Evaluation of Convergence Behavior of Metamodeling Techniques for Bridging Scales in Multi-scale Multimaterial Simulation

Oishik Sen[a], Sean Davis[b], Gustaaf Jacobs[b,*], H. S. Udaykumar[a,**]

[a] *Mechanical and Industrial Engineering, The University of Iowa, Iowa City, Iowa, 52242*
[b] *Aerospace Engineering, San Diego State University, San Diego, California 92115*

Abstract

The effectiveness of several metamodeling techniques, viz. the Polynomial Stochastic Collocation method, Adaptive Stochastic Collocation method, a Radial Basis Function Neural Network, a Kriging Method and a Dynamic Kriging Method is evaluated. This is done with the express purpose of using metamodels to bridge scales between micro and macro-scale models in a multiscale multimaterial simulation. The rate of convergence of the error when used to reconstruct hypersurfaces of known functions is studied. For sufficiently large number of training points, Stochastic Collocation methods generally converge faster than the other metamodeling techniques, while the DKG method converges faster when the number of input points is less than 100 in a two-dimensional parameter space. Because the input points correspond to computationally expensive micro/meso-scale computations, the DKG is favored for bridging scales in a multi-scale solver.

Keywords: Eulerian-Lagrangian, DNS, multi-scale, radial basis functions, artificial neural networks, convergence, stochastic collocation, particle drag model, Kriging, metamodeling

*Corresponding author
**Principal Corresponding author
Email addresses: oishik-sen@uiowa.edu (Oishik Sen), sean.davis@mail.mcgill.ca (Sean Davis), gjacobs@sdsu.edu (Gustaaf Jacobs), hs-kumar@uiowa.edu (H. S. Udaykumar)

1. Introduction

1.1. Motivation and Applications

A wide variety of problems in multi-material dynamics including the passage of a shockwave through a gas laden with particles [1], problems involving crack propagation in heterogeneous materials such as bones [2–5] or concrete structures [6, 7] involve the intricate coupling of physics at two or more distinct length and time scales. Further examples of such problems include modeling of heterogenous explosives [8–11], flow of mixtures including sediment transport in river beds [12], flow through fluidized beds [13] and flow of blood, i.e. plasma carrying cells and macromolecules [14]. In such systems, the physics of the micro/meso-scale needs to be represented in macro-scale simulations. This can be achieved by averaging over the heterogenous micro/meso-scale. In such volume-averaged macro models [15], or homogenized models [16–18], micro/meso physics appear in the form of closure terms in the macro-scale equations.

Process-scale computations typically demand macro-scale governing equations and simulation techniques. For example, in the problem of a shock wave interacting with a dusty gas, the number of dust particles is extremely large. To follow the evolution of the gas-solid mixture, a common practice is to define a computational particle as an agglomerate of a number of dust particles and to adopt a mixed Eulerian-Lagrangian viewpoint [17], as in Figure 1a. Particle paths are traced in a Lagrangian reference frame while solving the fluid equations in a fixed Eulerian frame. In this approach, the computational particles are modeled as singular point sources, which couple with the carrier fluid through momentum exchange modeled via source terms in the fluid equations [17, 19, 20]. The source terms close the unresolved momentum exchange between the fluid and solid (particle) phases, providing the forces on the particles. For small particle Reynolds numbers and incompressible flow, the drag on a spherical particle can be determined analytically using Stokes drag law [21]. A range of empirical drag laws exist, which incorporate the effect of inertia [22], compressibility [23, 24], slip coefficients [25], various shape factors [26] and/or viscosity ratio for droplets [27] for more complex flows. In general, closure models are obtained in the form of correlations developed in a physical experiment.

Empirical closure models such as drag correlations are only applicable in limited parameter spaces. To overcome this limitation, high resolution micro-scale methods that resolve the dynamics at the particle scale, as can be seen in Figure 1b [28], can be used as surrogates for physical experiments to obtain closure models connecting the meso-scale physics to the macro-scale.

In [29] for example, an artificial neural network (ANN) is used to construct a closure model for particle-laden shocked flow based on computational experiments. The neural network then supplies closure terms (drag force) to the macro-scale simulation. Further examples of closure terms constructed from computational experiments using an ANN can be seen in [2–7, 30].

1.2. Bridging Scales in a Multiscale Multimaterial Model

There are three components to the multiscale modeling approach described above: a macro-scale solver which computes the interaction of a large number of particles with a carrier flow, a meso-scale solver, which resolves the fine-scale particle-fluid dynamics of a smaller number of particles and a closure model which calculates the drag and other relevant parameters from the meso-scale solver for use in the macro-scale solver. Generation of a closure model derived from an ensemble of full-resolution meso-scale computations requires quantifying the output from the meso-scale dynamics (for example, drag forces) under a number of different input parameters such as shock strength, particle loading, particle size distribution, etc.

1.3. Metamodels as Surrogates to Bridge Scales

A metamodel, or a 'model of a model' [31], builds a hypersurface from a limited amount of input/output data and approximates the output over a much wider parameter space. An excellent overview of metamodeling techniques is given in [32–34]. Several studies have compared metamodels for reconstructing hypersurfaces from computational experiments. A review of the challenges and concerns in metamodeling techniques can be found in [35] and [36]. In addition, Jin et. al. [37] compared the hypersurfaces approximated by a Polynomial Response Surface Method (RSM), a Kriging method, a Radial Basis Function Neural Network (RBFANN), and Multivariate Adaptive Regression splines (MARS) for 14 different test functions. Fang et. al. [38] compared the RBFANN method and the RSM method, with the express purpose of reconstructing hypersurfaces in multi-objective crashworthiness optimization. However, these studies have been limited to comparing the quality of approximation only for a given number of input points, and not over a range of input points.

The choice of a "good" metamodeling technique depends on the application and the purpose of the metamodel. Because metamodels are constructed from expensive numerical computations in multi-scale modeling and because the multiscale method should converge with increasing degrees of freedom, convergence of the metamodels with respect to the number of

input points for a wide variety of hypersurfaces warrants careful investigation. This study shows that some metamodelling techniques converge faster than others only for a certain classes of hypersurface. Furthermore, some metamodels converge faster when the number of sampled input points is low, while other metamodels converge faster when the number of inputs is high. The focus of the current work is to examine the rate of convergence of the following three classes of metamodels for their suitability in bridging scales in a multiscale framework:

1. An interpolation method; Stochastic Collocation (PSC) methods [39–41] - the Polynomial Stochastic Collocation method (PSC) and the Adaptive Stochastic Collocation Method (ASC) are chosen as representative interpolation methods.
2. A fitting method; the RBFANN method [42–44] is the fitting method considered in the study.
3. A method which first fits a global response surface and then interpolates local departures from the global fit; the Kriging method [45–47] and the Dynamic Kriging Method (DKG) [48] are chosen in this study.

The methods, represent, in their respective classes as approximators, typical and state-of-the-art techniques for assimilating and representing the complex relationships between input parameters and the resultant outputs in a multi-dimensional parameter space.

The modeling methods are summarized in Section 2. In Section 3, the evaluation criteria that is used to assess the performance of the metamodels is discussed, and their convergence behavior is analyzed in Sections 4 and 5 for several analytic functions and existing empirical particle drag models. To elucidate their metamodeling capabilities, the approximation error of the metamodels on several known functions is analyzed.

