The scaling is to bring the f values to be within the interval of $[-1\ 1]$, so that the range of function values is comparable with those of the basis functions. If all the possible combinations are considered, the number of bases will be

$$m = \sum_{i=0}^{L} \frac{d!}{(d-i)!i!} \tag{18}$$

Also all of the components are subtracted by their average value to place the origin of the new coordinate system at the center

$$\Psi = \left[\psi_1 - \bar{\psi}_1, \psi_2 - \bar{\psi}_2, \ldots, \psi_m - \bar{\psi}_m, \tilde{f} - \bar{\tilde{f}}\right] \tag{19}$$

where $\bar{\psi}_1, \bar{\psi}_2, \ldots, \bar{\psi}_m$ are the average values of the bases $\psi_1, \psi_2, \ldots, \psi_m$ over the existing data and similarly $\bar{\tilde{f}}$ is the average value of $\tilde{f}$. It is clear that the new set of coordinates has $(m+1)$ members. Assume that N sample data exist for building the approximation. The original data can be shown as an N×d matrix. The data are transformed to the new set by computing all bases and f values, and putting them in the matrix Ψ. After transformation, the new dataset is an $N \times (m+1)$ matrix. If the SVD procedure is performed on the new data matrix, the corresponding right singular vectors matrix Q will be a set of linear transformations between the coordinates (basis functions) with the property that the first one accounts for the maximum variation and the last one accounts for the minimum variation. In other words, the last column will be the *linear combination* that gives the minimum possible amount of variation. Set the linear combination in the last column to be zero, one can have

$$\alpha_1(\psi_1 - \bar{\psi}_1) + \alpha_2(\psi_2 - \bar{\psi}_2) + \cdots + \alpha_m(\psi_m - \bar{\psi}_m) + \alpha_{m+1}\left(\tilde{f} - \bar{\tilde{f}}\right) \approx 0 \tag{20}$$

where $[\alpha_1, \alpha_2, \ldots, \alpha_{m+1}]^T$ is the last column of Q. Therefore the approximation model $\tilde{f}$ can be found by the following:

$$\tilde{f} = \bar{\tilde{f}} + \frac{1}{\alpha_{m+1}} \times \left[-\alpha_1(\psi_1 - \bar{\psi}_1) - \alpha_2(\psi_2 - \bar{\psi}_2) - \cdots - \alpha_m(\psi_1 - \bar{\psi}_m)\right] \tag{21}$$

The PCA-HDMR approximation coefficients are calculated using the procedure presented above and it offers a number of advantages as compared with RS-HDMR. The most important advantage is that the new method does not use any integral approximation. Therefore, the errors coming from the integral approximations are eliminated. Second, uniform sampling is no longer needed because Monte Carlo approximation is not used in the new method. If the density of the sampling is changed in some sub-regions, it will affect the approximation accuracy much less in comparison with RS-HDMR and again it comes from not using any integrals. Moreover, the ratio of the minimum singular value to other singular values can determine the accuracy of the approximation before building the model. If the last column of the matrix Q that is used as coefficients of Eq. (20) corresponds to a very small singular value, it means that the black-box function f can be well built using linear combination of bases for the given points. The third advantage of PCA-HDMR is that being accurate with nonuniform sampling and having no singularity issue make the PCA-HDMR a method that can accommodate samples of different weights. This means that a user can emphasize a region by repeating sample points falling into the region without incurring new sample points. In other words, PCA-HDMR can not only function

as a global model but also a local metamodel in a concentrated region. We call this the ability as "zoom in, zoom out," which shows great promises for supporting optimization of HEB problems.

On the other hand, the proposed method may seemingly have some disadvantages than RS-HDMR. First, the SVD procedure may be slow as it may involve large size matrices with very large amount of data. Second, if some sample points are added to the data, PCA should be performed again for updating the model while in RS-HDMR a simple summation could adaptively update the model. Both disadvantages can be easily eliminated. For the first one, in PCA-HDMR procedure, only the projection matrix Q and singular value matrix Δ are used, not the left singular vector matrix P. Therefore, if matrixes Q and Δ can be obtained using other calculations, SVD can be removed from the procedure. If X in Eq. (12) is an $N \times (m+1)$ matrix including m bases and the black-box function values in N sample points, then Q and Δ^2 are eigenvectors matrix and eigenvalues matrix of X^TX, respectively. Thus, instead of using SVD, it is sufficient to calculate the eigenvalues and eigenvectors of X^TX, which is always an $(m+1) \times (m+1)$ matrix. The second disadvantage is also avoided by only calculating eigenvalues and eigenvectors of X^TX instead of SVD matrices. New points can be added adaptively to update X, and the remaining work is to calculate the eigenvalues and eigenvectors of X^TX, not to perform the more costly SVD. Therefore, using SVD can be replaced by the mentioned algebraic calculations. In the testing section, the above-mentioned procedure is used instead of SVD.

Until now, just the last column of eigenvector matrix, corresponding to the smallest eigenvalue, is used for building PCA-HDMR model. If the coefficients of the basis functions are fixed in the whole approximation region, the last column guarantees the best combination of the basis functions for approximating the black-box function. If the basis functions are chosen as polynomials up to the second order, PCA-HDMR using only the last column gives the least square coefficients, which is mathematically equivalent to the response surface method (RSM) with second order polynomials. PCA-HDMR, however is not limited to polynomial basis functions or second order. Moreover, in smaller sub-regions of the space, with PCA-HDMR one can use not only the last column of the eigenmatrix to obtain other combinations of the basis functions, which may give better approximations in sub-regions. In fact, if the last eigenvalue is not zero, it can be concluded that the approximation using just the last column is not necessarily the best for all sub-regions. In other words, other columns of eigenvector matrix have useful information of the black-box problem that can be used for metamodeling. If the difference between the smallest and the second-smallest eigenvalues is large, it indicates that the second last column (assuming columns are sorted in descending order according to their eigenvalues) has small effect on the approximation but if the two eigenvalues are close to each other, both of the corresponding eigenvectors are important. The same observation can be made for other eigenvalues and eigenvectors. The effect can be shown visually in Fig. 1. The dotted line corresponds to the eigenvector with the smallest eigenvalue and shows the minimum variation from the original function in the whole approximation space. If all the sample points were located on the dashed line, it could be concluded that using the dotted direction in PCA-HDMR guarantees the best approximation in all sub-regions for the existing sample points. However, now that the sample points are not located exactly on the dashed line, it indicates that the dashed direction should be used for modeling in some sub-regions. The elegance of PCA-HDMR is that it reveals other directions' information that can be used for approximation.

