

Radial basis functions as surrogate models with a priori bias in comparison with a posteriori bias

Kaveh Amouzgar^{1,3}  · Niclas Strömberg²

Received: 22 February 2016 / Revised: 6 July 2016 / Accepted: 15 August 2016 / Published online: 29 September 2016
© The Author(s) 2016. This article is published with open access at Springerlink.com

Abstract In order to obtain a robust performance, the established approach when using radial basis function networks (RBF) as metamodels is to add a posteriori bias which is defined by extra orthogonality constraints. We mean that this is not needed, instead the bias can simply be set a priori by using the normal equation, i.e. the bias becomes the corresponding regression model. In this paper we demonstrate that the performance of our suggested approach with a priori bias is in general as good as, or even for many test examples better than, the performance of RBF with a posteriori bias. Using our approach, it is clear that the global response is modelled with the bias and that the details are captured with radial basis functions. The accuracy of the two approaches are investigated by using multiple test functions with different degrees of dimensionality. Furthermore, several modeling criteria, such as the type of radial basis functions used in the RBFs, dimension of the test functions, sampling techniques and size of samples, are considered to study their affect on the performance of the approaches. The power of RBF with a priori bias for

surrogate based design optimization is also demonstrated by solving an established engineering benchmark of a welded beam and another benchmark for different sampling sets generated by successive screening, random, Latin hypercube and Hammersley sampling, respectively. The results obtained by evaluation of the performance metrics, the modeling criteria and the presented optimal solutions, demonstrate promising potentials of our RBF with a priori bias, in addition to the simplicity and straight-forward use of the approach.

Keywords Metamodeling · Radial basis function · Design optimization · Design of experiment

1 Introduction

With exponentially increasing computing power, designers have today the possibility by simulation driven product development to create new innovative complex products in a short time. In addition, simulation based design also reduces the cost of product development by eliminating the need of creating several physical prototypes. Furthermore, a designer can create an optimized design with respect to multiple objectives with several constraints and design variables. However, the models and simulations, particularly those pertained in multidisciplinary design optimization (MDO), can be very complex and computationally expensive, see e.g. the multi-objective optimization of a disc brake in Amouzgar et al. (2013). Surrogate or metamodels have been accepted widely in the MDO community to deal with this issue. A metamodel is an explicit approximation function that predicts the response of a computational expensive simulation based model such as a non-linear

✉ Kaveh Amouzgar
kaveh.amouzgar@his.se

Niclas Strömberg
niclas.stromberg@oru.se

¹ Product Development Department, School of Engineering, Jönköping University, Jönköping, 55111 Sweden

² Department of Mechanical Engineering, School of Science and Technology, University of Örebro, Örebro, 70182 Sweden

³ School of Engineering Science, University of Skövde, P.O. Box 408, 541 28 Skövde, Sweden

finite element model. It also develops a relation between the input variables and their corresponding responses. In general, the aim of a metamodel is to create an approximation function of the original function over a given design domain. Many metamodeling methods have been developed for metamodel based design optimization problems. Some of the most recognized and studied metamodels are response surface methodology (RSM) or polynomial regression (Box and Wilson 1951), Kriging (Sacks et al. 1989), radial basis functions (Hardy 1971), support vector regression (SVR) (Vapnik et al. 1996) and artificial neural networks (Haykin 1998). Extensive surveys and reviews of different metamodeling methods and their applications are e.g. given by Simpson et al. (2001a, b, 2008), Wang and Shan (2007) and Forrester and Keane (2009).

Several comparative studies, investigating the accuracy and effectiveness of various surrogate models, can be found in the literature. However, one cannot find an agreement on the dominance of one specific method over others. In an early study, Simpson et al. (1998) compared second-order response surfaces with Kriging. The metamodels were applied on a multidisciplinary design problem and four optimization problems. Jin et al. (2001) conducted a systematic comparison study of four different metamodeling techniques: polynomial regression, Kriging, multivariate adaptive regression splines and radial basis function. They used 13 mathematical test functions and an engineering test problem considering various characteristics of the sample data and evaluation criteria. They concluded that in overall RBF performed the best for both large and small scale problems with high-order of non-linearity. Fang et al. (2005) studied RSM and RBF to find the best method for modeling highly non-linear responses found in impact related problems. They also compared the RSM and RBF models with a highly non-linear test function. Despite the computation cost of RBF, they concluded dominance of RBF over RSM in such optimization problems. Mullur and Messac (2006) compared extended radial basis function (E-RBF), with three other approaches; RSM, RBF and Kriging. A number of modelling criteria including problem dimension, sampling technique, sample size and performance criteria were employed. The E-RBF was identified as the superior method since parameter setting was avoided and the method resulted in an accurate metamodel without a significant increase in computation time. Kim et al. (2009) performed a comparative study of four metamodeling techniques using six mathematical functions and evaluated the results by root mean squared error. Kriging and moving least squares showed promising results in that study. In another study by Zhao and Xue (2010), four metamodeling methods are compared by considering three characteristics of quality of the sample (sample size, uniformity and noise) and four performance measures (accuracy, confidence, robustness and

efficiency). Backlund et al. (2012) studied the accuracy of RBF, Kriging and support vector regression (SVR) with respect to their capability in approximating base functions with large number of variables and variant modality. The conclusion was that Kriging appeared to be the dominant method in its ability to approximate accurately with fewer or equivalent number of training points. Also, unlike RBF and SVR, the parameter tuning in Kriging was automatically done during training process. RBF was found to be the slowest in building the model with large number of training points. In contrast, SVR was the fastest in large scale multi-modal problems.

In most of the previously conducted comparison studies, RBF has shown to perform well in different test problem and engineering applications. Therefore, in this paper, we don't recognize a need to compare RBF with other metamodeling techniques again. Instead we focus on a detailed comprehensive comparison of our proposed RBF with a priori bias with the classical augmented RBF (RBF with a posteriori bias). The factors that are present during the construction of a metamodel (modeling criteria), range from the dimension of the problem, the type of radial basis functions used in RBF, the sampling technique and sample size. The evaluation of the modeling criteria and their affect on the accuracy, performance and robustness of a metamodel will help the designer to chose an appropriate metamodeling technique for their specific application. A recent comparison study of these two approaches have been conducted by the authors Amouzgar and Strömberg (2014). The preliminary results revealed the potential of RBF with a priori bias in predicting the test problem values. This potential is evaluated in detail in this paper for nine established mathematical test functions. A pre-study on the performance of our RBF with a priori bias in metamodel based design optimization is also performed for two benchmarks. The results clearly demonstrate that our RBF with a priori bias is most attractive for the choice of surrogate model in MDO.

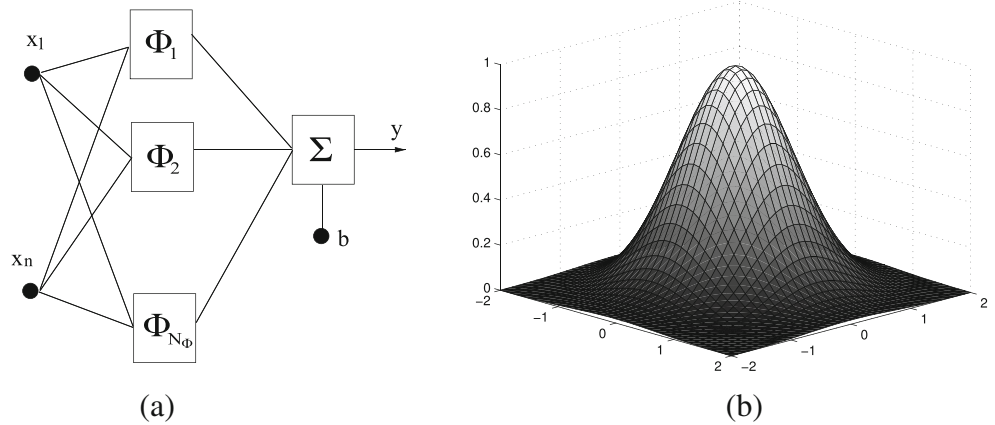
2 Radial basis functions networks

Radial basis functions were first used by Hardy (1971) for multivariate data interpolation. He proposed RBFs as approximation functions by solving multi-quadratic equations of topography based on coordinate data with interpolation. A radial basis function network of ingoing variables x_i collected in \mathbf{x} can be written as

$$f(\mathbf{x}) = \sum_{i=1}^{N_{\Phi}} \Phi_i(\mathbf{x})\alpha_i + b(\mathbf{x}), \quad (1)$$

where $f = f(\mathbf{x})$ is the outgoing response of the network, $\Phi_i = \Phi_i(\mathbf{x})$ represents the radial basis functions, N_{Φ} is the

Fig. 1 **a** A radial basis functions network; **b** the Gauss function



number of radial basis functions, α_i are weights and $b = b(\mathbf{x})$ is a bias. The network is depicted in Fig. 1.