These functions are chosen to highlight the strengths and weaknesses of each class of metamodels in approximating a wide variety of hypersurfaces; irregularly sampled data and presence of noise in the input data are also studied. Empirical drag laws are then considered to illustrate the complexities that will be encountered by the metamodeling techniques in building a closure model for a full multi-scale solver. Conclusions drawn from the quantification of the modeling error are discussed in Section 6. To the best of the authors' knowledge, no comparison of the rate of convergence of these classes of metamodels, with the express purpose of ascertaining the suitability of the methods as an ingredient in the framework of multiscale modeling, have been previously performed.

2. Inter-scale Coupling Methods

The metamodeling techniques are summarized in the following section. Broadly, the problem of metamodeling is the estimation of the value of a function $f(\mathbf{x})$ at a point $\mathbf{x}_0$ where $f(\mathbf{x}_0)$ is unknown and the value of $f(\mathbf{x}) : \mathbb{C}^n \to \mathbb{C}$ is only known at certain discrete (distinct) points, $\mathbf{x}_j$. Here, $\mathbb{C}^n$ [**0**,**1**] is a bounded subspace of $\mathbb{R}^n$, with **0** and **1** being an n-dimensional vector with all entries 0 and 1 respectively. The points, $\mathbf{x}_j, j = 1, 2, ..., N$, are the "input points" to the metamodel.

2.1. Stochastic Collocation Methods

Stochastic Collocation (SC) interpolation methods [39–41] rely on sparse grids generated using the Smolyak algorithm [49] to build a multivariate interpolation method by recursively taking the tensor products of univariate interpolation formulae. The resultant nodal architecture takes advantage of the recursive nature of the algorithm to enhance sparsity, i.e. limit the number of new data points required to improve the order of accuracy of the interpolation function. In SC methods, the estimated value of the function, $\widetilde{f}(\mathbf{x}_0)$, is given by [39–41],

$$\widetilde{f}(\mathbf{x}_0) = \sum_{j_1=1}^{m_{i_1}} \dots \sum_{j_n=1}^{m_{i_n}} f(x_{j_1}^{i_1}, \dots, x_{j_n}^{i_n})(a_{j_1}^{i_1} \otimes \dots \otimes a_{j_n}^{i_n}), \tag{1}$$

where i is the level of interpolation, m_i is the number of input points required for level i and a_j are the basis functions used in n dimensions. An overview of the SC methods used in this study follow. For a more detailed derivation the reader is referred to [39–41].

The convergence of the SC method depends on the choice of basis function as well as the nature of the hypersurface being interpolated. In this study, two SC methods are considered, which will be referred to as the Polynomial Stochastic Collocation (PSC) method and the Adaptive Stochastic Collocation (ASC) method. The input points for PSC method are based on the the end knots of a Chebyshev polynomial on a Clenshaw-Curtis grid (Figure 2a) [39–41]. The PSC method is particularly effective in interpolating globally smooth functions because of the fit based on Lagrange polynomials. However, for steeper gradients and highly localized features, the PSC method displays Gibbs phenomena. Adaptive methods are therefore required to avoid these spurious oscillations in the solution. The support nodes in a Clenshaw-Curtis grid are not suited for adaptivity because they must be predetermined at each level. In the ASC method, input/output

pairs are therefore located on a Newton-Cotes grid with equidistant nodes [41] (Figure 2b). The grid is locally refined around points where the hierarchical surplus, defined as,

$$\mathbf{w_j^i} = f(x_{j_1}^{i_1}, \ldots, x_{j_N}^{i_N}) - \tilde{f}^{i-1}(x_{j_1}^{i_1}, \ldots, x_{j_N}^{i_N}), \tag{2}$$

at level i on point j is above a threshold value, ϵ. Unless otherwise noted, $\epsilon = 0.001$ will be used here. Local linear spline functions are used as the polynomial bases in the ASC method because high order polynomial basis functions suffer from Gibbs's phenomenon on uniform grids [41].

A major advantage of SC methods is the availability of an *a priori* error estimate. For the PSC method, the interpolation error in the maximum norm [40, 41] is on the order of

$$||f(x_0) - \tilde{f}^i(x_0)||_\infty = \mathcal{O}\left(N^{-2}|log_2 N|^{3(n-1)}\right) \tag{3}$$

where n is the number of dimensions and N is the total number of interpolation points. In the ASC method, the additional error depends on the threshold hierarchical surplus value, ϵ [41],

$$||\tilde{f}^i(x_0)_{\mathrm{PSC}} - \tilde{f}^i(x_0)_{\mathrm{ASC}}||_\infty \leq N_2\epsilon \tag{4}$$

where N_2 is the difference between the full Smolyak sparse grid and the number of input points in the locally refined grid.

2.2. Artificial Neural Networks Using Radial Basis Functions

A Radial Basis Function Artificial Neural Network (RBFANN) is a fitting method, particularly suited to function approximation and pattern recognition [50], which comprises a (finite) set of identical basis functions, called Radial Basis Functions (RBF) centered around several distinct points in the input space. In an RBFANN method, the estimated value of the function at a point $\mathbf{x}_0$, is given by [42]

$$\widetilde{f}\left(\mathbf{x_0}\right) = \sum_{k=1}^{M} \lambda_k \phi\left(\mathbf{x_0}, \mathbf{x}_{c_k}; \theta_{c_k}\right) \tag{5}$$

where λ_k is the weight associated with each of the basis functions $\phi\left(\mathbf{x_0}, \mathbf{x}_{c_k}; \theta_{c_k}\right)$, while the parameter $\mathbf{x}_{c_k}$ is the position of the basis function and the parameter θ_{c_k} is the shape parameter. A typical choice of the basis function is a Gaussian $\phi\left(\mathbf{x_0}, \mathbf{x}_{c_k}; \theta_{c_k}\right) = exp\left(-d_{c_k}^2/\theta_{c_k}^2\right)$ where $d_{c_k} = \|\mathbf{x_0} - \mathbf{x}_{c_k}\|$, is the

Euclidean Norm; this basis function has non-compact support [44, 51–53]. If the Gaussians are equally spaced, they form a set of Reisz Bases for the input space [54].

The weights in (5) are given by

$$\boldsymbol{\lambda} = \mathbf{H}^{\dagger}\mathbf{f} \tag{6}$$

where $\mathbf{H} = H_{jk} = \phi\left(\left\|\mathbf{x}_j - \mathbf{x}_{c_k}\right\|; \theta_{c_k}\right); j = 1, 2, ..., N,\ k = 1, 2, ..., M$, with N being the number of inputs and M the number of Gaussians used. $\mathbf{H}^{\dagger}$ is the pseudoinverse of the interpolation matrix, $\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 & \lambda_2 & \cdots & \lambda_M \end{bmatrix}^T$ and $\mathbf{f} = \begin{bmatrix} f(\mathbf{x}_1) & f(\mathbf{x}_2) & \cdots & f(\mathbf{x}_N) \end{bmatrix}^T$. Equation (6) is thus the least-squares solution to the overdetermined system of equations given by $\mathbf{H}\boldsymbol{\lambda} = \mathbf{f}$. Because $\mathbf{H}$ is a dense matrix and often ill-conditioned [55, 56], a singular value decompostion of the interpolation matrix is performed to obtain $\mathbf{H}^{\dagger}$. The determination of optimal parameters, M, $\mathbf{x}_{c_k}$ and θ_{c_k} of an RBFANN is a subject of active reseach [42, 57–64]. In the current approach the parameters are determined using an unsupervised training process (which means that a non-linear optimization algorithm is not performed to determine the parameters). The RBF algorithm involves the following stages.