The model is built with combinations of different component PCA-HDMR models with weights to be found in every sub-region. Equation (22) shows the PCA-HDMR metamodel using more than one eigenvector columns

$$f_{\text{PCA-HDMR}} = \frac{c_1 f_{\text{PCA-HDMR}^1} + c_2 f_{\text{PCA-HDMR}^2} + c_3 f_{\text{PCA-HDMR}^3} + \cdots}{\sum c_i} \tag{22}$$

where $f_{\text{PCA-HDMR}^i}$ shows the PCA-HDMR model built using the $(m+2-i)$-th column of the eigenvector matrix (i-th component PCA-HDMR) and c_i is the corresponding weight of the column approximation in the sub-region. $f_{\text{PCA-HDMR}^1}$ is the PCA-HDMR component that uses only the last eigenvector of the PCA-HDMR matrix, as shown previously. The c_i values show the importance of the component PCA-HDMR metamodels in the sub-regions. In this paper, c_1 is set to be equal to one and other c_i values are changing between zero and one. For every single test point, a specified number of closest sample points around the point (N_{close}) are considered and the c_i values are chosen in a way that the error between model values and actual values become minimum. Changing N_{close} affects the accuracy of the model. Choosing the best N_{close} value depends on the density of the sample points in the space. In this paper we use $2 \times d$ in which d is the number of variables. However, more intelligent ways can be used for selecting N_{close} which is left as future work. Different methods can be used for minimizing the error. In this paper, the values are simply changed between zero and one and the best values are selected. c_i values are chosen one-by-one from c_2 to c_L in which L is the number of terms (component PCA-HDMR) used for metamodeling. Therefore, c_i values are specified separately by order of their importance. Other methods such as optimizing c_i values together can be used as well.

If just the last eigenvector is used for metamodeling, and if only up to second order polynomial basis functions are used, the final PCA-HDMR metamodel will be the same as RSM metamodel. RSM can be therefore considered as a special case of PCA-HDMR. However, PCA-HDMR has more advantages that cause its superiority as compared with RSM:

(1) The major difference between PCA-HDMR and RSM is that PCA-HDMR reveals other possible combinations of basis functions along different principle component directions, which makes PCA-HDMR working in both global and local regions. The results related to use of other eigenvectors in metamodeling will be shown in Sec. 4.
(2) Because of using HDMR structure and the orthogonality of the HDMR components, all the HDMR properties are inherited in PCA-HDMR model. One of them is decomposing the effect of different variables. The effect of independent variables and the joint effect of the variables are obtained separately, similar to Eq. (1). The next one is the efficient and easy computation of sensitivity indices, as mentioned in Ref. [20].
(3) Any orthonormal basis functions can be used in PCA-HDMR. Using the basis functions similar to the shape of the black-box function improves the accuracy of the metamodel.

In conclusion, the advantages of RS-HDMR in approximating black-box functions are retained in PCA-HDMR and new advantages are added to more accurately calculate the coefficients.

4 Method Testing

In this section, PCA-HDMR is tested with a number of benchmark functions of different dimensionalities and different variable correlations. Then, the proposed method is applied to an engineering problem. The model accuracy is evaluated using three error metrics and the results are compared with the RS-HDMR approximation with the same basis functions, sample points, and test points. The benchmark functions are treated like black-box functions. The input sample data are generated using pseudorandom

Table 1 Comparison of RS-HDMR and PCA-HDMR accuracies (benchmark function 10, sampling types 1 and 2)

		R-square				RAAE				RAME			
		RS-HDMR		PCA-HDMR		RS-HDMR		PCA-HDMR		RS-HDMR		PCA-HDMR	
Sampling type	NSP	Ave.	STD	Ave.	STD	Ave.	STD	Ave.	STD	Ave.	STD	Ave.	STD
1	1000	−8.4807	1.9115	0.9901	0.0012	1.7573	0.2082	0.5065	0.0932	7.5537	1.3463	1.9794	0.2916
	2000	−2.3200	0.3897	0.9696	0.0014	1.2103	0.1228	0.3048	0.0666	5.2414	0.9629	1.1867	0.2374
2	1000	−6.4277	1.5783	0.9903	0.0009	1.6147	0.1744	0.4734	0.0732	6.8093	1.1928	1.9394	0.3420
	2000	−1.9220	0.3941	0.9688	0.0021	1.1982	0.0950	0.3298	0.0876	4.9481	0.7138	1.2518	0.3261

values drawn from the standard uniform distribution on the interval $[0, 1]$ scaled to the interval between lower bounds and upper bounds. Note that each variable has its own lower bound and upper bound that should be used in scaling the variable. For investigating the effect of nonuniformity in the PCA-HDMR modeling, two different types of sample data are generated:

(1) Random uniform samples: in this case, the sample points are generated completely randomly without any preference in the entire feasible space. MATLAB command rand(.) is used for generating the random points separately for every input variable and they are scaled separately bases on the variable bounds.
(2) Random nonuniform samples: in this case, again MATLAB command rand(.) is used for generating the sample points in [0, 1] and then the values are scaled. However, 80% of the points are sampled in the entire feasible region and 20% are sampled in the region with $\hat{x}_i < r$, where $\hat{x}_i$ is the sample random value before scaling. The value r ($r = 0.1^{\frac{1}{d}}$) is calculated for each problem so that the region with $\hat{x}_i \leq r$ entails 10% of the entire region volume. Thus, 10% of the entire space has a higher density in comparison with the remaining 90%.

For testing the method accuracy, three different metrics (R-square, relative average absolute error (RAAE), and relative maximum absolute error (RMAE), see Appendix B) are calculated for each modeling using the same sample and testing points for both PCA-HDMR and RS-HDMR. For comparing the accuracy of PCA-HDMR with RS-HDMR, different benchmark functions with different number of input variables are selected [31,32] and the model accuracies are compared. The benchmark functions as well as the variable ranges are shown in the Appendix. The Appendix A table contains two other parameters, NC and Space. The first one, NC, is the number of coefficients that should be calculated for building PCA-HDMR and RS-HDMR models, which can be obtained as

$$\mathrm{NC} = \sum_{i=1}^{L} \binom{d}{i} s^i \tag{23}$$

where L and s are the maximum order of correlation between the variables and the number of bases in the model, respectively. Note that the number of data points should be equal or more than the number of coefficients (unknowns). Therefore, for building PCA-HDMR model, at least NC sample points are needed. The variable Space shows the existing space that the model is being built in and is obtained using multiplication of the variable ranges in each function. This parameter is one of the criteria showing the difficulty of the problem.