Examples of popular radial basis functions are

$$\begin{aligned}
 \text{Linear: } & \Phi_i(r) = r, \\
 \text{Cubic: } & \Phi_i(r) = r^3, \\
 \text{Gaussian: } & \Phi_i(r) = e^{-\theta_i r^2}, \quad 0 \leq \theta_i \leq 1, \\
 \text{Quadratic: } & \Phi_i(r) = \sqrt{r^2 + \theta_i^2}, \quad 0 \leq \theta_i \leq 1,
 \end{aligned} \tag{2}$$

where θ_i represents the shape parameters and

$$r(\mathbf{x}) = \sqrt{(\mathbf{x} - \mathbf{c}_i)^T (\mathbf{x} - \mathbf{c}_i)} \tag{3}$$

is the radial distance. The shape parameters controls the width of the radial basis functions. A radial basis function with a small value of θ_i gives a narrower effect on the surrounding region. In other words, the nearby points of an unknown point will affect the prediction of the response on that point. In this case the risk of overfitting will occur, which means the sample points will influence only on a very close neighbourhood. An overfitted response surface does not capture the true function accurately, it rather describes the noise, even in noise free data sets. \mathbf{c}_i is the center point for each radial basis function. The number of center points is commonly set equal to the number of sample points. We have found that using the sample points as the center points will usually result to a more accurate model.

In this work we consider the bias to be a polynomial function, which is considered to be known either a priori or a posteriori. The bias is formulated as

$$b = \sum_{i=1}^{N_\beta} \xi_i(\mathbf{x}) \beta_i, \tag{4}$$

where $\xi_i(\mathbf{x})$ represents the polynomial basis functions and β_i are constants. N_β is the number of terms in the polynomial function.

Thus, for a particular signal $\hat{\mathbf{x}}_k$ the outcome of the network can be written as

$$f_k = f(\hat{\mathbf{x}}_k) = \sum_{i=1}^{N_\Phi} A_{ki} \alpha_i + \sum_{i=1}^{N_\beta} B_{ki} \beta_i, \tag{5}$$

where

$$A_{ki} = \Phi_i(\hat{\mathbf{x}}_k) \text{ and } B_{ki} = \xi_i(\hat{\mathbf{x}}_k). \tag{6}$$

Furthermore, for a set of signals, the corresponding outgoing responses $\mathbf{f} = \{f_i\}$ of the network can be formulated compactly as

$$\mathbf{f} = \mathbf{A}\boldsymbol{\alpha} + \mathbf{B}\boldsymbol{\beta}, \tag{7}$$

where $\boldsymbol{\alpha} = \{\alpha_i\}$, $\boldsymbol{\beta} = \{\beta_i\}$, $\mathbf{A} = [A_{ij}]$ and $\mathbf{B} = [B_{ij}]$.

2.1 Bias known a priori

We suggest to set up the RBF in (1) by treating the bias as known a priori. This is presented here. The established approach by letting the bias be unknown is presented next.

The network in (1) is trained in order to fit a set of known data $\{\hat{\mathbf{x}}_k, \hat{f}_k\}$. We assume that the number of data is N_d and we collect all \hat{f}_k in $\hat{\mathbf{f}}$. The training is performed by minimizing the error

$$\boldsymbol{\epsilon} = \mathbf{f} - \hat{\mathbf{f}} \tag{8}$$

in the least squares sense. We begin by considering this problem when the constants $\boldsymbol{\beta} = \hat{\boldsymbol{\beta}}$ are known a priori. The minimization problem then reads

$$\min_{\boldsymbol{\alpha}} \frac{1}{2} (\mathbf{A}\boldsymbol{\alpha} - (\hat{\mathbf{f}} - \mathbf{B}\hat{\boldsymbol{\beta}}))^T (\mathbf{A}\boldsymbol{\alpha} - (\hat{\mathbf{f}} - \mathbf{B}\hat{\boldsymbol{\beta}})). \tag{9}$$

The solution to this problem is given by

$$\hat{\boldsymbol{\alpha}} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T (\hat{\mathbf{f}} - \mathbf{B}\hat{\boldsymbol{\beta}}). \tag{10}$$

An obvious possibility to define $\hat{\boldsymbol{\beta}}$ a priori, which is used

in this work, is to use the following optimal regression coefficients:

$$\hat{\beta} = (B^T B)^{-1} B^T \hat{f}. \tag{11}$$

2.2 Bias known a posteriori

If the bias is considered not to be known a priori, then (9) modifies to

$$\min_{(\alpha, \beta)} \frac{1}{2} (A\alpha + B\beta - \hat{f})^T (A\alpha + B\beta - \hat{f}). \tag{12}$$

Furthermore, if we also assume that $N_\Phi + N_\beta > N_d$, then the following orthogonality constraint is introduced:

$$\sum_{i=1}^{N_\Phi} \xi_j(c_i) \alpha_i = 0, \quad j = 1, \dots, N_\beta. \tag{13}$$

This can be written on matrix format as

$$R^T \alpha = 0, \tag{14}$$

where

$$R = [R_{ij}], \quad R_{ij} = \xi_j(c_i). \tag{15}$$

In conclusion, for a bias known a posteriori, we have to solve the following problem:

$$\begin{cases} \min_{(\alpha, \beta)} \frac{1}{2} (A\alpha + B\beta - \hat{f})^T (A\alpha + B\beta - \hat{f}) \\ \text{s.t. } R^T \alpha = 0. \end{cases} \tag{16}$$

The corresponding Lagrangian function is given by

$$\mathcal{L}(\alpha, \beta, \lambda) = \frac{1}{2} (A\alpha + B\beta - \hat{f})^T (A\alpha + B\beta - \hat{f}) + \lambda^T R^T \alpha. \tag{17}$$

The necessary optimality conditions become

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \alpha} &= A^T (A\alpha + B\beta - \hat{f}) + R\lambda = 0, \\ \frac{\partial \mathcal{L}}{\partial \beta} &= B^T (A\alpha + B\beta - \hat{f}) = 0, \\ \frac{\partial \mathcal{L}}{\partial \lambda} &= R^T \alpha = 0. \end{aligned} \tag{18}$$

The optimality conditions in (18) can also be written on matrix format as

$$\begin{bmatrix} A^T A & A^T B & R \\ B^T A & B^T B & 0 \\ R^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \lambda \end{bmatrix} = \begin{bmatrix} A^T \hat{f} \\ B^T \hat{f} \\ 0 \end{bmatrix}. \tag{19}$$

By solving this system of equations, the radial basis function network with a bias known a posteriori is established.

If the center points c_i are chosen to be equal \hat{x}_i , then $R = B$, the network becomes an interpolation and (19) can be reduced to

$$\begin{bmatrix} A & B \\ B^T & 0 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \hat{f} \\ 0 \end{bmatrix}. \tag{20}$$

This is the established approach in setting the RBF in (1). We suggest one can simply use (10) and (11), which are nothing more than two normal equations; (10) is the normal equation to (9) and (11) is the normal equation to the corresponding regression problem. Obviously, the two approaches will produce different RBFs. This is demonstrated in Fig. 2, where the biases are compared using both approaches for the same benchmark problem. In the following the performance of these two approaches are studied in detail. Further on in the present paper, RBF with the bias known a posteriori is briefly called a *posteriori RBF* and abbreviated by RBF_{pos} , and radial basis functions with bias known a priori is called a *a priori RBF* and abbreviated by RBF_{pri} .

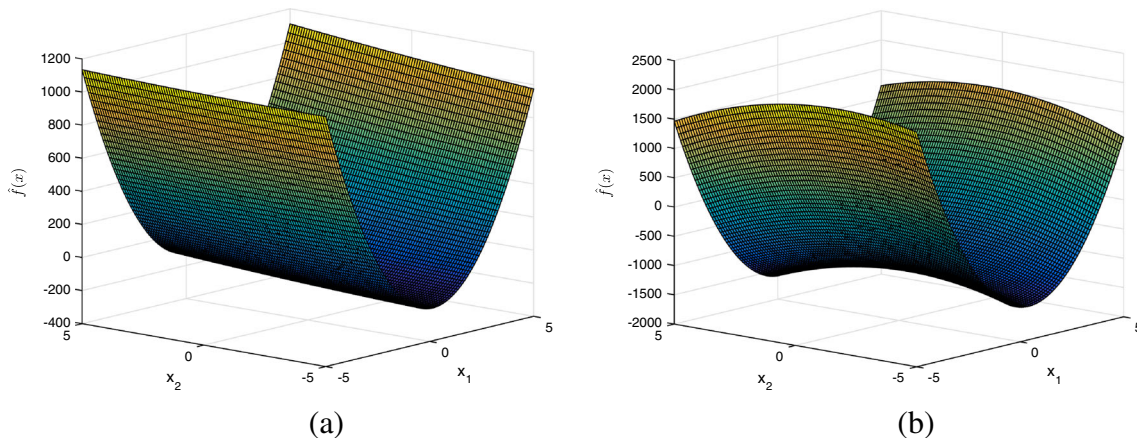


Fig. 2 The bias plotted for test function 4 using both approaches: (a) RBF with a priori known bias, (b) RBF with a posteriori known bias

3 Test functions

The comparison of the two RBF approaches are based on 9 different mathematical test functions presented below. These test functions are commonly used as benchmarks for unconstrained global optimization problems.

1. Branin-Hoo function (Branin 1972)

$$f_1 = 2(x_2 - \frac{5.1x_1^2}{4\pi^2} + \frac{5x_1}{\pi} - 6) + 10(1 - \frac{1}{8\pi}) \cos(x_1) + 10. \tag{21}$$

2. Goldstein-Price function (Goldstein and Price 1971)

$$f_2 = (1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)) \times (30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)). \tag{22}$$

3. Rastrigin function

$$f_3 = 20 + \sum_{i=1}^N x_i^2 - 10 \cos(2\pi x_i). \tag{23}$$

In this study, the Rastrigin function with 2 variables is used (N=2).