1. *Determination of M:* In order to avoid "memorizing" [65] the inputs, in a typical RBFANN, the number of RBFs is chosen to be less than the number of inputs. However, in the context of a multiscale framework, RBFANN is used to "learn" from fully-resolved micro-scale computational experiments. Since such computations are expensive, the method must create a hypersurface from as few inputs as possible. The number of Gaussians are therefore chosen to be approximately 1.1 times the number of inputs.
2. *Determination of* $\mathbf{x}_{c_k}, k = 1, 2, ..., M$. The RBFs are initially uniformly distributed in the domain and are updated by a K-means clustering algorithm [66] to avoid the possibility of an empty cluster in case of non-uniform inputs.
3. *Determination of the shape parameter,* θ_{c_k}*, of the RBFs:* The shape parameter is chosen to be equal to the mean distance of an RBF to its five nearest neighboring RBF such that they span the entire domain of the input space [66].

2.3. Kriging methods

The third class of metamodels studied are derived from Kriging methods, which have their origin in mining and geostatistical applications involving

spatially and temporally correlated data [67, 68]. Kriging methods combine a global (polynomial) model which fits to the given response surface, while the local departures from the global fit are estimated using semi-variogram models [45]. The resulting approximation interpolates the sampled input points. In a Kriging method, the estimated value of the function, denoted by $\tilde{f}(\mathbf{x}_0)$ can be expressed as [46],

$$\tilde{f}(\mathbf{x}_0) = \sum_{l=0}^{r} \lambda_l p_l(\mathbf{x}_0) + Z(\mathbf{x}_0) \tag{7}$$

where $p_l(\mathbf{x}_0)$ denotes the l-th order term in a polynomial basis functions of maximum order r and λ_l is the least-squares solution to the set of normal equations $\tilde{f}(\mathbf{x}_j) = \lambda_l p_l(\mathbf{x}_j)$, $j = 1, 2, ..., N$ [48]. In Equation(7), $Z(\mathbf{x}_0)$ is a realization of a Gaussian random process with zero mean, $E[Z(\mathbf{x}_j)] = 0$ and a covariance structure, $E[Z(\mathbf{x}_j) Z(\mathbf{x}_q)] = \sigma^2 R_{jq}$, $j, q = 1, 2, ..., N$ where $\sigma^2$is the process variance, and R_{jq} is the correlation model of the process [69] . The correlation model, R_{jq} is often chosen to be of the form [69]

$$\mathbf{R} = R_{jq} = R(\theta, \mathbf{x}_j, \mathbf{x}_q) = \prod_{k=1}^{n} \gamma_k(\theta_k, d_k), \tag{8}$$

with a shape parameter θ where $d_k = (x_{kj} - x_{kq}), k = 1, 2, ..., n$. An example of such a correlation model is a Gaussian model, where $\gamma_k = e^{-\theta_k d_k^2}$, $k = 1, 2, ..., n$. Although any value of θ would provide an interpolation formula, the optimal value of θ in (8) is selected to maximize the following log-likelihood function of the model parameters [48]

$$l = -\frac{N}{2} \ln\left[2\pi\sigma^2\right] - \frac{1}{2} \ln\left[|\mathbf{R}|\right] - \frac{1}{2\sigma^2} (\mathbf{f} - \mathbf{P}\boldsymbol{\lambda})^T \mathbf{R}^{-1} (\mathbf{f} - \mathbf{P}\boldsymbol{\lambda}) \tag{9}$$

where $\mathbf{f} = \begin{bmatrix} f(\mathbf{x}_1) & f(\mathbf{x}_2) & \cdots & f(\mathbf{x}_N) \end{bmatrix}^T$, $\boldsymbol{\lambda} = \begin{bmatrix} \lambda_0 & \lambda_1 & ... & \lambda_r \end{bmatrix}^T$and $\mathbf{P} = P_{jl} = p_l(\mathbf{x}_j)$. In the current work, the Kriging method with a first-order mean structure and a Gaussian correlation model is applied using a MATLAB code, DACE [47].

An improved version of Kriging algorithm called the Dynamic Kriging Method with adaptive sampling (DKG) [48, 69] has been implemented as well. The detailed algorithm can be found in [48], the key aspects of the algorithm are summarized here.

1. The DACE code uses a modified Hooke and Jeeves algorithm [47] to find the optimum value for θ. However, this method often fails to

provide a global optimum of θ and has therefore been replaced in the DKG method by a Global Pattern Search (GPS) algorithm [69].

2. The choice of the correlation model and the mean structure depends on the hypersurface to be approximated [70] and is not known a priori. In the DACE code, the order of the mean-structure, r, and the correlation function are selected by the user a priori. However, the DKG method compares between three mean structures corresponding to $r = 0, 1$ and 2, using a Cross-Validation (CV) error estimate [48]. The method also evaluates seven different correlation models, as listed in Table (1) and selects the best one, i.e. the model which maximizes the likelihood function.
3. In a typical Kriging model, samples are either supplied by the user heuristically or are supplied by a sampling strategy like the Latin Hypercube Sampling method [71, 72] or Latin Centroidal Voronoi Tessellation (LCVT) method [73].These methods usually generate input points in the domain uniformly. However, the Dynamic Kriging model is integrated with an adaptive sampling strategy, which selects more samples at highly non-linear portions of the hypersurface, thus aiming to obtain a better approximation using a parsimonious distribution of input points. The sample insertion criterion is described in detail in [69].

3. Evaluation of Techniques for Metamodeling

In order to be used for the generation of closure models in a multiscale modeling framework, it is desirable that the metamodels described in the previous sections satisfy certain restrictions on the error behavior and model construction. These include:

1. *Parsimonious representation*: Computational experiments are expensive to perform and a single high-resolution realization can take several hours to compute, even on multiprocessor architectures. Thus, the metamodel should be accurate and converge rapidly when supplied with information obtained from a minimum number of high resolution simulations (input data points).
2. *Monotonic convergence:* The inclusion of additional meso-scale simulations must result in improved approximation of the closure model. Because the closure model will not be known *a priori*, monotonic convergence is required so the modeling error can be estimated and additional micro-scale computations can be performed to improve the accuracy of the metamodel.

3. *Multidimensional representations*: Since multidimensional parameter spaces are expected, the method must be easily extendable to multiple input dimensions without suffering from the "curse of dimensionality", i.e. the number of input points should not increase exponentially with the increase in the dimension of the input space. This consideration obviates the use of methods like Lagrange interpolation and discourages the use of methods which operate on a dense grid of input points.
4. *Flexibility and Re-use*: Since the metamodelling technique relies on expensive high resolution simulations as inputs, previous results must be utilized when expanding the parameter space. In addition, if the parameter space is expanded to include a larger domain of approximation, the augmented parameter space and corresponding data values must be included in generating an updated hypersurface. This becomes difficult, for example, when a metamodelling approach relies on fixed collocation points (for example, Gauss points in the computational domain, zeroes of a Chebyshev polynomial, etc) for constructing closure models because additional interpolation would be needed to fit data onto the predefined nodes.
5. *Treatment of noisy data*: Since numerical noise is expected in the meso-scale results, it is preferred to have the metamodel filter noisy data to build a smooth approximation without adding excessive filtering errors.