Table 1 shows the R-square, RAAE, and RMAE values related to the approximation of the benchmark function #10 mentioned in Appendix A, using both PCA-HDMR and RS-HDMR for completely random data (sampling type 1) and random nonuniform data (sampling type 2). Two different numbers of sample data are selected and the approximations are built 20 times. First and second order basis functions are used ($s = 2$) and the maximum number of correlations is two ($L = 2$). The number of sample points used for the approximations are shown by NSP in the table. The values reported in Table 1 include the average and standard deviation of the 20 runs. As can be seen from the table, for sampling type 1, R-square values of PCA-HDMR are much closer to one than the values of RS-HDMR. This is because PCA-HDMR finds the coefficients that minimize the variation of the model from the black-box function. For calculating RAAE and RMAE values, 200 random test points are used. RAAE and RMAE values of PCA-HDMR are closer to zero than the values of RS-HDMR. The standard deviation values of PCA-HDMR are also less than the values of RS-HDMR. All the benchmark functions are tested in the same manner as for function #10 with similar results; detailed results are omitted for brevity. The PCA-HDMR models are more accurate than RS-HDMR for all the cases for a given set of sampling points (R-square) and for most of the cases with new test points (RAAE and RAME). For ease of comparing the results and brevity, just the average values related to the first ten benchmark functions (using one NSP value) are shown by plots in Figs. 2–4. Note that the chosen numbers of sample points are more than the NC values for all the functions.

Figure 2 shows the R-square values without negative values. By looking at R-square definition, the value is always less than one and the distance from one shows the accuracy of the model. Therefore, the negative values of the results show poor models built by RS-HDMR and are not shown in the figure. As anticipated using the properties of PCA, all R-square values are closer to one in PCA-HDMR than RS-HDMR. R-square values show that function #4 is a very difficult one for modeling. Figures 3 and 4, respectively, show RAAE and RMAE values related to the first ten benchmark functions for both RS-HDMR and PCA-HDMR approaches. It can be seen that RAAE and RMAE values are less in PCA-HDMR than RS-HDMR in all the cases except for the function #4. As stated before, PCA-HDMR gives the best RS-HDMR model with respect to sample points, not necessarily the model with the best extrapolation capability when tested with new test points. Function #4 is an example of this statement. However, RAAE and RMAE values are very close in this case.

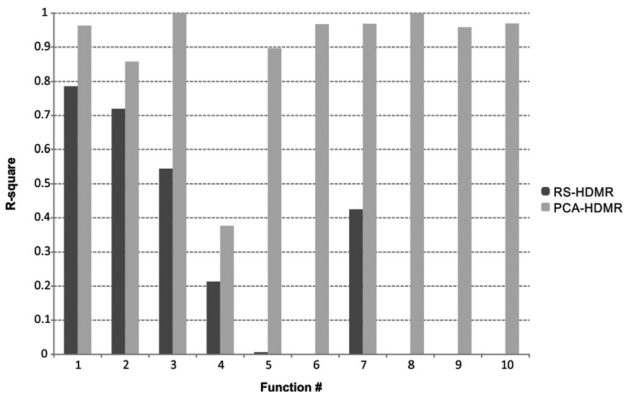


Fig. 2 R-square values of the first ten benchmark functions (average of 20 runs, sampling type 1)

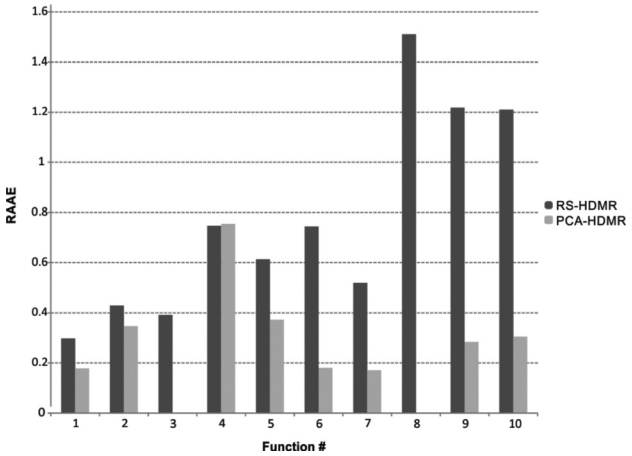


Fig. 3 RAAE values of the first ten benchmark functions (average of 20 runs, sampling type 1)

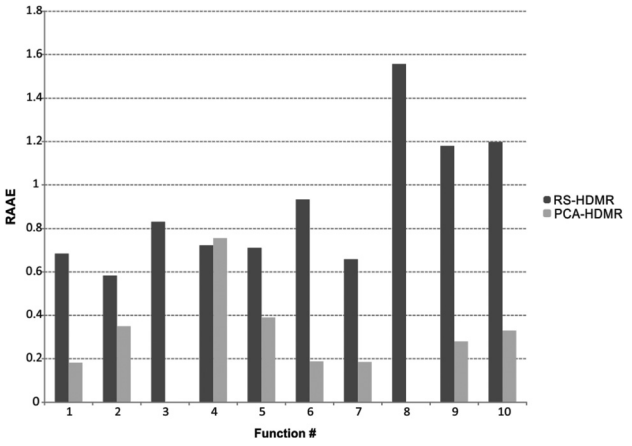


Fig. 5 RAAE values of the first ten benchmark functions (average of 20 runs, sampling type 2)

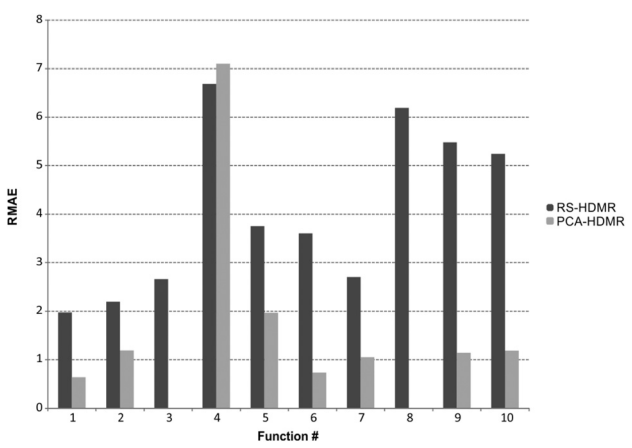


Fig. 4 RMAE values of the first ten benchmark functions (average of 20 runs, sampling type 1)