4. Three-Hump Camel function

$$f_4 = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1x_2 + x_2^2. \tag{24}$$

5. Colville function

$$f_5 = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1). \tag{25}$$

6. Math 1

$$f_6 = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7. \tag{26}$$

7. Rosenbrock-10 function (Rosenbrock 1960)

$$f_7 = \sum_{n=1}^{N-1} (100(x_{n+1} - x_n^2)^2 + (x_n - 1)^2). \tag{27}$$

In this study, the Rosenbrock function with 10 variables is used (N=10).

8. Math 2 (A 10-variable mathematical function)

$$f_8 = \sum_{m=1}^{10} \left(\frac{3}{10} + \sin(\frac{16}{15}x_m - 1) + \sin(\frac{16}{15}x_m - 1)^2 \right). \tag{28}$$

9. Math 3 (A 16-variable mathematical function) (Jin et al. 2001)

$$f_9 = \sum_{m=1}^{16} \sum_{n=1}^{16} a_{mn}(x_m^2 + x_m + 1)(x_n^2 + x_n + 1), \tag{29}$$

where *a* is defined in Jin et al. (2001).

The properties of the test functions are summarized in Table 1.

4 Modelling and performance criteria for comparison

Standard statistical error analysis is used to evaluate the accuracy of the the two RBF approaches. Details of this analysis are presented in this section.

4.1 Performance metrics

The two standard performance metrics are applied to the off-design test points: (i) Root Mean Squared Error (RMSE) and (ii) Maximum Absolute Error (MAE). The lower the RMSE and MAE values, the more accurate the metamodel will be. The aim is to have these two error measures as near to zero as possible.

Table 1 Mathematical test functions

Function	Function name	No. of variables	Design range(s)
<i>f</i> ₁	Branin-Hoo	2	<i>x</i> ₁ : [-5, 10], <i>x</i> ₂ : [0, 15]
<i>f</i> ₂	Goldstein-Price	2	<i>x</i> ₁ , <i>x</i> ₂ : [-2, 2]
<i>f</i> ₃	Rastrigin	2	<i>x</i> ₁ , <i>x</i> ₂ : [-5.12, 5.12]
<i>f</i> ₄	Three-Hump Camel	2	<i>x</i> ₁ , <i>x</i> ₂ : [-5, 5]
<i>f</i> ₅	Colville	4	<i>x</i> _{<i>i</i>} : [-10, 10], <i>i</i> = 1, 2...4
<i>f</i> ₆	Math 1	7	<i>x</i> _{<i>i</i>} : [-10, 10], <i>i</i> = 1, 2...7
<i>f</i> ₇	Math 2	10	<i>x</i> _{<i>i</i>} : [-1, 1], <i>i</i> = 1, 2...10
<i>f</i> ₈	Rosenbrock-10	10	<i>x</i> _{<i>i</i>} : [-5, 10], <i>i</i> = 1, 2...10
<i>f</i> ₉	Math 3	16	<i>x</i> _{<i>i</i>} : [-1, 1], <i>i</i> = 1, 2...16

The RMSE is calculated by

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (\hat{f}_i - f_i)^2}{n}} \quad (30)$$

and MAE is defined by

$$\text{MAE} = \max|\hat{f}_i - f_i|, \quad (31)$$

where n is the number of off-design test points selected to evaluate the model, \hat{f}_i is the exact function value at the i_{th} test point and f_i represents the corresponding predicted function value.

RMSE and MAE are typically at the same order of the actual function values. These error measures will not indicate the relative performance quality of the RBFs across different functions independently. Therefore, to compare the performance measures of the two approaches over test functions, the normalized values of the two errors, NRMSE and NMAE, by using the actual function values, are calculated by

$$\text{NRMSE} = \sqrt{\frac{\sum_{i=1}^n (\hat{f}_i - f_i)^2}{\sum_{i=1}^n (\hat{f}_i)^2}}, \quad (32)$$

$$\text{NMAE} = \frac{\max|\hat{f}_i - f_i|}{\sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{f}_i - \bar{f}_i)^2}}, \quad (33)$$

where \bar{f} denotes the mean of the actual function values at the test points.

In addition, the NRMSE and NMAE of a priori RBF is compared to the a posteriori RBF approach by defining the corresponding relative differences. The relative difference in NRMSE (D^{NRMSE}) of a posteriori RBF is given by

$$D_{RBF_{pos}}^{NRMSE} = \frac{NRMSE_{RBF_{pos}} - NRMSE_{RBF_{pri}}}{NRMSE_{RBF_{pri}}} \times 100\%, \quad (34)$$

and the relative difference in NMAE (D^{NMAE}) of a posteriori RBF is defined by

$$D_{RBF_{pos}}^{NMAE} = \frac{NMAE_{RBF_{pos}} - NMAE_{RBF_{pri}}}{NMAE_{RBF_{pri}}} \times 100\%, \quad (35)$$

where NRMSE and NMAE values of the RBF_{pos} approach are referred by $NRMSE_{RBF_{pos}}$ and $NMAE_{RBF_{pos}}$; and $NRMSE_{RBF_{pri}}$ and $NMAE_{RBF_{pri}}$ are the corresponding NRMSE and NMAE values of the RBF_{pri} approach.

4.2 Radial basis functions

Several different radial basis functions can be used in constructing the RBF, as mentioned in Section 2. Each will yield to a different result depending to the nature of the

problem. However, in real world applications, the mathematical properties of the problem is usually not known in advance. Thus, a designer needs a robust choice of radial basis function which is as independent as possible to the nature of the problem and will result to a acceptably accurate metamodel. In this paper, four different radial basis functions: (i) linear, (ii) cubic, (iii) Gaussian, and (iv) quadratic, formulated in (2), are used to study the effect of radial basis functions on the accuracy of metamodels.

4.3 Sampling techniques

Sampling techniques are used to create DoEs for which the particular RBF then is fitted to. A robust sampling technique is desired for a designer to avoid dependencies to sampling techniques, as much as possible, for different problems. In other words, one would like to have a meta-modeling technique that is as independent as possible to the sampling technique. In this study, three different sampling techniques are chosen: (i) Random sampling (RND), (ii) Latin hypercube sampling (LHS) and (iii) Hammersley sequence sampling (HSS) and their effects on the accuracy of the two approaches are investigated. For the optimization problems studied at the end, we also compare these sampling techniques to a successive screening approach for generating appropriate DoEs.

In random sampling, a desired set of uniformly distributed random numbers within the variable bounds of each test function is chosen. As expected there is no uniformity in the created set of DoE. The Latin hypercube sampling technique creates samples that are relatively uniform in each single dimension while subsequent dimensions are randomly paired to fit a m-dimensional cube. LHS can be regarded as a constrained Monte Carlo sampling scheme developed by McKay et al. (1979) specifically for computer experiments. Hammersley sequence sampling produces more uniform samples over the m-dimensional space than LHS. This can be seen in Fig. 3 which illustrates the uniformity of a set of 15 sample points over a unit square, using RND, LHS and HSS. Hammersley sequence sampling uses a low discrepancy sequence (Hammersley sequence) to uniformly place N points in a m-dimensional hypercube given by the following sequence:

$$Z_m(n) = \left(\frac{n}{N}, \phi_{R_1}(n), \phi_{R_2}(n), \dots, \phi_{R_{m-1}}(n) \right), \quad n = 1, 2, \dots, N, \quad (36)$$

where R_1, R_2, \dots, R_{m-1} are the first $m - 1$ prime numbers. $\phi_R(n)$ are constructed by reversing the order of the digits of any integer, written in radix-R notation, around the decimal points. In this work, HSS is coded in Matlab based on the theory in the original paper by Kalagnanam and Diwekar

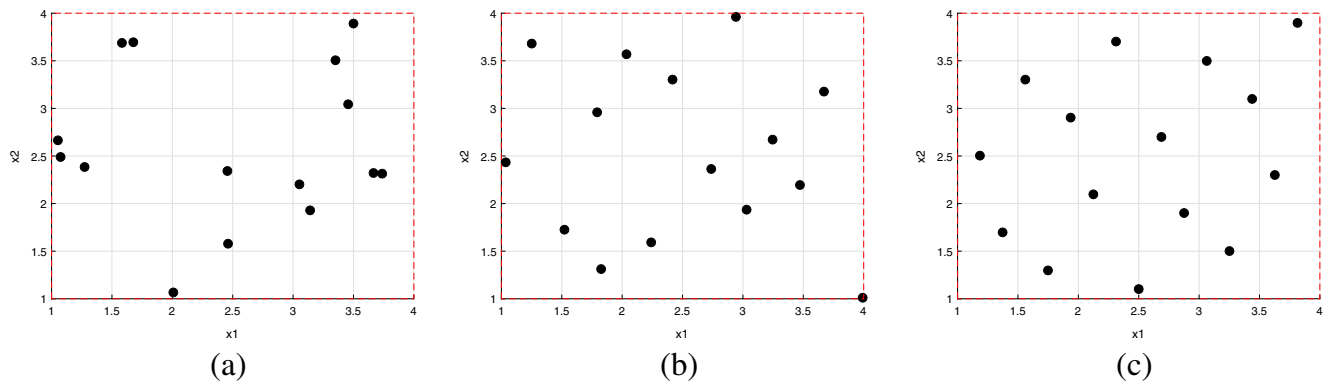


Fig. 3 Uniformity of different sampling techniques: (a) RND, (b) LHS, (c) HSS

(1997), where detailed definition and theory of Hammersley points can be found.