4. Analysis of Convergence Behavior of the Metamodeling Techniques for Analytical Functions

To evaluate and compare the metamodels in light of the above mentioned criteria, hypersurfaces for several predetermined functions are generated and the approximations are compared to the exact functions. The method of comparing the metamodels is as follows:

1. *Training the metamodels:* A number (say N) of input points of a given function are provided to the metamodels as training points. These sample inputs are spaced at regular intervals along each axis throughout the parameter space for the RBFANN and the Kriging method. Because the input points of a PSC method are predetermined for each level of refinement, a comparable number ($\sim N$) of nodal collocation points are provided as inputs for the PSC method. In the DKG and ASC methods, the sample insertion criteria is adjusted such that when

the maximum number of samples reach N, no further inputs are generated and the training process is terminated. Thus, the approximation of the metamodels are evaluated at a comparable number of training points.

2. *Building the hypersurface to test accuracy:* Once trained, the metamodel is used to predict the value of the function at $S = 100 \times 100$ uniformly distributed points in the domain of approximation.
3. *Evaluating the Approximation Error and the Rate of Convergence:* Because metamodels are constructed from analytical functions, it is possible to compare the predicted values of the metamodels with the exact values of the function at these S points. In order to quantify the accuracy of approximation at these points, a normalized sum-of-squares error is calculated:

$$\epsilon_{L^2} = \sqrt{\frac{\sum_{j=1}^{S} \left(f\left(\mathbf{x}_j\right) - \widetilde{f}\left(\mathbf{x}_j\right) \right)^2}{\sum_{j=1}^{S} \left(f\left(\mathbf{x}_j\right) \right)^2}}, \tag{10}$$

where $f\left(\mathbf{x}_j\right)$ is the exact value of the function at the point $\mathbf{x}_j$ and $\widetilde{f}\left(\mathbf{x}_j\right)$ is the value approximated by a metamodel. The error, ϵ_{L^2}, is calculated and plotted for different values of N.

4. *Local Error Estimation:* Since the error measure ϵ_{L^2} is a global measure of approximation errors, a normalized local error field, $\delta\left(\mathbf{x}_j\right)$, defined as

$$\delta\left(\mathbf{x}_j\right) = \frac{f\left(\mathbf{x}_j\right) - \widetilde{f}\left(\mathbf{x}_j\right)}{\frac{1}{S}\sum_{j=1}^{S} f\left(\mathbf{x}_j\right)}, \tag{11}$$

is calculated to quantify the local approximation error of a hypersurface by a metamodel at the points $\mathbf{x}_j$, $j = 1, 2, ..., S$. Representative plots showing the contours of the local error field for a given value of N are also shown in the subsequent section.

For the purpose of illustration, representative contours of the hypersurfaces and local error plots are shown for $N = 144$ points.

4.1. Harmonic Test Function

The first function considered is a smooth harmonic function, given by

$$f(x, y) = \sin(2\pi x)\cos(4\pi y) + 2, \tag{12}$$

where x and y range from 0 to 1.

As the convergence plot in Figure 3 shows, if the number of input points is below 60, the hypersurface is best approximated by the RBFANN and DKG methods. However, as the number of input points increases, the rate of convergence of the RBFANN method decreases. The value of the shape parameter, $\theta^{\mathbf{c}_j}$, decreases as the number of input points increases for the RBFANN method. The system of equations defined by (6) becomes ill-conditioned and the SVD solver essentially "filters" out higher frequencies of the interpolation matrix. As the higher frequencies are removed, the accuracy of the representation decreases and the rate of convergence decreases if the number of inputs approaches and exceeds 100 points.

The error in approximation by the Kriging method, applied using the DACE code, does not decrease monotonically, evinced by the spikes seen in Fig. 4. The rate of convergence changes when the number of input points changes from 64 to 81 and from 81 to 100. To further investigate this, the value of the shape parameter θ, as estimated in the Kriging Method is examined and compared with that obtained by the DKG method in Table 2. The value of θ for the Kriging method undergoes sharp changes when the number of input points change from 64 to 81 and from 81 to 100. This corresponds to those points in Figure 3 when the rate of convergence of the Kriging method also changes. In comparison, the shape parameter estimated by the DKG method decreases uniformly (Table 2) and the error in the DKG method also decreases uniformly (Figure 3). This numerical example illustrates the advantage of the GPS algorithm for determining the optimum value of θ over the modified Hooke and Jeeves algorithm used in DACE.

The error in approximation of the PSC method is initially constant, as shown in Fig. 3, and decreases as the number of input points exceeds 30, finally decreasing spectrally when the number of inputs increases beyond 60. In contrast, the error of approximation of the ASC method does not change for any given number of input points. The sinusoidal variation of the function along the x axis is satisfactorily reconstructed, but the cosine waves in the y direction are not recognized all together, as is shown in the approximation of the hypersurface by the ASC method in Fig. 4. The ASC method initially operates on a mesh where the collocation nodes lie along the boundaries and the centerlines of the input domain (as seen in Figure 2). Because of the adequate representation of the sinusoidal values at the boundary and along the centerlines, the hierarchical surplus falls below the threshold value and input nodes are not successively refined on the interior of the domain. Hence, the hypersurface reconstructed by the ASC method is constant with respect to the coordinates along the y axis causing an

aliasing error. The sinusoidal variation of the function along the x axis is satisfactorily reconstructed, but the cosine waves in the y direction is not predicted by the ASC method.

To appropriately characterize the convergence of the ASC on a smooth function, the aliasing error is eliminated by introducing slight phase-shifts in the sine and cosine waves of the function,

$$f(x, y) = \sin(2\pi x + 0.25)\cos(4\pi y + 0.5) + 2, \tag{13}$$

where x and y range from 0 to 1. The aliasing error seen in Fig. 4 does not appear in this function because the boundaries of the domain no longer trace a simple sine wave. The Newton-Cotes grid in Figure 2 does not line up directly with the sine wave so grid refinement occurs on the interior of the domain away from the centerlines and the sum-of-squares error converges per the theoretical expectation when the number of input points is greater than 20, which is shown in Figure 5.

4.2. Hypersurface Reconstructed from Irregularly Spaced Input Points

In a computational experiment, it is possible to specify the input points in several different ways. Input points can be specified at the nodes of a structured grid, i.e. at strictly regular intervals along each axis. In sampling methods like Latin Hypercube Sampling Method [71], LCVT method [73] input points are distributed uniformly along the domain, although the points may not be necessarily specified at strictly regular intervals along each axis. The resulting input grid, in such a case, is unstructured. Because in a multiscale model it is not always possible or advisable to perform meso-scale experiments at strictly regular intervals in the parameter space, the input grid of a metamodel may not necessarily be structured. Therefore, a metamodeling approach which is fairly insensitive to the distribution of input points is preferable. In this section, a comparison is made between the hypersurfaces created by regularly spaced input points and irregularly spaced input points using the RBFANN and Kriging methods. Note that the PSC method is trained from inputs placed at specific locations in the input domain (see Fig 2), while the ASC method and the DKG method are integrated with a sampling strategy and are therefore not tested in this section.