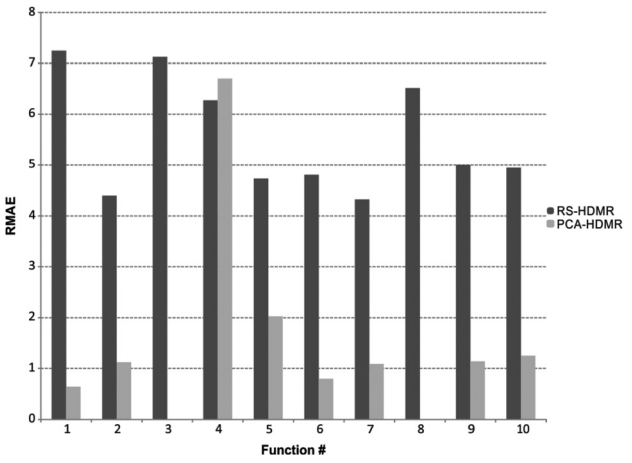


Fig. 6 RMAE values of the first ten benchmark functions (average of 20 runs, sampling type 2)

For comparing the effect of sampling, the second type of sampling is performed as mentioned before (nonuniform) for all the cases with the same number of sample points. The results related to benchmark function #10 are shown in Table 1. The only difference between the tests comparing to previous cases is the type of sampling. Comparing the R-square, RAAE, and RMAE values of sampling types 1 and 2, the RS-HDMR approximations with sampling type 2 have very poor R-square values and large errors, but the PCA-HDMR approximations are still good. The observation can be explained by the use of Monte Carlo summations in RS-HDMR that are accurate only with uniform sampling. Therefore, PCA-HDMR makes better approximations than RS-HDMR with nonuniform sample points for a given set of sample points for all the cases, and better model accuracy at new test points for most of the cases.

Again for the ease of comparing the results, RAAE and RMAE values related to the first ten benchmark function are plotted in Figs. 5 and 6, respectively. Similar to the uniform sampling case, the values are better in PCA-HDMR than RS-HDMR for all the cases except for function #4. In almost all the cases, the values are worse than the uniform sampling case (sampling type 1) in both RS_HDMR and PCA-HDMR but the differences are small for PCA-HDMR. Almost all the R-square values became negative in RS-HDMR using sampling type 2 and it means that nonuniformity of the sampling has huge effect on RS-HDMR. Because almost all the R-square values of RS-HMDR are negative, R-square values are thus not shown graphically.

As mentioned in Sec. 3, with PCA-HDMR, the user can put weights on certain regions and make the approximation more accurate in these regions by repeating sample points. Assume that a multimodel black-box function is needed to be modeled only with the first and second order basis functions. It is clear that RS-HDMR cannot well model the function due to the multimodal shape of the function. But PCA-HDMR can zoom in small regions and model it accurately only with the first and second order basis functions. For having a weight of w on a region, one can simply repeat w times the samples falling into the region.

To demonstrate this concept, a sinusoidal function ($f(x) = \sin(x)$) is selected in a specific range ($-\pi \leq x \leq \pi$). Figure 7 shows the original function as well as different PCA-HDMR models for the function. A set of $N = 11$ uniformly distributed points between the lower and upper bounds of the problem are used as the modeling points. Due to the multimodality of the sine function, the normal PCA-HDMR cannot model it well globally and due to its origin symmetry just a line is used for modeling (dash-dotted). A weight of 10 is put on the last 6 points (second half including the mean) and the result is shown by the dashed curve. It can be seen that the curve is more similar to the second half of the original function. The dotted line shows the PCA-HDMR model with a weight of 1000 on the last 6 points. The metamodel curve becomes very close to the original function, but just in the region $x \geq 0$. In other words, the metamodel "zoomed in" to the region $0 \leq x \leq \pi$. Note that when the weight is put on a specific region, the model becomes worse in other regions. For example in the sine example, when the weights are put on the region $0 \leq x \leq \pi$, then the other half ($-\pi \leq x < 0$) became far from the original function. It is not shown in the figure due to its scale.

Until now, all the PCA-HDMR models were built using only the last eigenvector. However, as mentioned in Sec. 3,

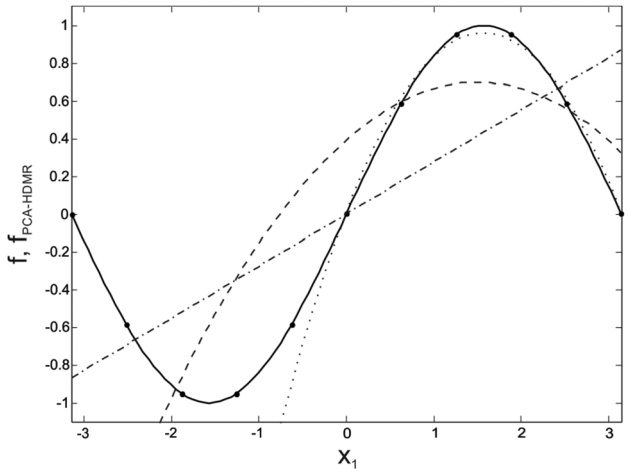
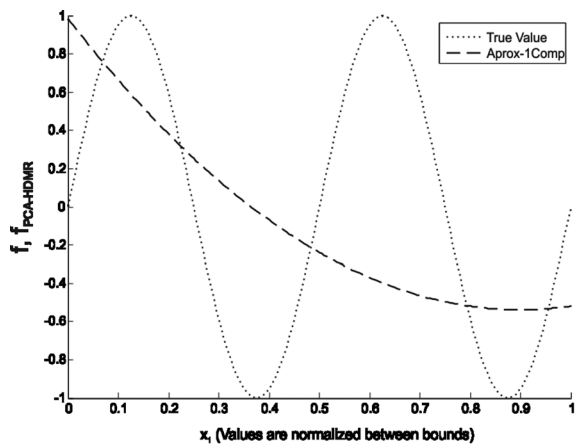