4.4 Size of samples

The DoE size (size of samples) has an important effect on choosing an accurate surrogate model. In general, increasing the size of DoE will improve the quality of metamodels when using the RBF approach, however over fitting is a critical issue in these approaches. Three different sample size are used in this paper: (i) Low, (ii) Medium and (iii) High. The number of samples for each sampling group is proportional to a reference value for low and high dimension problems. The number of coefficients $k = (m+1)(m+2)/2$ in a second order polynomial with m number of variables is used as a reference. For all the test functions the number of DoE is chosen as a coefficient of k . The sample size for low dimension test function are: (i) $1.5k$ for low sample size, (ii) $3.5k$ for medium sample size equals, and (iii) $6k$ for high sample size. High dimensional test functions have the size of DoE defined as: i) $1.5k$ for low sample size, (ii) $2.5k$ for medium sample size, and (iii) $5k$ for high sample size.

4.5 Test functions dimensionality

Dimension of a test function or the number of the variables in a problem, is one of the most important properties in generating an accurate surrogate model. In order to investigate the effect of this modelling criteria on the two approaches we divided the test functions into two categories: (i) Low, where the number of variables are less than or equal to 4, and (ii) High, for test functions with the number of variables of more than 4. Labelling the second group by “high”, implies the relative meaning of higher number of variables compared to the first group. Otherwise, high dimensional engineering problems generally consist of considerable higher number of variables. The results are grouped separately for low and high dimension test

functions in all modeling criteria. Our goal is that a final conclusion can be drawn by studying the results.

5 Comparison procedure

In this section, we describe the procedure used to compare the two metamodeling approaches (RBF_{pri} , RBF_{pos}) under multiple modeling criteria mentioned in previous sections. The comparison is based on the 9 mathematical test functions and the performance metrics described in previous sections. We summarize the comparison procedure into 6 steps as follow:

- Step 1: The number of DoEs is determined based on the three sample size groups (low, medium and high) in Table 2, for each test function.
- Step 2: The design domains are mapped linearly between 0 and 1 (unit hypercube). The surrogate models are fitted on the mapped variables by using the two approaches. For calculating the performance metrics the metamodel is mapped back to the original space.
- Step 3: To avoid any probable sensitivity of metamodels to a specific DoE, 50 distinctive sample sets are generated for each sample size of step 1 by using RND and LHS described in the previous section. The sensitivity of surrogate models to a specific DoE is avoided to a great extent. Since HSS technique is deterministic, only one sample set is generated by using this method for each sample size. The Latin Hypercube sampling techniques (LHS) is performed by using the Matlab function “lhsdesign”. The Latin hypercube samples are created with 20 iterations to maximize the minimum distance between points. 50 different sets of sample points are created for each sample size by using the LHS and the random (RND) sampling technique. The Hammersely (HSS) samples, are created from Hammersley quasirandom sequence using successive primes as bases by using an in-house Matlab code.

Table 2 Modeling criteria of test functions

Function	Function name	No. of variables	Problem dimension	Sample size			No. of test points
				Low	Medium	High	
f_1	Branin-Hoo	2	Low	9	30	60	1000
f_2	Goldstein-Price	2	Low	9	30	60	1000
f_3	Rastrigin	2	Low	9	30	60	1000
f_4	Three-Hump Camel	2	Low	9	30	60	1000
f_5	Colville	4	Low	23	75	150	1000
f_6	Math 1	7	High	54	90	180	1000
f_7	Math 2	10	High	99	165	330	1000
f_8	Rosenbrock-10	10	High	99	165	330	1000
f_9	Math 3	16	High	229	380	765	1000

- Step 4: Metamodels are constructed using the two RBF approaches (RBF_{pri} and RBF_{pos}) with each of the four different radial basis functions (linear, cubic, Gaussian and quadratic) to be compared for each set of DoE generated by the three sampling techniques. Therefore, for each test function 2 (RBF approaches) \times 4 (radial basis functions) \times 3 (sampling techniques) \times 3 (sample sizes) \times 50 (set of DoE) = 3600 surrogate models are constructed.
- Step 5: 1000 test points are randomly selected within the design space. The exact function value \hat{f}_i and the predicted function value f_i at each test point is calculated. RMSE, MAE, and the corresponding normalized values are computed by using (30) to (33). The average of the normalized errors is calculated across the 50 sample sets. The average of the normalized root mean squared and maximum absolute error are simply shown by $NRMSE$ and $NMAE$ in this paper. Finally, the relative difference measures of the computed average errors, $NRMSE$ and $NMAE$ for RBF_{pos} are calculated by using (34) and (35).
- Step 6: The procedure from step 1 to 5 is repeated for all test problems. In addition to the mean normalized errors ($NRMSE$ and $NMAE$), the average of low dimension problems (the first five test functions) denoted by “Ave. Low”, the average of high dimension problems (test functions 6 to 9) expressed by “Ave. High” and the average error metrics of all 9 test functions shown by “Ave. All” are computed for the surrogate approaches using different sampling techniques.

It should be noted that, because the variables are mapped to a unit cube (in step 2), the parameter setting can be done without considering the magnitude of the design variables. Thus, the parameter θ used in the radial basis functions in (2) is set to one ($\theta = 1$). The bias chosen for this study, in (4), is a quadratic polynomial with 6 terms.

6 Results and discussion

In this section, the results gathered from the metamodels constructed according to the comparison procedure in previous section are presented. The effect of each modeling criteria is discussed by comparing the two main error measures, $NRMSE$ and $NMAE$, for the two RBF approaches by presenting them in several tables and charts. Including all modeling criteria in the comparison study of each criteria for all test functions requires an extensive and very detailed results section which should incorporate all 3600 surrogate models. This is out of scope of this work and can be the topic of future studies. Therefore, for studying the effect of each modeling criteria, a specific selection of other criteria is chosen. They are mentioned in the forthcoming sections.

Before presenting the results, it is worth mentioning that the computational cost of the proposed RBF_{pri} is less than RBF_{pos} . This has been investigated by calculating the training time of the two approaches for test functions 3 and 8 with 100 variables and 15453 sampling points by using the cubic radial basis function and HSS sampling method. The computational times related to f_3 are 346.67 and 396.97 seconds for RBF_{pri} and RBF_{pos} , respectively. Test function 8 is trained in 350.48 and 591.76 seconds by using RBF_{pri} and RBF_{pos} , respectively.

6.1 Effect of basis functions

Table 3 shows the $NRMSE$ and $NMAE$ values of high sample size and LHS sampling technique of RBF_{pri} and RBF_{pos} by using the four different basis functions across all test problems. The bold faced values highlight the lowest errors for each test function. It can be seen that the minimum errors are varied through different basis functions between the test functions. However, cubic basis function results in lower values in both $NRMSE$ and $NMAE$ for f_1 , f_8 and f_9 .

Table 3 NRMSE and NMAE (LHS sampling with high sample size)

Test function	RBF approach	NRMSE				NMAE			
		Linear	Cubic	Gaussian	Quadratic	Linear	Cubic	Gaussian	Quadratic
f_1	RBF_{pri}	0.1908	0.0952	0.4538	0.1349	1.8655	0.8358	6.0043	1.5904
	RBF_{pos}	0.1951	0.0975	0.1979	0.4017	2.5765	0.9711	2.5429	6.3368
f_2	RBF_{pri}	0.3158	0.2594	0.1874	0.1620	3.0204	2.4426	1.9849	1.5635
	RBF_{pos}	0.3735	0.2496	0.3674	0.1769	3.6027	2.4930	3.5192	1.8219
f_3	RBF_{pri}	0.3080	0.4122	8.1544	8.1544	2.3752	4.7238	225.677	87.9028
	RBF_{pos}	0.3067	0.4162	0.3078	10.9940	2.4914	4.2085	2.4926	312.287
f_4	RBF_{pri}	0.3409	0.2634	0.1184	0.1521	2.0918	1.7894	1.3465	1.4218
	RBF_{pos}	0.4612	0.2709	0.4538	0.1473	3.0001	1.9679	3.0208	1.6017
f_5	RBF_{pri}	0.2012	0.1967	0.1590	0.1752	1.2435	1.2760	1.4980	1.3941
	RBF_{pos}	0.3220	0.2146	0.3219	0.1767	2.4984	1.5238	2.4598	1.6206
f_6	RBF_{pri}	0.4469	0.5012	0.6332	0.5617	2.1897	2.5543	3.3058	2.9341
	RBF_{pos}	0.6063	0.5254	0.7355	0.6154	3.1961	2.8564	4.1352	3.3001
f_7	RBF_{pri}	0.1249	0.1185	0.1247	0.1178	3.1090	2.9903	3.3601	3.0411
	RBF_{pos}	0.1162	0.1175	0.1162	0.1141	2.7253	2.9346	2.7001	2.9427
f_8	RBF_{pri}	0.1683	0.1646	0.1741	0.1659	1.1983	1.2398	1.3317	1.2620
	RBF_{pos}	0.1842	0.1653	0.1847	0.1697	1.4382	1.2617	1.5064	1.3617
f_9	RBF_{pri}	0.0211	0.0190	0.0215	0.0196	0.4572	0.3441	0.4860	0.3834
	RBF_{pos}	0.0329	0.0209	0.0388	0.0248	1.0555	0.4795	1.5252	0.7643

Also by studying and comparing the results obtained from all 3600 constructed metamodels one can conclude that the cubic basis function is the preferred choice, because of its robust behaviour under different criteria, when there is no prior knowledge on the mathematical property of the problem. This may be because of lack of any extra parameter in the cubic radial basis function. Thus, parameter setting and

finding the optimal shape parameter is not required in cubic radial basis function.