To train the RBFANN methods and the Kriging methods, the harmonic function defined by Equation (12) is considered. Here, the input points are chosen at random within the domain. Because the location of a given number of input points in the domain is not unique, five such random input

distributions are used to calculate the average normalized sum-of-squares error, given by Equation (10). This is then compared to the error calculated from the approximation of a structured grid of training points. Figures 6a and 6b show the rate of convergence of the error for the RBFANN and Kriging methods respectively.

The convergence rates shown in Figures 6a and 6b demonstrate that structured grids give the lowest error for the given harmonic function. However, the convergence of the Kriging Method trained with an unstructured grid of sample points closely follows the convergence when using a structured grid of input points. However, if the number of input points increases beyond 81, the rate of convergence of the RBFANN method trained with regular input points is significantly different from that trained with random input points. Because the current architecture of the RBFANN uses more Gaussians than the number of input points, the RBFANN is sensitive to the distribution of input points. Thus, the rate of convergence of the current RBFANN model not only depends on the number of training points, but also on the location of the input points in the domain. Unless otherwise mentioned, an RBFANN method will be trained with regularly spaced inputs in subsequent sections of the present work.

4.3. Reconstruction of a Hypersurface from Noisy Data

Noise is unavoidable in solutions obtained from computational and physical experiments. While it is possible to filter out noise from any approximation obtained from a metamodel with a pre/post processing algorithm, additional errors may be added if a filter is used. Therefore, a metamodel that is relatively insensitive to noise is preferred.

To analyze the metamodels' response to numerical noise, each of the metamodels is given a fixed number of inputs from Equation (13) and a (white) noise of maximum amplitude 0.1 is superposed onto the training samples. The metamodels are then used to predict the hypersurface of Equation (13). The contour plots for each of the reconstructed hypersurfaces are shown in Figure 7.

The hypersurface predicted using the RBFANN method is noise-free. Because an RBFANN filters out all frequencies beyond a certain limit [54], it can filter out the noise components most effectively. As seen in Figure 7, the hypersurfaces predicted using the SC methods are noisy because the PSC and ASC are interpolation methods in which the reconstructed hypersurface must pass through all the values given at the training points exactly. Also, the Kriging and DKG methods have an inherent mean-structure which filters out the noise partially, but the process of minimization of the departure from

the local fit result in an interpolation. Therefore, as can be seen in Figure 7, the noise from the training data is retained in the hypersurface predicted by these methods. Unless otherwise mentioned, the training data in the subsequent sections of the work is noise-free.

4.4. A Radially Symmetric Steep Gradient Test Function

The previous examples provided valuable insight into the convergence of the metamodels for a smooth function. To study the response of the metamodels to a hypersurface with steep gradients localized in a region in the interior of the input domain, consider the function,

$$f(x,y) = \begin{cases} \frac{C_1}{2\sqrt{0.4}} \ln\left|\frac{\sqrt{x^2+y^2}+\sqrt{0.4}}{\sqrt{x^2+y^2}-\sqrt{0.4}}\right| & \text{if } \sqrt{x^2+y^2} \leq \sqrt{0.3}, \\ \frac{C_2}{2\sqrt{0.2}} \ln\left|\frac{\sqrt{x^2+y^2}-\sqrt{0.2}}{\sqrt{x^2+y^2}+\sqrt{0.2}}\right| & \text{if } \sqrt{x^2+y^2} > \sqrt{0.3}, \end{cases} \tag{14}$$

where

$$C_1 = \frac{16\sqrt{0.4}}{\ln\left|\frac{\sqrt{0.3}+\sqrt{0.4}}{\sqrt{0.3}-\sqrt{0.4}}\right|}$$

$$C_2 = \frac{16\sqrt{0.2}}{\ln\left|\frac{\sqrt{0.3}-\sqrt{0.2}}{\sqrt{0.3}+\sqrt{0.2}}\right|}$$

and x and y range from 0 to 1. This function is an integral of Equation (65) from [41]. The contours of $f(x,y) =$ constant are radially symmetric with respect to the origin. Steep gradients arise along the arc $\sqrt{x^2+y^2} = \sqrt{0.3}$, while the hypersurface varies more slowly as the distance from the arc increases. The exact hypersurface is shown in Fig. 9a.

Figure 8 shows the convergence of the errors of the metamodels with respect to the number of input points. In the convergence plot (Figure 8), the convergence rates for the metamodels are similar if the number of input points is below 900. However, the magnitude of the error is higher for the SC methods because the nodes are concentrated along the boundaries and centerlines of the domain, while the regions of high gradient are radially symmetric. The input points are therefore not collocated with the highly non-linear regions of the hypersurface and the local features of the hypersurface are therefore not well resolved. The maximum values of the function are underrepresented when using the SC methods and oscillations occur. These oscillations increase as the distance between a point and the

arc $\sqrt{x^2+y^2} = \sqrt{0.3}$ decreases resulting in the corresponding higher local errors seen in Figures 10a and 10b.

Figure 9 shows that the hypersurfaces reconstructed by the RBFANN, Kriging and the DKG methods also display spurious oscillations, but these oscillations are smaller compared to those seen in the PSC reconstruction. When the number of input points is greater than 900, the PSC method converges exponentially because a larger number of input nodes fall on the arc containing the steeper gradients. If the number of input points increases beyond the scope of this study ($> 10^3$), the ASC method converges exponentially because of the increased number of nodes near the steep gradients. An example of the node distribution for such a case (i.e. $N > 10^3$) is shown in Figure 11.

5. Analysis of Convergence Behavior of the Metamodeling Techniques for Empirical Drag Models

In addition to specific analytical functions designed to quantify the metamodels' approximation error on smooth harmonic functions and steep interior gradients, empirical drag models are considered. The hypersurfaces of these functions are expected to be similar to those of the closure models that the metamodeling techniques would be required to approximate in a multiscale modeling framework.

5.1. Boiko's Model for Drag on a Particle in a Shocked Flow

The first drag model considered is a model proposed by Boiko et. al. [74], and is given by

$$C_D = \left(0.38 + \frac{24}{Re_p} + \frac{4}{Re_p{}^1\!/_2}\right)\left(1 + e^{-\frac{0.43}{M_p^{4.67}}}\right), \tag{15}$$

This function includes the effects of particle Reynolds number, $Re_p = |v_f - v_p| d_p/\nu$, and relative Mach number, $M_p = |v_f - v_p|/\sqrt{T_f}$, in the drag coefficient equation. The model is limited to relative Mach numbers of $M_p \leq 1.2$ and Reynolds numbers of $Re_p \leq 1 \times 10^4$. However, for the present study, the model is considered to apply for $0.1 \leq M_p \leq 3$ and $100 \leq Re_p \leq 10000$. The macro-scale EL code in [17, 19, 20] uses this empirical function to compute the particle drag coefficient.

The drag predicted by this model has two sharp zones of transition: at transonic Mach numbers ranging from $0.5 \lesssim M_p \lesssim 1.5$ and at $Re_p \lesssim 200$

when the flow changes from viscous Stokes' flow to an inertia dominated flow. The contour of the hypersurface of this figure is shown in Figure 14a.