Fig. 7 Sine function (solid), normal PCA-HDMR approximation (dash-dotted), PCA-HDMR with weight 10 after $x \geq 0$ (dashed), and PCA-HDMR with weight 1000 after $x \geq 0$ (dotted)

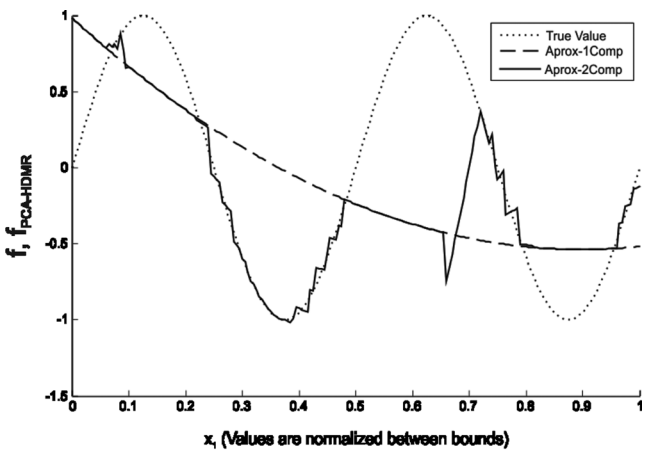
PCA-HDMR gives out more information about the shape of the function that can be used in metamodeling. For showing the effect of other eigenvectors on metamodeling, a one dimensional sinusoidal function ($f(x_1) = \sin(x_1), -2\pi \leq x_1 \leq 2\pi$) is considered as the black-box function and PCA-HDMR models with a combination of different numbers of eigenvectors were built.

The original function and the PCA-HDMR model using just the last column are shown in Fig. 8(*a*). The x values are normalized to be between zero and one. First order and second order orthonormal polynomial basis functions are used for metamodeling using 100 random sample points. It can be seen that polynomial basis functions are not good choices for this function and the approximation is poor. The next PCA-HDMR with the second last eigenvector is added to the metamodel using Eq. (22). For every single point, the closest two sample points are chosen ($N_{\text{close}} = 2$) and the corresponding c_2 value is changed between zero and one to find the minimum error in the chosen points. Figure 8(*b*) shows the original function, and PCA-HDMR approximation with both $f_{\text{PCA}-\text{HDMR}^1}$ and $f_{\text{PCA}-\text{HDMR}^2}$. It can be seen that the shape of the function is well predicted by the approximation in some subregions. It can be seen that the second last eigenvector is important for this approximation and improves the model. Because two basis functions are used for building the PCA-HDMR matrix, it has three eigenvectors that can be used in the approximation. The next term $f_{\text{PCA}-\text{HDMR}^3}$ is added to the model. c_2 is fixed to the best value obtained before and c_3 is changed between zero and one. Figure 8(*c*) shows the results including the approximation using three component PCA-HDMR models. Comparing with Fig. 8(*b*), the metamodel is improved when the third component PCA-HDMR, $f_{\text{PCA}-\text{HDMR}^3}$, is added to the approximation.

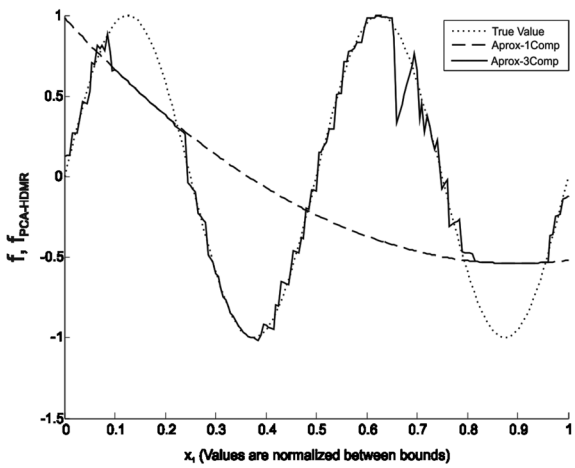
Because the sample points are randomly chosen, the same metamodeling process is repeated ten times and the results are shown in Table 2. RAAE and RMAE values are obtained using 200 test points similar to the previous examples. All three measurement metrics are improved when new terms are added to the



(a)



(b)



(c)

Fig. 8 PCA-HDMR results using different number of components

Table 2 Comparison of PCA-HDMR results using different number of components (1D sinusoidal function)

	R-square		RAAE		RMAE	
	Ave	STD	Ave	STD	Ave	STD
PCA-HDMR, 1 Component	0.07581	0.0810	0.8010	0.0310	1.8514	0.1180
PCA-HDMR, 2 Components	0.56329	0.0360	0.4290	0.0566	1.7058	0.1760
PCA-HDMR, 3 Components	0.94494	0.0187	0.1769	0.0319	0.7436	0.1336

Table 3 Comparison of PCA-HDMR results using different number of components (2D sinusoidal function)

	R-square		RAAE		RMAE	
	Ave.	STD	Ave.	STD	Ave.	STD
PCA-HDMR, 1 Component	0.3859	0.0424	0.6816	0.0584	3.5567	1.5725
PCA-HDMR, 2 Components	0.5792	0.0502	0.5213	0.0414	1.8606	0.3684
PCA-HDMR, 3 Components	0.7415	0.0481	0.4041	0.0499	1.6257	0.2610
PCA-HDMR, 4 Components	0.8163	0.0418	0.3449	0.0458	1.5944	0.3100

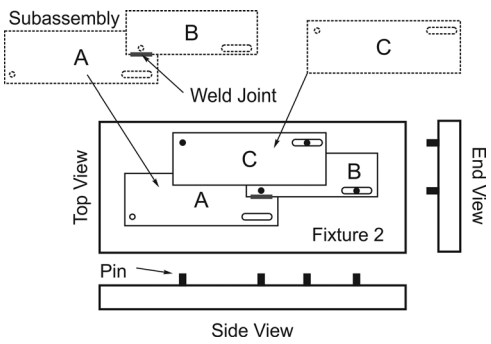


Fig. 9 Three-part assembly problem and the related fixtures [33]

PCA-HDMR model and the small standard deviation shows that the metamodel is robust as well.