It is cumbersome to compare the two approaches under each modeling criteria by using all the radial basis functions. Therefore, for each test function and modeling criteria a radial basis function is chosen, and the two metamodels are constructed by using that radial basis function. Table 4

Table 4 Summary of chosen basis functions

Test function	Sampling technique	Sample size	Problem dimension	Overall accuracy
f_1	Cubic	Cubic	Cubic	Cubic
f_2	Quadratic	Quadratic	Quadratic	Quadratic
f_3	Linear	Cubic	Cubic	Linear
f_4	Cubic	Cubic	Cubic	Quadratic
f_5	Quadratic	Cubic	Cubic	Cubic
f_6	Cubic	Cubic	Cubic	Cubic
f_7	Cubic	Cubic	Cubic	Cubic
f_8	Cubic	Cubic	Cubic	Cubic
f_9	Cubic	Cubic	Cubic	Cubic

summarizes the chosen radial basis functions for each test function and modeling criteria.

In cases where the best performed basis function is different for the two approaches under a modeling criteria, the basis function which performed better by using the RBF_{pos} is selected. This will enable a more reliable comparison between the two approaches.

6.2 Effect of sampling technique

The error measures of surrogate models constructed by the two approaches using the three sampling techniques are shown in Table 5. The values are extracted based on the basis functions chosen according to Table 4. Figure 4 depicts a summary of the results in Table 5 by comparing the performance metrics of “Ave. Low”, “Ave. High” and “Ave. All” rows. By observing the NRMSE values in Fig. 4a, the lowest errors for both approaches correspond

to the HSS technique followed by the LHS and the random sampling technique which has the highest values for NRMSE. The only exceptions where the LHS generate a better metamodel, are in test function 3 (f_3) and the last high dimensional test function (f_9). Considering the NMAE values in Fig. 4b, the HSS method yields to the lowest errors. Also, the low dimension problems perform better with random sampling technique in comparison to LHS technique. Both RBF_{pri} and RBF_{pos} perform better when LHS sampling technique is used in high dimension problems (Fig. 4), however this gain is marginal compared to the two other techniques.

The “ave. all” bars in Fig. 4a and b along with the data in Table 5, show 4.7 % and 6.8 % improve in NRMSE and NMAE when using HSS technique instead of LHS in RBF_{pri} approach. While these values are 11.1 % and 14.3 % for the RBF_{pos} approach. The advantage of RBF_{pri} over RBF_{pos} , as being more robust in terms of NRMSE and

Table 5 NRMSE and NMAE of each sampling technique (high sample size)

Test function	RBF approach	NRMSE			NMAE		
		RND	LHS	HSS	RND	LHS	HSS
f_1	RBF_{pri}	0.1171	0.0952	0.0747	1.0179	0.8358	0.8646
	RBF_{pos}	0.1221	0.0975	0.0752	1.1759	0.9711	1.3284
f_2	RBF_{pri}	0.2365	0.1620	0.1397	2.0912	1.5635	1.9509
	RBF_{pos}	0.3110	0.1769	0.1602	3.1755	1.8219	2.3215
f_3	RBF_{pri}	0.3164	0.3080	0.3144	2.5913	4.7238	2.1904
	RBF_{pos}	0.3117	0.3067	0.3098	2.5876	4.2085	2.4636
f_4	RBF_{pri}	0.3250	0.2634	0.1126	2.2031	1.7894	0.7961
	RBF_{pos}	0.3270	0.2709	0.1260	2.2347	1.9679	1.1646
f_5	RBF_{pri}	0.1853	0.1752	0.1649	1.3390	1.2760	1.3270
	RBF_{pos}	0.1863	0.1767	0.1730	1.6783	2.8564	1.6052
Average Low	RBF_{pri}	0.2361	0.2008	0.1613	1.8485	2.0377	1.4258
	RBF_{pos}	0.2516	0.2058	0.1688	2.1704	2.3652	1.7767
f_6	RBF_{pri}	0.5021	0.5012	0.4839	2.5461	2.5543	2.6711
	RBF_{pos}	0.5300	0.5254	0.5090	2.8477	2.8564	2.7938
f_7	RBF_{pri}	0.1196	0.1185	0.1138	3.0491	2.9903	2.9283
	RBF_{pos}	0.1186	0.1175	0.1134	2.9861	2.9346	2.8446
f_8	RBF_{pri}	0.1669	0.1646	0.1586	1.2700	1.2398	1.3377
	RBF_{pos}	0.1674	0.1653	0.1615	1.2915	1.2617	1.4356
f_9	RBF_{pri}	0.0192	0.0190	0.1586	0.3513	0.3441	2.0628
	RBF_{pos}	0.0209	0.0209	0.0233	0.4878	0.4795	0.6263
Average High	RBF_{pri}	0.2020	0.2008	0.2287	1.8041	1.7821	2.2500
	RBF_{pos}	0.2092	0.2073	0.2018	1.9032	1.8831	1.9251
Average all	RBF_{pri}	0.2209	0.2008	0.1913	1.8288	1.9241	1.7921
	RBF_{pos}	0.2328	0.2064	0.1835	2.0517	2.1509	1.8426

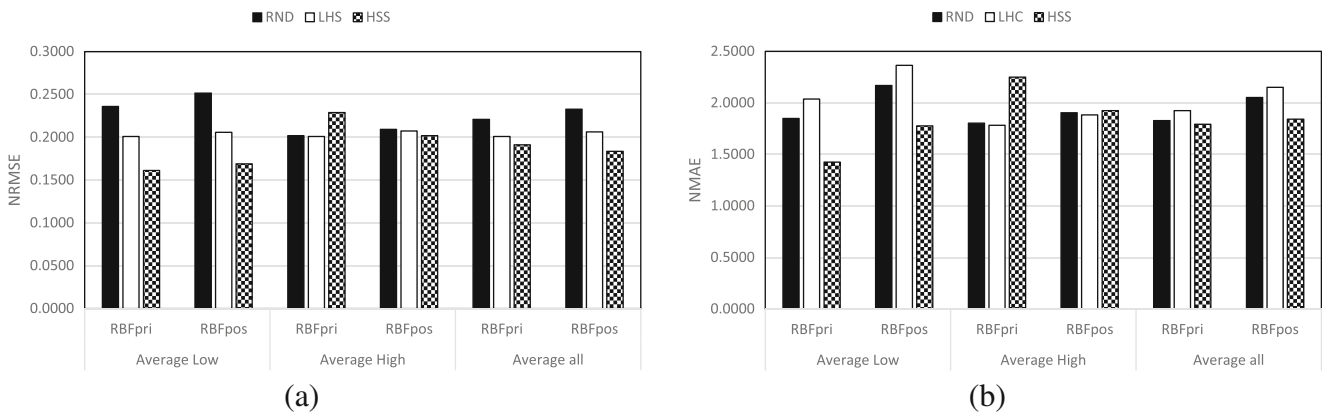


Fig. 4 Comparison of different sampling techniques: (a) normalized root mean squared error (RMSE); (b) normalized maximum absolute error (NMAE)

NMAE with regards to the change of sampling technique can be seen in the aforementioned percentages.

6.3 Effect of sampling size

Figure 5a depicts the NRMSE values of “Ave. Low”, “Ave. High” and “Ave. all” for the three different sample sizes by using the LHS sampling technique, while Fig. 5b shows the similar NMAE values. Both metamodeling approaches will improve in quality with increasing sample size, regardless of the problem’s dimension. Table 6 shows the relative differences (in percentage) of NMRSE and NMAE comparing the RBF_{pri} and RBF_{pos} with regards to the different sample size. The RBF_{pos} approach performs better than RBF_{pri} with low sample size considering both error metrics, which the negative percentage values in Table 6 reveals the exact degree of superiority. However, by increasing the sample size to medium and high the performance changes and RBF_{pri} appears to be the dominant approach. This advantage is noticeable in the NMAE values, in contrast to the marginal improvement of NRMSE values when

using RBF_{pri} . Specially in high dimensional problems with medium sample size the relative difference is less than one percent (0.74 %). By looking at the “Ave. All” row of Table 6 we can observe a 13.2 % improvement in NMAE value by using RBF_{pri} with high sample size and 4 % better accuracy in NRMSE value.