The rate of convergence of the error of the metamodels with respect to the number of input points is shown in Figure 12. The function is most accurately represented by the SC methods. This is because in the SC methods collocation points are more concentrated close to the boundary (Figure 2). The sharpest transition region in the hypersurface in 14a lies along the low Re_p boundary. Because the concentration of nodes in the SC methods coincide with the highly non-linear regions of the hypersurface, the SC methods most accurately represent the function. It can be further observed from Figure 12 that among the SC methods, when the number of input points is below 400, the PSC method best approximates the hypersurface. This is because the high order global basis functions in the PSC method capture the flatter portions and smooth transition regions more accurately than the lower order local basis functions used by the ASC. If the number of input points is increased beyond 400, the ASC method is more accurate (Figure 12). This is because of the adaptive refinement based sampling strategy in the ASC method, which results in higher number of input points in the two transition zones and the ASC method converges exponentially.

As shown in Figure 12, the magnitude of the error of approximation by the RBFANN, Kriging and DKG methods is higher than the SC methods. In the RBFANN and the Kriging method, the predicted hypersurface shows spurious oscillations (from Figure 15). The magnitude of local error is highest for the Kriging method and the RBFANN method, while the magnitude of local error is higher for the DKG method than the SC methods. An important observation in Figure 15 that unlike the SC methods, the hypersurface obtained from the other metamodels is most errorneous along the transition region from low Re to high Re. This implies that unlike the SC methods, the number of input points for the other metamodeling techniques in these regions is not adequate to represent the sharp transition from low Re to high Re. Figure 13 shows the training points of the DKG method in. The number of training points is uniformly distributed throughout the domain, unlike the SC methods. Similarly, the RBFANN methods and the DKG methods are also trained using regularly spaced training points and hence the error of approximation of the RBFANN method, the Kriging method and the DKG method is higher than the SC methods, (Figure 12 and Figure 15).

5.2. Loth's Model for Drag on a Particle in a Shocked Flow

To further investigate the ability to capture steep gradients in the interior of the domain, consider the drag model proposed by Loth et. al. [24]. This

model also corrects for high particle Mach and Reynolds numbers but over a wider range, $Re_p \leq 1 \times 10^5$ and $M_p \leq 5$,

$$C_D = \frac{24}{Re_p}\left(1 + 0.25Re_p^{0.687}\right)\left(1.0 - \frac{0.258C}{1.0 + 514G}\right) + \frac{0.42C}{1 + 4.25 \times 10^4 Re_p^{-1.16}G}, \tag{16}$$

where C and G are defined as,

$$C = \begin{cases} \frac{5}{3}\tanh(3.0\log(M_p + 0.1)) & \text{if } M_p \leq 1.45, \\ 2.044 + 0.2e^{-1.8\left(\log\left(\frac{M_p}{2}\right)\right)^2} & \text{if } M_p > 1.45, \end{cases} \tag{17}$$

$$G = \begin{cases} 1.0 - 1.525M_p^4 & \text{if } M_p \leq 0.89, \\ 0.0002 + 0.0008\tanh\left(12.77\left(M_p - 2.02\right)\right) & \text{if } M_p > 0.89, \end{cases} \tag{18}$$

The function produces very steep gradients arranged in a series of steps in the interior of the domain. The contour of the hypersurface of Equation (16) is shown in Figure 17a.

As shown in Figure 16, most of the metamodels show a first order convergence rate in the normalized sum-of-squares error, but the ASC method converges exponentially. Similar to the drag model discussed in the previous section, because the adaptive refinement algorithm in the ASC adds additional training nodes at highly non-linear regions in the hypersurface, the ASC method converges exponentially.

As opposed to the function given by Equation (15), many of the complex characteristics of Loth's model lie in the interior of the domain as can be seen in 17a. But from Figure 2, it can be seen that the number of nodes of the PSC method are scarce in the interior of the domain. This lack of training data along with the requirement that the approximation be exact on the training nodes causes spurious oscillations in regions of high gradient (Figure 17). Because the Kriging and DKG methods also interpolate the hypersurface through the available inputs, spurious oscillations can also be seen in the hypersurface predicted by the Kriging method and the DKG method, and is shown in Figure 17. The oscillations in the DKG method is more localized than the Kriging method. To investigate this, the correletion model used in the DKG method is studied, and it is found that for any given number of input points, the DKG method approximated the hypersurface of Equation (16) using a General Exponential correletion model. The correlation model used in the Kriging method is a Gaussian model, but

a General Exponential correlation model is used in the DKG method. A General Exponential model is more localized than a Gaussian and the use of the General Exponential correlation model approximates the localized features of the highly non-linear portions of the hypersurface. Because the RBFANN method also uses non-compact Gaussian basis functions, spurious oscillations can also be seen in the hypersurface approximated by the RBFANN in Figure 17. Furthermore, the ASC model also uses local basis functions and adaptively places additional nodes near the higher gradient regions of the input domain thus eliminating these oscillations in Figure 17.

5.3. Tong's Model for Drag on a Particle in a Shocked Flow

Tong et. al. [26] have extended Loth's model [24] to include variations of particle shape and particle volume fraction, α,

$$C_D = C_{da} + 0.5048\alpha \left(1.0 + \frac{34.8}{Re_p^{0.5707}}\right)^4 + 0.9858\alpha \left(1.0 + \frac{34.8}{Re_p^{0.5707}}\right), \quad (19)$$

where C_{da} is the drag coefficient calculated using (16). This model illustrates the complex dependence of the particle drag coefficient on many different parameters, including the Knudsen number (set equal to 10 here), M_p (set to 1 in here), Re_p (varied) and α (varied). The hypersurface is shown in Figure 20a.

The convergence of the metamodels with respect to the number of input points is shown in Figure 19. The PSC method and the DKG method approximate the hypersurface most accurately. Despite the higher concentration of input nodes in the low Re_p, high α boundary, the local basis functions in the ASC method do not approximate the function as well. The RBFANN method, employing Gaussians as basis-functions, over-fits the nearly linear variation of the drag coefficient at higher volume fraction and lower Reynolds number. Therefore, as is shown in the contour plots of the reconstructed hypersurfaces in Figure 20, oscillations arise when the RBFANN method is used to build the approximation. Similar to the case of the harmonic function in Section 4, the Kriging model constructed using the DACE code does not converge monotonically. The value of the shape parameter as calculated using the Kriging method and the DKG method are compared against the number of input points in Table 3. The value of the shape parameter, θ, in the Kriging method is equal to 0.76655 for less than or equal to 400 input points but increases sharply to $\theta = 20$ past 400, where a non-monotonic jump is seen in the convergence plot in Figure 19. For the DKG method, the value of θ monotonically increases, indicating that the correlating model

becomes more localized with an increased number of input points. This numerical example also illustrates the advantage of the GPS algorithm [48] for determining the optimum value of θ over the modified Hooke and Jeeves algorithm use in DACE.

6. Conclusions

The performance of five metamodeling techniques, the PSC, ASC, RBFANN, Kriging and DKG methods, is compared for use as the coupling algorithm or a metamodel in a multi-scale solver. The magnitude and the rates of the representation error of each of these methods has been characterized by their sum-of-squares error from Equation (10) and the local errors (11).

For a large number of training points, the SC methods generally approximate most of the hypersurfaces most acurately. In particular, the adaptive refinement of the ASC method around steep gradients on the interior of the input domain captures the complex regions of high gradient in the hypersurfaces of the empirical drag functions tested. But the number of input points required to accurately predict a hypersurface using the SC methods is roughly equal to or more than 100 for most of the hypersurfaces. Because in a multiscale modeling framework, input points correspond to high resolution mesoscale computations, generation of such a high number of input points is expensive. Additionally, both the PSC and ASC methods are constructed using a strict predetermined nodal architecture and lack the flexibility of the Kriging and the RBFANN methods with respect to placement of input data. For example, with the SC-based methods, expanding the parameter space would entail discarding the input from a previous set of data or introducing additional interpolation errors. This would result in waste of computational time and resources when an expanded parameter space is required.