The next example is a two dimensional sinusoidal function ($f(x_1, x_2) = \sin(x_1)\sin(x_2), -\pi \le x_1, x_2 \le \pi$). Similar to previous examples, first and second order basis functions are used for the metamodeling. 200 randomly scattered sample points are chosen for building PCA-HDMR matrix. For adding the new component PCA-HDMR terms, the closest four sample points around the test points are chosen and the error is minimized. Table 3 shows the average and standard deviation of the results of 10 independent runs. R-square, RAAE, and RMAE values are all improved when the new terms are added to PCA-HDMR model.

After testing the method with benchmark functions, an engineering problem is selected to study the effectiveness of the method in practice. A three-part assembly variation problem, shown in Fig. 9 is chosen from Ref. [33] and both PCA-HDMR and RS-HDMR models are built for the variation of its specific key characteristic (KC) and the results are compared. The parts can be assembled together in different ways. In this example, at the first step, part A and part B are assembled and then, part C is joined to the subassembly of part A and part B. The fixture locations are input variables of the problem. The distance between the lower left corner of part A and the upper right corner of part C defines the KC and the six-sigma variation of the KC is the objective function to be approximated.

First, the model is created in 3 DCS software [34] with defined dimensions, 400 mm length and 200 mm width. Holes, slots, and pins are defined with diameter equal to 10 mm for holes and 9 mm for pins. Tolerances are defined for hole, slot, and pin sizes with a range of ± 0.5 mm with normal distribution. Three holes and three slots exist in the model and for defining each of them X and Y coordinate values are needed. Therefore, the problem has 12 input variables in total. The six-sigma value of the specified KC is obtained from Monte Carlo simulation in 3DCS, which is considered a black-box function that should be modeled.

The potential locations for holes and slots are defined by a grid of points with increments of 10 mm and at least 10 mm away from the edges. Again, different numbers of random points are selected and RS-HDMR and PCA-HDMR models are built. The results are presented in Table 4.

By increasing the number of sample points, R-square value of RS-HDMR is increased and the RAAE value is decreased. RMAE value varies by increasing the NSP. It means that the model is getting more accurate overall with more points. The R-square value is decreasing in PCA-HDMR with the increasing number of sample points. It is expected because PCA-HDMR tries to find the coefficients in a way that the model becomes close to all of the sample points. By increasing the number of sample points it becomes harder to do that. The same phenomenon can be observed in most of the benchmark function tests. However, the R-Square values of PCA-HDMR are clearly better than that of RS-HDMR. By comparing the RAAE values of Table 4, it can be concluded the

Table 4 Comparison of RS-HDMR and PCA-HDMR accuracies for the three-part assembly problem

	R-square		RAAE		RMAE	
NSP	RS-HDMR	PCA-HDMR	RS-HDMR	PCA-HDMR	RS-HDMR	PCA-HDMR
500	−2.5751	0.6878	1.6012	1.4363	7.5942	7.5084
1000	−1.2096	0.5313	1.3747	0.7996	7.6629	4.0883
2000	−0.6484	0.4360	0.9387	0.5061	6.7493	6.6233
5000	−0.2883	0.3638	0.6462	0.4229	4.5317	4.1593
10,000	−0.0458	0.3666	0.5430	0.3995	6.5530	6.3246
20,000	0.0007	0.3557	0.5161	0.3740	5.6276	5.8543
40,000	0.0441	0.3484	0.4644	0.3339	10.3197	10.1934

PCA-HDMR model is getting more accurate by increasing the number of sample points. Again, RMAE value of PCA-HDMR, similar to RS-HDMR, varies with the increasing sample points and it means that the model compromises between the global and local accuracy. By increasing the number of sample points, the global accuracy is increased but the model becomes less accurate in one or some local regions. In general, comparing the R-square, RAAE, and RMAE values of two methods in all the *NSP* values, it can be seen that PCA-HDMR is doing a better job than RS-HDMR.

5 Conclusions

In this paper, a new approach is proposed to efficiently model black-box functions. Rooted in the random sampling high dimensional model representation (RS-HDMR) structure with orthogonal basis functions, the proposed principal component analysis (PCA) based HDMR, PCA-HDMR, finds the best basis function coefficients. In this approach, the sample data are first transferred to another space containing all bases and the black-box function. PCA is performed on the new data and therefore the linear combination of bases with minimum variation is identified and coefficients of the bases are calculated. Theoretically, based on SVD calculations PCA-HDMR yields the most accurate model among all possible RS-HDMR models with the same sample points.

Three different performance metrics are defined to test the accuracy with both the modeling points and test points, and also the accuracy in global and local regions. The method is compared with RS-HDMR with 15 benchmark functions of different numbers of input variables (2–50) and a high dimensional engineering problem (12 variables). It is concluded from the results that PCA-HDMR generates more accurate model than RS-HDMR for all the cases for a given set of sampling points, as well as for most of the cases at new test points. Moreover, two types of sampling are tested and results are compared. Changing from completely random uniform to random nonuniform sampling, RS-HDMR model falls apart but such a change has a small effect on the PCA-HDMR model. This finding can be explained by the nature of RS-HDMR that is integral-based and uses Monte Carlo summation for approximation. If the sampling differs from being uniform, the integrals and the corresponding Monte Carlo summations become inaccurate and therefore the model becomes inaccurate as well. In addition, PCA-HDMR allows a user exerting more weights in a local region by simply repeating the sample points falling into the region. Lastly, the usage of more than one component PCA-HDMR terms leads to more accurate models, which makes PCA-HDMR a metamodel working in both global and local regions. Higher accuracy, flexibility of taking nonuniform sampling, and the ability of local intensification, make PCA-HDMR promising to support optimization for HEB problems.

Acknowledgment

Funding from Natural Science and Engineering Research Council of Canada (CRDPJ421433-11) and General Motors Company is gratefully acknowledged.