6.4 Effect of dimension

The effect of test function’s dimension on metamodel performance can be studied by summarizing the average NRMSE and NMAE values of low and high dimension problems in Table 7. The values are obtained by averaging the “Ave. Low” and “Ave. High” rows over all sampling techniques in Table 5. Considering the NRMSE, the RBF_{pri} approach performs better with low dimensional problems, while the RBF_{pos} approach generates better performance metrics in high dimensional problems. Although the advantage of RBF_{pos} in high dimensional problem compared to low dimensional is only around 1 % for NRMSE, it will increase to approximately 10 % for the

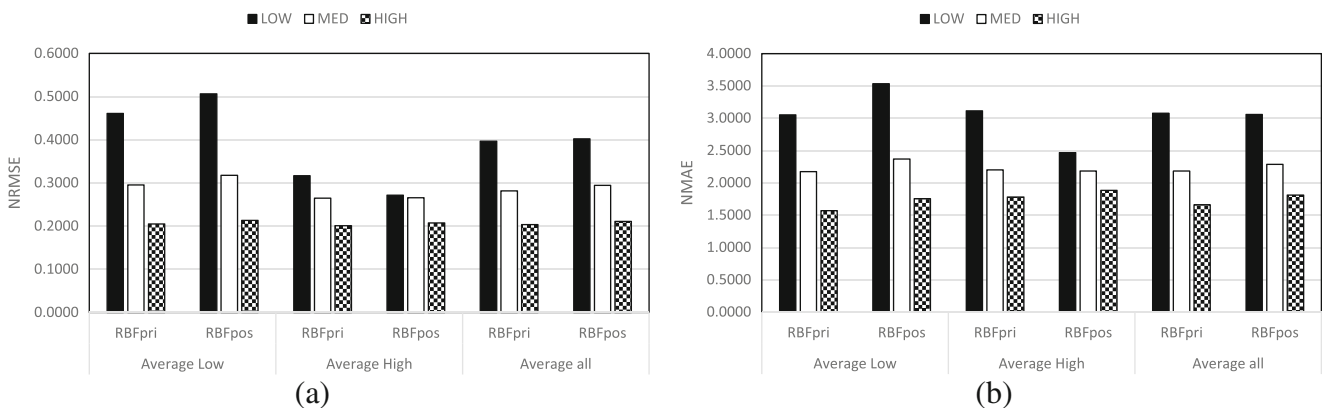


Fig. 5 Comparison of different sample size using LHS technique: (a) normalized root mean squared error (RMSE); (b) normalized maximum absolute error (NMAE)

Table 6 Relative differences of NRMSE and NMAE comparing RBF_{pri} and RBF_{pos} considering sampling size

Error metrics	$D_{NRMSE}(\%)$			$D_{NMAE}(\%)$		
	LOW	MED	HIGH	LOW	MED	HIGH
Average Low	9.10	6.64	4.47	16.36	9.68	13.40
Average high	-17.61	0.74	3.41	-12.84	6.83	12.77
Average All	-2.77	4.02	4.00	3.38	8.42	13.12

NMAE metric. The RBF_{pri} has a superiority in performance of around 9.5 % in problems with low dimension in comparison to high dimensional problems considering NMAE. In addition, the last two rows in Table 7 compares the performance of RBF_{pri} with RBF_{pos} for low and high dimension problems separately. The results confirm the advantage of RBF_{pri} over RBF_{pos} in low dimensional problems with the superiority degree of 4.6 % and 17.2 % with regards to NRMSE and NMAE respectively. On the other hand, the better performance of RBF_{pos} compared to RBF_{pri} in high dimensional test functions is marginal and are around 2 % for both NRMSE and NMAE.

6.5 Overall accuracy

For the test functions with two input variables (the first four test functions) the three-dimensional surface plots are shown in Figs. 6–9, respectively. The plots depict the actual function and the corresponding metamodels constructed by using RBF_{pri} and RBF_{pos} . The metamodel surfaces in Figs. 6, 7, 8 and 9 are generated by using the same set of DoE, created with HSS technique and high sample size, for each test function. The overall accuracy comparison of RBF_{pri} and RBF_{pos} can be studied by observing the surface plot figures and using Table 8. The table presents the relative difference in NRMSE and NMAE (as percentage) and the average values of low and high dimensional of all test functions. The values are extracted for each metamodeling approach by using the basis function mentioned in Table 4, the three sampling technique with high sample size. With regards to NRMSE relative differences, 6 out of 9 test functions by using LHS and 7 out of 9 test functions

by using RND and HSS have a positive percentage, which clearly reveals the advantage of the new approach over RBF_{pos} . This advantage is more recognizable for NMAE, with 8 test functions with a positive relative difference values for all sampling techniques. The average rows show approximately 3 % related to RND and LHS and 16 % related to HSS better performance of RBF_{pri} in NRMSE values regardless of the dimension of the test functions, while this superiority is around 16 %, 13 % and 23 % when considering the NMAE for RND, LHS and HSS respectively. This difference, demonstrates the leverage of RBF_{pri} in predicting the local deviations of functions, which is provided by MAE metric. On the other hand, the superiority of RBF_{pri} in measuring the global error, by using RND and LHS, is minor compared to the RBF_{pos} approach.

7 Optimization examples

RBF is a most attractive choice for surrogate models in metamodel based design optimization. This is demonstrated here by studying two examples using our approach of RBF with a priori bias. We begin our study with the following non-linear example:

$$\begin{cases} \min_{x_i} \sqrt{1000 \left(\frac{4}{x_1} - 2\right)^4 + 1000 \left(\frac{4}{x_2} - 2\right)^4} \\ \text{s.t. } (x_1 - 0.5)^4 + (x_2 - 0.5)^4 - 2 \leq 0. \end{cases} \quad (37)$$

The analytical optimal solution is (1.5, 1.5) and the minimum of the unconstrained objective function is found at (2, 2). The objective function is plotted in Fig. 10.

Table 7 NRMSE, NMAE and their related relative differences values averaged over all sampling techniques

Performance metrics	RBF approach	Average Low	Average high
NRMSE	RBF_{pri}	0.1994	0.2105
	RBF_{pos}	0.2087	0.2061
NMAE	RBF_{pri}	1.7707	1.9454
	RBF_{pos}	2.1041	1.9038
$D_{NRMSE}(\%)$	RBF_{pos}	4.59	-2.12
$D_{NMAE}(\%)$	RBF_{pos}	17.21	-2.16

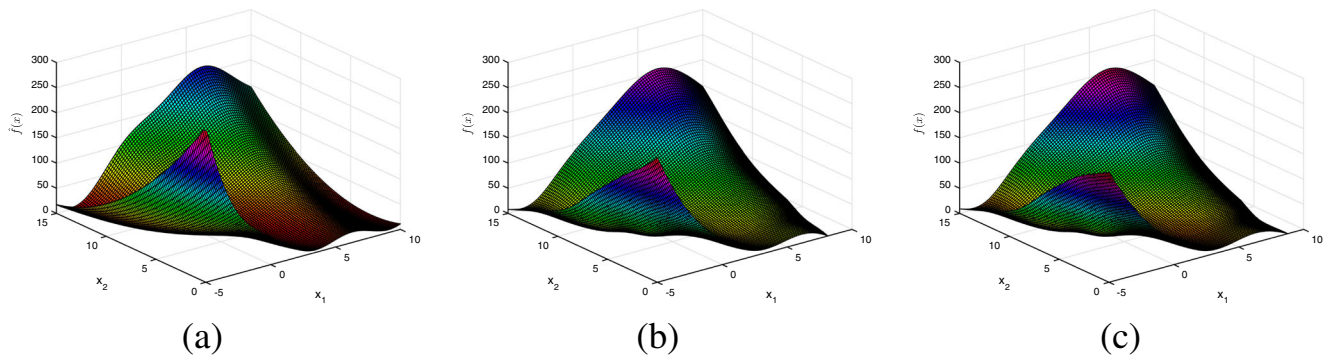


Fig. 6 Test function 1: Branin function (a) actual function; (b) RBF_{pri} ; (c) RBF_{pos}

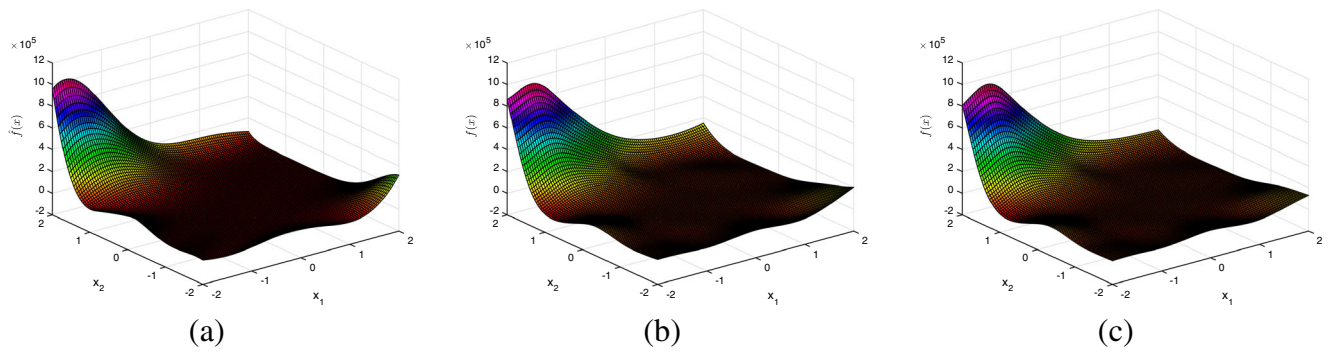


Fig. 7 Test function 2: Goldstein-Price function (a) actual function; (b) RBF_{pri} ; (c) RBF_{pos}

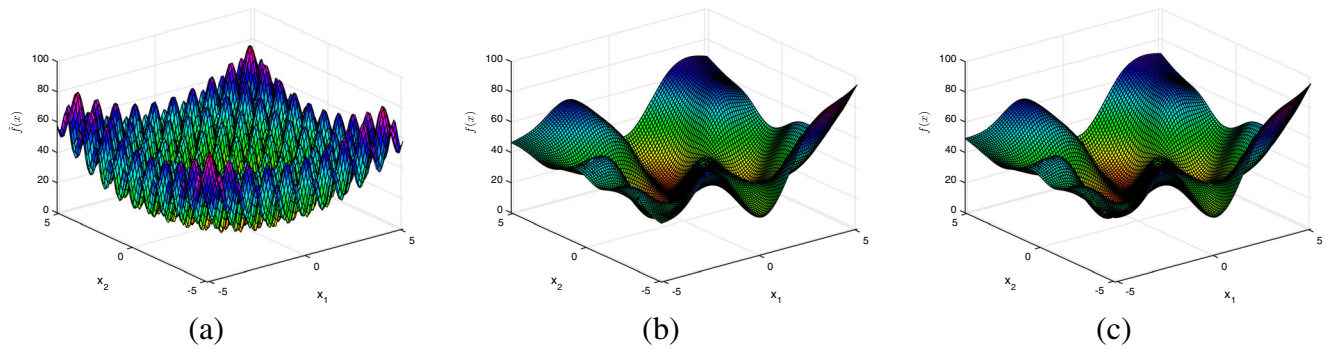


Fig. 8 Test function 3: Rastrigin function (a) actual function; (b) RBF_{pri} ; (c) RBF_{pos}