The input points of the RBFANN and the Kriging methods can be randomly placed throughout the domain with little or no effect on the convergence of the metamodel, as seen in Section 4.3. Because of this flexibility, the parameter space can be expanded to include a larger domain of approximation while continuing to utilize previous data. However, the RBFANN and Kriging methods have the highest sum-of-squares error in approximating most of the functions tested and do not converge at as high of rates as the SC methods. Additionally, the Kriging method using the DACE code does not converge monotonically in some cases. The parameter estimation technique integrated within the DACE code (i.e. the use of modified Hooke and Jeeves algorithm) leads to the selection of a local extremum value of

the shape parameter θ as the global extremum in the maximum likelihood estimation process.

The non-monotonic convergence of the Kriging method is circumvented in the DKG method by a Global Pattern Search (GPS) algorithm using a maximum likelihood estimator with a penalty function and by the use of dynamic selection of correlation models and mean structure. The DKG method is not only monotonically convergent for all the functions considered in the current work, but at roughly 100 input nodes, has either the lowest sum-of-squares error or is close to the lowest (i.e. relative to the SC methods). Therefore, metamodels may be built using less than 100 training points using the DKG method. Thus, for the functions approximated in the current work, the DKG method is the optimal choice to serve as the coupling algorithm for the multi-scale solver.

Acknowledgements

We gratefully acknowledge the financial support by the Air Force Office of Scientific Research under grant number FA9550-12-1-0115 and the National Science Foundation under grant number DMS-115631. The authors ackowledge Dr. K.K. Choi at the University of Iowa, Hyeongjin Song and Nicholas Gaul for their active support and enthusiasm, and for providing the in-house DKG code for performing the computations.

References

[1] A. V. Fedorov, Y. V. Kharlamova, T. A. Khmel, Reflection of a shock wave in a dusty cloud, Combustion, Explosion, and Shock Waves 43 (1) (2007) 104–113. doi:10.1007/s10573-007-0015-4.

[2] R. Hambli, Numerical procedure for multiscale bone adaptation prediction based on neural networks and finite element simulation, Finite Elements in Analysis and Design 47 (7) (2011) 835–842. doi:10.1016/j.finel.2011.02.014.

[3] R. Hambli, Multiscale prediction of crack density and crack length accumulation in trabecular bone based on neural networks and finite element simulation, International Journal of Numerical Methods in Biomedical Engineering 27 (2011) 461–475. doi:10.1002/cnm.

[4] R. Hambli, Apparent damage accumulation in cancellous bone using neural networks., Journal of the Mechanical Behavior of Biomedical Materials 4 (2011) 868–78. doi:10.1016/j.jmbbm.2011.03.002.

[5] R. Hambli, H. Katerchi, C.-L. Benhamou, Multiscale methodology for bone remodelling simulation using coupled finite element and neural network computation., Biomechanics and modeling in mechanobiology 10 (2011) 133–145. doi:10.1007/s10237-010-0222-x.

[6] J. F. Unger, C. Könke, Coupling of scales in a multiscale simulation using neural networks, Computers & Structures 86 (21-22) (2008) 1994–2003. doi:10.1016/j.compstruc.2008.05.004.

[7] J. F. Unger, S. Eckardt, Multiscale Modeling of Concrete, Archives of Computational Methods in Engineering 18 (3) (2011) 341–393. doi:10.1007/s11831-011-9063-8.

[8] J. B. Bdzil, R. Menikoff, S. F. Son, A. K. Kapila, D. S. Stewart, Two-phase modeling of deflagration-to-detonation transition in granular materials: A critical examination of modeling issues, Physics of Fluids 11 (2) (1999) 378. doi:10.1063/1.869887.

[9] A. K. Kapila, R. Menikoff, J. B. Bdzil, S. F. Son, D. S. Stewart, Two-phase modeling of deflagration-to-detonation transition in granular materials: Reduced equations, Physics of Fluids 13 (10) (2001) 3002. doi:10.1063/1.1398042.

[10] R. Menikoff, Hot spot formation from shock reflections, Shock Waves 21 (2) (2011) 141–148. doi:10.1007/s00193-011-0303-5.

[11] A. Kapahi, H. S. Udaykumar, Dynamics of void collapse in shocked energetic materials: physics of voidvoid interactions, Shock Waves 23 (6) (2013) 537–558. doi:10.1007/s00193-013-0439-6.

[12] Y. Abdelhamid, U. El Shamy, Multiscale modeling of flood-induced scour in a particle bed, Bridges 10 (2014) 740–749.

[13] M. van der Hoef, M. van Sint Annaland, N. Deen, J. Kuipers, Numerical simulation of dense gas-solid fluidized beds: a multiscale modeling strategy, Annual Review of Fluid Mechanics 40 (1) (2008) 47–70. doi:10.1146/annurev.fluid.40.111406.102130.

[14] W. Pan, D. A. Fedosov, B. Caswell, G. E. Karniadakis, Predicting dynamics and rheology of blood flow: a comparitive study of multiscale and low-dimensional models of red blood cells, Microvascular Research (82).

[15] C. T. Sun, R. S. Vaidya, Prediction of composite properties from a representative volume element, Composites Science and Technology 56 (1996) 171–179.

[16] W. E, B. Engquist, Z. Huang, Heterogeneous multiscale method: A general methodology for multiscale modeling, Physical Review B 67 (9) (2003) 1–4. doi:10.1103/PhysRevB.67.092101.

[17] G. B. Jacobs, W.-S. Don, A high-order WENO-Z finite difference based particle-source-in-cell method for computation of particle-laden flows with shocks, Journal of Computational Physics 228 (5) (2009) 1365–1379.

[18] B. Shotorban, G. B. Jacobs, O. Ortiz, Q. Truong, An eulerian model for particles nonisothermally carried by a compressible fluid, International Journal of Heat and Mass Transfer 65 (0) (2013) 845 – 854.

[19] S. Davis, T. Dittman, G. B. Jacobs, W. S. Don, High-Fidelity Eulerian-Lagrangian Methods for Simulation of Three Dimensional, Unsteady, High-Speed, Two-Phase Flows in High-Speed Combustors, in: 47th AIAA/ASME/SAE/ASEE Joint Propulsion Conference, 2011.

[20] S. Davis, T. Dittmann, G. Jacobs, W. Don, Dispersion of a cloud of particles by a moving shock: Effects of the shape, angle of rotation, and aspect ratio, Journal of Applied Mechanics and Technical Physics 54 (6) (2013) 900–912.

[21] G. G. Stokes, On the effect of the internal friction of fluids on the motion of pendulums., Transactions of the Cambridge Philosophical Society 9.

[22] R. Clift, J. R. Grace, M. E. Weber, Bubbles, Drops, and Particles, Academic Press, 1978.

[23] V. M. Boiko, S. V. Poplavskii, Drag of nonspherical particles in a flow behind a shock wave, Combustion, Explosion, and Shock Waves 41 (1) (2005) 71–77. doi:10.1007/s10573-005-0008-0.