Appendix A: Benchmark Functions

No.	d	Function	Variable ranges	NC	Space
1	2	$f = 100\left(x_2 - x_1^2\right)^2 + (1 - x_1)^2$	$-2 \le x_i \le 2,$	8	16
2	2	$f = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	$-2 \le x_i \le 2,$	8	16
3	3	$f = (x_1 + x_2)^2 + (x_2 + x_3)^2$	$-2 \le x_i \le 2,$	18	64
4	4	$f = -x_1x_2x_3x_4$	$-2 \le x_i \le 2,$	32	256
5	5	$f = (x_1 - x_2)^2 + (x_3 - 1)^2 + (x_4 - 1)^4 + (x_5 - 1)^6$	$-2 \le x_i \le 2,$	50	1024
6	10	$f(\boldsymbol{x}) = (x_1 - 1)^2 + (x_{10} - 1)^2 + 10\sum_{i=1}^{9}(10 - i)\left(x_i^2 - x_{i+1}\right)^2$	$-3 \le x_i \le 2,$	200	9765625
7	10	$f(\boldsymbol{x}) = \left[\sum_{i=1}^{10} i^3(x_i - 1)^2\right]^3$	$-3 \le x_i \le 3,$	200	60466176
8	20	$f(\boldsymbol{x}) = \sum_{i=1}^{10}\left[100(x_i - x_{i+10})^2 + (x_i - 1)^2\right]$	$-3 \le x_i \le 5,$	800	$1.1529 \times 10^{+018}$
9	20	$f(\boldsymbol{x}) = \sum_{1}^{5}\Big[100\left(x_i^2 + x_{i+5}\right)^2 + (x_i - 1)^2 + 90\left(x_{i+10}^2 + x_{i+15}\right)^2 + (x_{i+10} - 1)^2 + 10.1\left[(x_{i+5} - 1)^2 + (x_{i+15} - 1)^2\right] + 19.8(x_{i+5} - 1) \times (x_{i+15} - 1)\Big]$	$-3 \le x_i \le 5,$	800	$1.1529 \times 10^{+018}$
10	20	$f(\boldsymbol{x}) = \sum_{1}^{5}\left[(x_i + 10x_{i+5})^2 + 5(x_{i+10} - x_{i+15})^2 + (x_{i+5} - 2x_{i+10})^4 + 10(x_i - x_{i+15})^4\right]$	$-2 \le x_i \le 5,$	800	$7.9792 \times 10^{+016}$
11	30	$f(\boldsymbol{x}) = 1 - \exp\left[-\frac{1}{60}\sum_{1}^{30}x_i^2\right]$	$0 \le x_i \le 3.5,$	1800	$2.0991 \times 10^{+016}$
12	30	$f(\boldsymbol{x}) = (\boldsymbol{x}^T A\boldsymbol{x})^2, A = \mathrm{diag}(1, 2, 3, \ldots, 30)$	$-2 \le x_i \le 3,$	1800	$9.3132 \times 10^{+020}$
13	30	$f(\boldsymbol{x}) = \sum_{i=1}^{29}\left[100\left(x_{i+1} - x_i^2\right)^2 + (1 - x_i)^2\right]$	$-2 \le x_i \le 2,$	1800	$1.1529 \times 10^{+018}$
14	50	$f(\boldsymbol{x}) = \boldsymbol{x}^T A\boldsymbol{x} - 2x_1, A = \begin{bmatrix} 1 & -1 & & & 0 \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & \cdots & \cdots & \cdots & 0 \\ & & \cdots & \cdots & \cdots \\ & & -1 & 2 & -1 \\ 0 & & & -1 & 2 \end{bmatrix}$	$0 \le x_i \le 25,$	5000	$7.8886 \times 10^{+069}$
15	20	$f(\boldsymbol{x}) = \sum_{1}^{20}x_i^2 + \left[\sum_{1}^{20}\frac{1}{2}ix_i\right]^2 + \left[\sum_{1}^{20}\frac{1}{2}ix_i\right]^4$	$0 \le x_i \le 5,$	800	$9.5367 \times 10^{+013}$

Appendix B: Modeling Methods Accuracy Metrics Definitions

(1) R-square

$$R^2 = 1 - \frac{\sum_{s=1}^{N}\left[f\left(x^{(s)}\right) - \hat{f}\left(x^{(s)}\right)\right]^2}{\sum_{s=1}^{N}\left[f\left(x^{(s)}\right) - \bar{f}\left(x^{(s)}\right)\right]^2} \tag{B1}$$

where $f\left(x^{(s)}\right)$ and $\hat{f}\left(x^{(s)}\right)$ are the black-box function and the approximated model function at the sample point $x^{(s)}$, respectively. $\bar{f}\left(x^{(s)}\right)$ denotes the mean of the black-box function over the N sample points in approximating the black-box function. The R-square indicates the overall accuracy of the approximated model with given sample points. The closer the value of R-square approaches one, the more accurate is the approximation model. However, R-square is not sufficient for comparing accuracies of two models because it only shows the accuracy with respect to modeling points, not the model's extrapolation ability.

(2) RAAE

$$\text{RAAE} = \frac{\sum_{t=1}^{n}\left|f\left(x^{(t)}\right) - \hat{f}\left(x^{(t)}\right)\right|}{n * \text{STD}} \tag{B2}$$

where $f\left(x^{(t)}\right)$ and $\hat{f}\left(x^{(t)}\right)$ are the black-box function and the approximated model function at n test points $x^{(t)}$, respectively. STD denotes the standard deviation of the black-box function prediction at the test points. RAAE indicates the overall accuracy of the approximated model in the entire design range, computed at random test points. The smaller is RAAE value, the more accurate is the approximation. Both R-square and RAAE show the average accuracy in the entire space but in some cases the maximum amount of error in local regions is worthy of study.

(3) RMAE

$$\text{RMAE} = \frac{\max\left(\left|f(\boldsymbol{x}_1) - \hat{f}(\boldsymbol{x}_1)\right|, \left|f(\boldsymbol{x}_2) - \hat{f}(\boldsymbol{x}_2)\right|, \ldots, \left|f(\boldsymbol{x}_n) - \hat{f}(\boldsymbol{x}_n)\right|\right)}{\text{STD}} \tag{B3}$$

In this metric, the maximum amount of error at random test points is divided by the standard deviation. Note that this error is a local metric and again the smaller RMAE value, the better is the approximation.

References

[1] Wang, G. G., and Shan, S., 2007, "Review of Metamodeling Techniques in Support of Engineering Design Optimization," ASME J. Mech. Des., **129**(4), pp. 370–380.

[2] Cresssie, N., 1988, "Spatial Prediction and Ordinary Kriging," Math. Geol., **20**(4), pp. 405–421.

[3] Fang, H., and Horstemeyer, M. F., 2006, "Global Response Approximation With Radial Basis Functions," J. Eng. Optim., **38**(4), pp. 407–424.