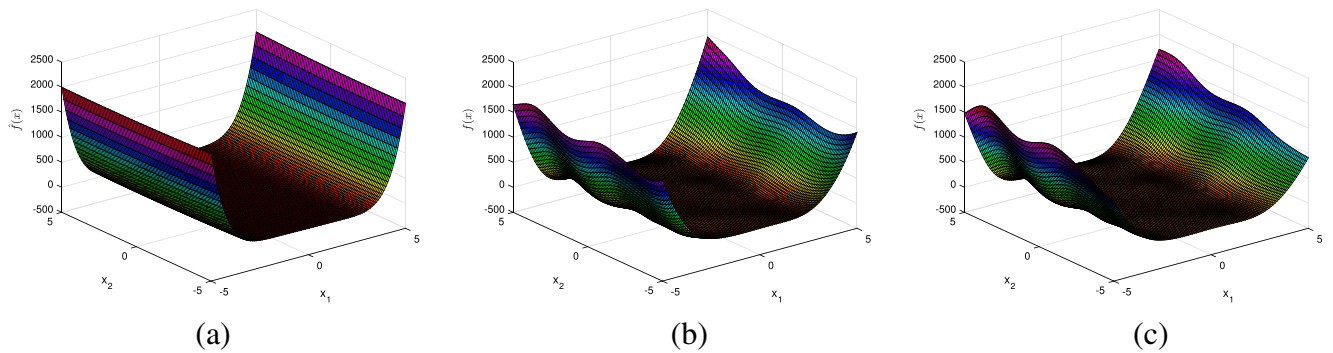
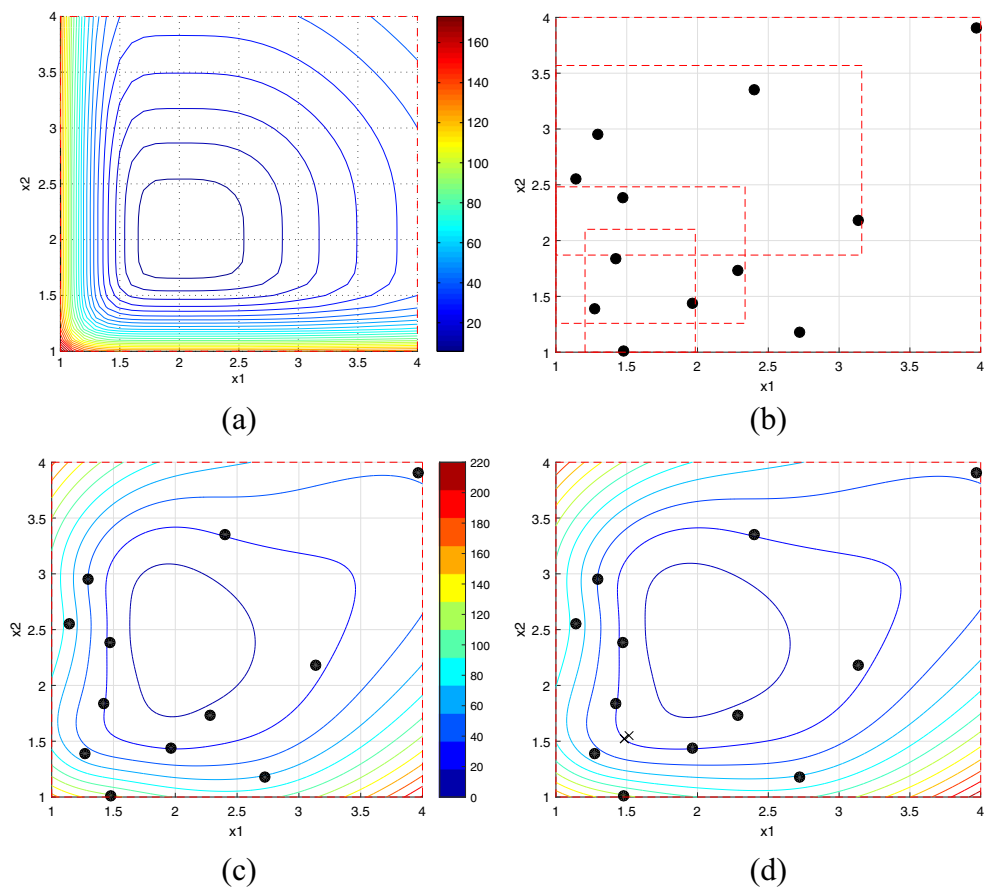


Fig. 9 Test function 4: Three-Hump Camel function (a) actual function; (b) RBF_{pri} ; (c) RBF_{pos}

Table 8 Overall accuracy performance of RBF_{pri} over RBF_{pos}

Test functions	RBF_{pos}					
	RND		LHS		HSS	
	$DNRMSE(\%)$	$DNMAE(\%)$	$DNRMSE(\%)$	$DNMAE(\%)$	$DNRMSE(\%)$	$DNMAE(\%)$
f_1	4,18	15,52	2,39	16,18	0,75	53,64
f_2	27,21	51,85	8,85	16,53	13,64	19,00
f_3	-1,50	-0,14	-0,42	4,90	-1,48	12,47
f_4	0,61	1,43	-3,18	12,65	4,05	9,93
f_5	12,59	25,35	8,71	19,42	14,55	20,97
Ave. Low	8,62	18,80	3,27	13,93	6,30	23,20
f_6	5,40	11,84	4,72	11,83	5,06	4,59
f_7	-0,85	-2,07	-0,84	-1,86	-0,37	-2,86
f_8	0,30	1,69	0,42	1,76	1,78	7,33
f_9	9,76	38,85	9,35	39,34	59,91	79,85
Ave. High	3,65	12,58	3,41	12,77	16,59	22,23
Ave. All	6,41	16,04	3,33	13,42	10,88	22,77

Fig. 10 Analytical “black-box” function (a) contour plot; (b) four successive iterations generating 12 sample points; (c) contour plot of the objective for the DoE with 12 sample points; (d) contour plot of the augmented DoE with 12+3 sample points. The three augmented points are marked with a cross



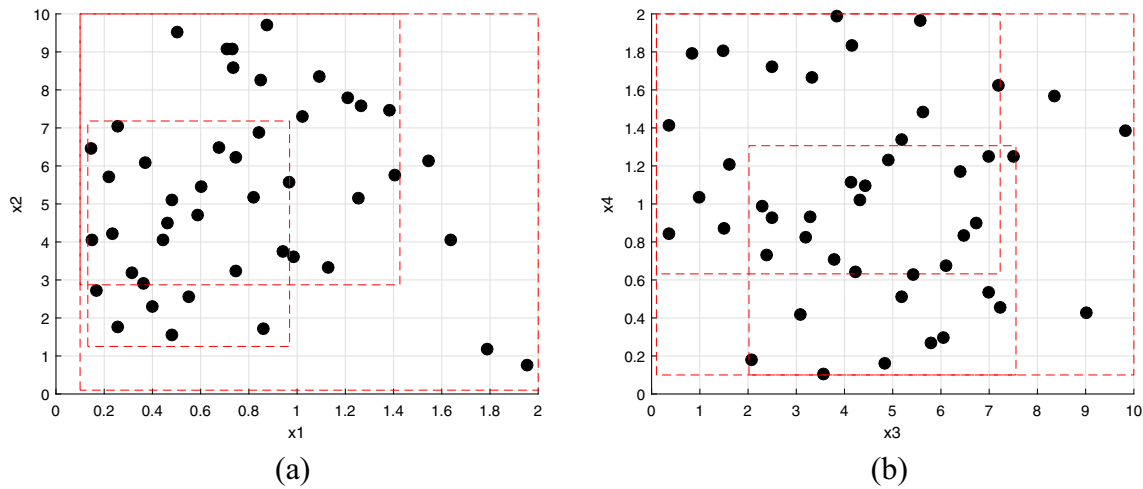


Fig. 11 Successive quadratic response surface screening generating 45 sampling points. The plots are showing the same DoE from two different views

The problem in (37) is now solved by performing a DoE procedure and setting up corresponding RBFs which in turn define a new optimization problem that is solved using a global search with a genetic algorithm and a local search with sequential linear and/or quadratic programming. First, a set of sampling points are generated by successive linear response surface optimization of the problem in (37) using four successive iterations with automatic panning and zooming (Gustafsson and Strömberg 2008). This screening generates 12 sampling points according to Fig. 10. Then, RBFs are fitted to this DoE and an optimal point is identified. The DoE is then augmented with this optimal point and the RBFs are set up again. This procedure is repeated three times generating in total a DoE with 12 sampling points from screening and three optimal points from RBFs. Finally, meta model based design optimization using our RBFs for this DoE of 12+3 sampling points is performed. The optimal solution generated with this procedure is (1.4962, 1.5049), which is very close to the analytical optimum of (37).