[4] Papadrakakis, M., Lagaros, M., and Tsompanakis, Y., 1998, "Structural Optimization Using Evolution Strategies and Neural Networks," Comput. Methods Appl. Mech. Eng., **156**(1–4), pp. 309–333.

[5] Friedman, J. H., 1991, "Multivariate Adaptive Regressive Splines," Ann. Stat., **19**(1), pp. 1–67.

[6] Shan, S., and Wang, G. G., 2010, "Survey of Modeling and Optimization Strategies to Solve High Dimensional Design Problems With Computationally Expensive Black-Box Functions," Struct. Multidiscip. Optim., **41**(2), pp. 219–241.

[7] Sobol, I. M., 1993, "Sensitivity Estimates for Nonlinear Mathematical Models," Math. Modell. Comput. Exp., **1**(4), pp. 407–414.

[8] Rabitz, H., and Alis, O. F., 1999, "General Foundation of High Dimensional Model Representation," J. Math. Chem., **25**, pp. 197–233.

[9] Li, G., Rosenthal, C., and Rabitz, H., 2001, "High Dimensional Model Representations," J. Phys. Chem., **105**(33), pp. 7765–7777.

[10] Wang, H., Tang, L., and Li, G. Y., 2011, "Adaptive MLSHDMR Metamodeling Techniques for High Dimensional Problems," Exp. Syst. Appl., **38**, pp. 14117–14126.

[11] Shan, S., and Wang, G. G., 2010, "Metamodeling for High Dimensional Simulation-Based Design Problems," ASME J. Mech. Des., **132**(5), pp. 1–11.

[12] Shan, S., and Wang, G. G., 2011, "Turning Black Box Into White Function," ASME J. Mech. Des., **133**(3), p. 031003.

[13] Li, G., Schoendorf, J., Ho, T., and Rabitz, H., 2004, "Multicut-HDMR With an Application to an Ionospheric Model," J. Comput. Chem., **25**, pp. 1149–1156.

[14] Tunga, M. A., and Demiralp, M., 2006, "Hybrid High Dimensional Model Representation (HHDMR) on the Partitioned Data," J. Comput. Appl. Math., **185**, pp. 107–132.

[15] Tunga, M. A., and Demiralp, M., 2008, "Introductory Steps for an Indexing Based HDMR Algorithm: Lumping HDMR," 1st WSEAS International Conference on Multivariate Analysis and its Application in Science and Engineering, MAASE'08, May 27–30, Istanbul, Turkey, pp. 129–135.

[16] Tunga, M. A., 2011, "An Approximation Method to Model Multivariate Interpolation Problems: Indexing HDMR," Math. Comput. Model., **53**, pp. 1970–1982.

[17] Li, G., and Rabitz, H., 2007, "Regularized Random-Sampling High Dimensional Model Representation (RS-HDMR)," J. Math. Chem., **43**(3), pp. 1207–1232.

[18] Thomas, P. S., Somers, M. F., Hoekstra, A. W., and Kroes, G. J., 2012, "Chebyshev High-Dimensional Model Representation (Chebyshev-HDMR) Potentials: Application to Reactive Scattering of H2 from Pt(111) and Cu(111) Surfaces," Phys. Chem. Chem. Phys., **14**, pp. 8628–8643.

[19] Kaya, H., Kaplan, M., and Saygin, H., 2004, "A Recursive Algorithm for Finding HDMR Terms for Sensitivity Analysis," Comput. Phys. Commun., **158**, pp. 106–112.

[20] Alis, O. F., and Rabitz, H., 2001, "Efficient Implementation of High Dimensional Model Representations," J. Math. Chem., **29**(2), pp. 127–142.

[21] Hajikolaei, K. H., and Wang, G. G., 2012, "Adaptive Orthonormal Basis Functions for High Dimensional Metamodeling With Existing Sample Points," Proceedings of the ASME 2012 International Design Engineering Technical Conference and Computers and Information in Engineering Conference, DETC2012-70480, Aug. 12–15, Chicago, IL.

[22] Abdi, H., and Williams, L. J., 2010, "Principal Component Analysis," Wiley Interdiscip. Rev. Comput. Stat., **2**(4), pp. 433–459.

[23] Li, G., Wang, S. W., Rabitz, H., Wang, S., and Jaffé, P., 2002, "Global Uncertainty Assessments by High Dimensional Model Representations (HDMR)," Chem. Eng. Sci., **57**(21), pp. 4445–4460.

[24] Li, G., Wang, S. W., and Rabitz, H., 2002, "Practical Approaches to Construct RS-HDMR Component Functions," J. Phys. Chem., **106**, pp. 8721–8733.

[25] Li, G., Artamonov, M., Rabitz, H., Wang, S., Georgopoulos, P. G., and Demirlap, M., 2002, "High-Dimensional Model Representations Generated from Low Order Terms—lp-RS-HDMR," J. Comput. Chem., **24**(5), pp. 647–656.

[26] Li, G., Hu, J., Wang, Sh., Georgopoulos, P. G., Schoendorf, J., and Rabitz, H., 2006, "Random Sampling-High Dimensional Model Representation (RS-HDMR) and Orthogonality of its Different Order Component Functions," J. Phys. Chem., **110**, pp. 2474–2485.

[27] Li, G., Rabitz, H., Wang, S., and Georgopoulos, P. G., 2002, "Correlation Method for Variance Reduction of Monte Carlo Integration in RS-HDMR," J. Comput. Chem., **24**(3), pp. 277–283.

[28] Li, G., and Rabitz, H., 2006, "Ratio Control Variate Method for Efficiently Determining High-Dimensional Model Representations," J. Comput. Chem., **27**, pp. 1112–1118.

[29] Pearson, K., 1901, "On Lines and Planes of Closest Fit to Systems of Points in Space," Philos. Mag., **6**, pp. 559–572.

[30] Hotelling, H., 1933, "Analysis of a Complex of Statistical Variables Into Principal Components," J. Educ. Psychol., **25**, pp. 417–441.

[31] Hock, W., and Schittkowski, K., 1980, "Test Examples for Nonlinear Programming Codes," J. Optim. Theory Appl., **30**(1), pp. 127–129.

[32] Schittkowski, K., 1987, *More Test Examples for Nonlinear Programming Codes*, Springer-Verlag, New York.

[33] Whitney, D. E., 2004, *Mechanical Assemblies*, Oxford University Press, New York.

[34] 3DCS Variation Analyst, 2013, Available at: www.3DCS.com