The DoEs presented in Fig. 3 are also studied for this example. The corresponding RBFs are set up and the optimal solutions for the random, LHS and HSS DoEs are obtained as (1.7842, 1.8003), (1.5618, 1.5076) and (1.5698, 1.553), respectively. It is clear that not only the choice of meta model will influence the result but also the choice of DoE. The solution from the random DoE is poor. The solutions from LHS and HSS DoEs are similar and acceptable. Thus, the successive screening procedure and optimal augmentation for generating DoE is for this problem superior and performs best. This is a general observation we have found for many examples. We have also used this strategy in order to solve reliability based design optimization problems using meta models. This is discussed in a most recent paper by Strömberg (2016), where also this first example is formulated as an reliability based

design optimization (RBDO) problem and is solved for variables with non-Gaussian distributions using a SORM-based RBDO approach. We also consider the following well-known engineering benchmark of a welded beam:

$$\begin{cases} \min_{x_i} & 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2) \\ \text{s.t.} & \begin{cases} \tau(x_i) - 13600 \leq 0 \\ \sigma(x_i) - 30000 \leq 0 \\ x_1 - x_4 \leq 0 \\ 0.125 - x_1 \leq 0 \\ \delta(x_i) - 0.25 \leq 0 \\ 6000 - P_c(x_i) \leq 0, \end{cases} \end{cases} \quad (38)$$

where definitions of shear stress $\tau(x_i)$, normal stress $\sigma(x_i)$, displacement $\delta(x_i)$ and critical force $P_c(x_i)$ can be found in e.g. the recent paper by Garg (2014), where also several solutions obtained by different algorithms are presented. In addition, the variables are bounded by $0.1 \leq x_1, x_4 \leq 2$ and $0.1 \leq x_3, x_4 \leq 10$. We obtain the following analytical solution (0.24437, 6.2175, 8.2915, 0.24437), which is more or less identical to the solution obtained by Garg: (0.24436, 6.2177, 8.2916, 0.24437).

Now, we solve this problem instead by generating a set of sampling points for which RBFs are fitted and then optimized. The procedure is similar to the one presented above. First, 45 sampling points are generated by quadratic response surface screening. This set of points are presented in Fig. 11. The choice of quadratic instead of linear screening depends on the non-linear constraint domain. Linear screening might result in an empty feasible domain. After screening, 15 additional points are added which are generated by optimum from sequentially augmented RBFs. Finally, for 45+15 sampling points, we set up the corresponding RBFs and we obtain the following optimal solution (0.414710, 3.925900, 6.620100, 0.414710). This

solution satisfies almost all constraints in (38). The first constraints is slightly violated $\tau(x_i) = 13729 > 13600$, but the five other constraints are fully satisfied. The value of the cost function is 3.113586, which is very close to the analytical optimum value of 2.381. This solution could of course be improved further by augmenting the DoE with additional optimal points.

8 Concluding remarks

In this paper, a new approach for setting up radial basis functions network is proposed by letting the bias be defined a priori by a corresponding regression model. Our new approach is compared with the established treatment of RBF, where the bias is obtained by using extra orthogonality constraints. It is numerically proven that our approach with a priori bias is in general as good as the performance of RBF with a posteriori bias. In addition, we mean that our approach is easier to set up and interpret. It is clear that the bias capture the global behavior and the radial basis functions tune the local response. It is also demonstrated that our RBF with a priori bias performs excellent in metamodel based design optimization and it captures both coarse and dense sampling densities simultaneously of DoEs generated from successive screening and optimal augmentation most accurately. In conclusion, the paper shows that our new RBF approach with a priori bias is a most attractive choice for surrogate model. We believe that our approach has a promising potential and opens up new possibilities for surrogate modelling in optimization, which we hope to be able to explore in a near future.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Amouzgar K, Strömberg N (2014) An approach towards generating surrogate models by using rbf with a priori bias. In: ASME 2014 International design engineering technical conferences and computers and information in engineering conference. American Society of Mechanical Engineers
- Amouzgar K, Rashid A, Strömberg N (2013) Multi-objective optimization of a disc brake system by using spea2 and rbf. In: Proceedings of the ASME 2013 international design engineering technical conferences, vol 3 B. American Society of Mechanical Engineers, Portland, DOI doi:[10.1115/DETC2013-12809](https://doi.org/10.1115/DETC2013-12809)
- Backlund PB, Shahan DW, Seepersad CC (2012) A comparative study of the scalability of alternative metamodeling techniques. *Eng Optim* 44(7):767–786
- Box GEP, Wilson KB (1951) On the experimental attainment of optimum conditions. *J R Stat Soc Series B (Methodological)* 13(1):1–45
- Branin FH (1972) Widely convergent method for finding multiple solutions of simultaneous nonlinear equations. *IBM J Res Develop* 16(5):504–522. doi:[10.1147/rd.165.0504](https://doi.org/10.1147/rd.165.0504). ISSN 0018-8646
- Fang H, Rais-Rohani M, Liu Z, Horstemeyer MF (2005) A comparative study of metamodeling methods for multiobjective crashworthiness optimization. *Comput Struct* 83(25–26):2121–2136. doi:[10.1016/j.compstruc.2005.02.025](https://doi.org/10.1016/j.compstruc.2005.02.025). ISSN 0045-7949
- Forrester AIJ, Keane AJ (2009) Recent advances in surrogate-based optimization. *Progress Aerospace Sci* 45(1–3):50–79. doi:[10.1016/j.paerosci.2008.11.001](https://doi.org/10.1016/j.paerosci.2008.11.001). ISSN 03760421. <http://linkinghub.elsevier.com/retrieve/pii/S0376042108000766>
- Garg H (2014) Solving structural engineering design optimization problems using an artificial bee colony algorithm. *J Ind Manag Optim* 10(3):777–794
- Goldstein AA, Price JF (1971) On descent from local minima. *Math Comput* 25(115):569–574. ISSN 00255718. <http://www.jstor.org/stable/2005219>
- Gustafsson E, Strömberg N (2008) Shape optimization of castings by using successive response surface methodology. *Struct Multidiscip Optim* 35(1):11–28
- Hardy RL (1971) Multiquadric equations of topography and other irregular surfaces. *J Geophys Res* 76(8):1905–1915
- Haykin S (1998) *Neural networks: a comprehensive foundation*, 2nd edn. Prentice Hall. ISBN 0132733501. <http://www.worldcat.org/isbn/0132733501>
- Jin R, Chen W, Simpson TW (2001) Comparative studies of metamodeling techniques under multiple modelling criteria. *Struct Multidiscip Optim* 23(1):1–13
- Kalagnanam JR, Diwekar UM (1997) An efficient sampling technique for off-line quality control. *Technometrics* 39(3):308–319. <http://www.jstor.org/stable/pdf/1271135.pdf?acceptTC=true>
- Kim B-S, Lee Y-B, Choi D-H (2009) Comparison study on the accuracy of metamodeling technique for non-convex functions. *J Mech Sci Technol* 23(4):1175–1181
- McKay MD, Beckman RJ, Conover WJ (1979) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 21(2):239–245
- Mullur A, Messac A (2006) Metamodeling using extended radial basis functions: a comparative approach. *Eng Comput* 21(3):203–217. doi:[10.1007/s00366-005-0005-7](https://doi.org/10.1007/s00366-005-0005-7)
- Rosenbrock HH (1960) An automatic method for finding the greatest or least value of a function. *Comput J* 3(3):175–184. doi:[10.1093/comjnl/3.3.175](https://doi.org/10.1093/comjnl/3.3.175). <http://comjnl.oxfordjournals.org/content/3/3/175.abstract>
- Sacks J, Schiller SB, Welch WJ (1989) Designs for computer experiments. *Technometrics* 31(1):41–47. doi:[10.2307/1270363](https://doi.org/10.2307/1270363). ISSN 00401706
- Simpson TW, Mauery TM, Korte JJ (1998) Multidisciplinary optimization branch, and Farrokh Mistree. Comparison of response surface and kriging models for multidisciplinary design optimization. In: AIAA paper 98-4758. 7 th AIAA/USAF/NASA/ISSMO Symposium on multidisciplinary analysis and optimization, pp 98–4755
- Simpson TW, Lin DKJ, Chen W (2001a) Sampling strategies for computer experiments: design and analysis. *Int J Reliab Appl* 2(3):209–240
- Simpson TW, Poplinski JD, Koch PN, Allen JK (2001b) Metamodels for computer-based engineering design: survey and recommendations. *Eng Comput* 17(2):129–150
- Simpson TW, Toropov V, Balabanov V, Viana FAC (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, pp 10–12
- Strömberg N (2016) Reliability based design optimization by using a slp approach and radial basis function networks. In: (to appear) ASME 2016 international design engineering technical conferences and computers and information in engineering conference. American Society of Mechanical Engineers
- Vapnik V, Golowich SE, Smola A (1996) Support vector method for function approximation, regression estimation, and signal processing. In: Advances in neural information processing systems, vol 9, pp 281–287. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.41.3139>
- Wang GG, Shan S (2007) Review of metamodeling techniques in support of engineering design optimization. *J Mech Des* 129(4):370–380
- Zhao D, Xue D (2010) A comparative study of metamodeling methods considering sample quality merits. *Struct Multidiscip Optim* 42(6):923–938. doi:10.1007/s00158-010-0529-3. ISSN 1615-147X. <http://www.springerlink.com/index/10.1007/s00158-010-0529-3>