[24] E. Loth, Compressibility and Rarefaction Effects on Drag of a Spherical Particle, AIAA Journal 46 (9) (2008) 2219–2228. doi:10.2514/1.28943.

[25] G. Tedeschi, H. Gouin, M. Elena, Motion of tracer particles in supersonic flows, Experiments in Fluids 26 (4) (1999) 288–296.

[26] X. Tong, M. Remotigue, E. Luke, J. Kang, Multiphase Simulations of Blast-Soil Interactions, in: Proceedings of the ASME 2013 Fluids Engineering Division Summer Meeting, no. FEDSM2013-16549, ASME, 2013.

[27] Z.-G. Feng, E. E. Michaelides, Drag Coefficients of Viscous Spheres at Intermediate and High Reynolds Numbers, Journal of Fluids Engineering 123 (4) (2001) 841–849.

[28] C. Lu, Artificial neural network for behavior learning from meso-scale simulations, applications to multi-scale multimaterial flows, Ph.D. thesis, The University of Iowa (2010).

[29] C. Lu, S. Sambasivan, A. Kapahi, H. S. Udaykumar, Multi-scale modeling of shock interaction with a cloud of particles using an artificial neural network for model representation, Procedia IUTAM 3 (2012) 25–52.

[30] S. Davis, O. Sen, G. Jacobs, H. Udaykumar, Coupling of micro-scale and macro-scale eulerian-lagrangian models for the computation of shocked particle-laden flows, in: ASME 2013 International Mechanical Engineering Congress and Exposition, Vol. 7A, ASME, 2013.

[31] J. P. Kleijnen, Statistical tools for simulation practitioners, Marcel Dekker, Inc., 1986.

[32] T. W. Simpson, J. D. Peplinski, P. N. Koch, J. K. Allen, Metamodels for computer-based engineering design : survey and recommendations, Engineering with Computers 17 (2001) 129–150.

[33] R. Jin, X. Du, W. Chen, The use of metamodeling techniques for optimization under uncertainty, Structural and Multidisciplinary Optimization 25 (2) (2003) 99–116.

[34] Y. Jin, A comprehensive survey of fitness approximation in evolutionary computation, Soft Computing 9 (1) (2005) 3–12.

[35] G. G. Wang, S. Shan, Review of metamodeling techniques in support of engineering design optimization, Journal of Mechanical Design 129 (4) (2007) 370–380.

[36] V. C. Chen, K.-L. Tsui, R. R. Barton, M. Meckesheimer, A review on design, modeling and applications of computer experiments, IIE transactions 38 (4) (2006) 273–291.

[37] R. Jin, W. Chen, T. W. Simpson, Comparative studies of metamodelling techniques under multiple modelling criteria, Structural and Multidisciplinary Optimization 23 (1) (2001) 1–13.

[38] H. Fang, M. F. Horstemeyer, Global response approximation with radial basis functions, Engineering Optimization 38 (04) (2006) 407–424.

[39] D. Xiu, J. S. Hesthaven, High-Order Collocation Methods for Differential Equations with Random Inputs, Siam J. Sci. Comput. 27 (3) (2005) 1118–1139.

[40] V. Barthelmann, E. Novak, K. Ritter, High dimensional polynomial interpolation on sparse grids, Advances in Computational Mathematics 12 (2000) 273–288.

[41] X. Ma, N. Zabaras, An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations, Journal of Computational Physics 228 (8) (2009) 3084–3113.

[42] S. Chen, C. F. M. Cowan, P. M. Grant, Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks, IEEE Transactions on Neural Networks 2 (2) (1991) 302–309.

[43] J. Park, I. Sandberg, Universal approximation using radial-basis-function networks, Neural Computation 3 (1991) 246–257.

[44] S. Haykin, Neural networks: a comprehensive foundation, 1994, Mc Millan, New Jersey.

[45] H. Li, S. S. Mulay, Meshless methods and their numerical properties, CRC Press, 2013.

[46] F. Trochu, A Contouring Program Based on Dual Kriging Interpolation, Engineering with Computers 9 (1993) 160–177.

[47] S. N. Lophaven, H. B. Nielsen, J. Søndergaard, Dace-a matlab kriging toolbox, version 2.0, Tech. rep. (2002).

[48] H. Song, K. K. Choi, D. Lamb, A Study on Improving the Accuracy of Kriging Models by Using Correlation Model/Mean Structure Selection and Penalized Log-Likelihood Function, in: 10th World Conference on Structural and Multidisciplinary Optimization, 2013, pp. 1–10.

[49] S. A. Smolyak, Quadrature and interpolation formulas for tensor products of certain classes of functions, Dokl. Akad. Nauk SSSR 4 (1963) 240–243.

[50] M. Sgarbi, V. Colla, L. M. Reyneri, A comparison between weighted radial basis functions and wavelet networks., in: ESANN, Citeseer, 1998, pp. 13–20.

[51] H. Wendland, Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, Advances in Computational Mathematics 4 (1) (1995) 389–396. doi:10.1007/BF02123482.

[52] Z. Wu, Compactly supported positive definite radial functions, Advances in Computational Mathematics 4 (1) (1995) 283–292. doi:10.1007/BF03177517.

[53] B. S. Morse, T. S. Yoo, P. Rheingans, D. T. Chen, K. R. Subramanian, Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions, in: ACM SIGGRAPH 2005 Courses, ACM, 2005, p. 78.

[54] S. Ferrari, M. Maggioni, N. A. Borghese, Multiscale Approximation with Hierarchial Radial Basis Function Networks, IEEE Transactions on Neural Networks 15 (1) (2004) 178–188.

[55] B. Fornberg, C. Piret, A stable algorithm for flat radial basis functions on a sphere, SIAM Journal on Scientific Computing 30 (1) (2007) 60–80.

[56] G. E. Fasshauer, M. J. McCourt, Stable evaluation of gaussian radial basis function interpolants, SIAM Journal on Scientific Computing 34 (2) (2012) A737–A762.

[57] B. Fornberg, T. A. Driscoll, G. Wright, R. Charles, Observations on the behavior of radial basis function approximations near boundaries, Computers & Mathematics with Applications 43 (35) (2002) 473–490. doi:10.1016/S0898-1221(01)00299-1.

[58] J. Leonard, M. Kramer, L. Ungar, Using radial basis functions to approximate a function and its error bounds, IEEE Transactions on Neural Networks 3 (4) (1992) 624–627. doi:10.1109/72.143377.

[59] S. Chen, P. M. Grant, C. F. N. Cowan, Orthogonal least-squares algorithm for training multioutput radial basis function networks, IEEE Proceedings-F 139 (6) (1992) 378–384.

[60] N. Benoudjit, M. Verleysen, On the Kernel Widths in Radial-Basis Function Networks, Neural Processing Letters 18 (2003) 139–154.

[61] C. Dachapak, S. Kanae, Z.-J. Yang, K. Wada, Orthogonal Least Squares for Radial Basis Function Network in Reproducing Kernel Hilbert Space, in: IFAC Workshop on Adaptation and Learning in Control and Signal Processing and IFAC Workshop on Periodic Control Systems, 2004, pp. 847–852.

[62] K.-L. Du, M. N. Swamy, Neural networks in a softcomputing framework, Springer, 2006.

[63] G.-B. Huang, P. Saratchandran, N. Sundararajan, A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation, IEEE Transactions on Neural Networks 16 (1) (2005) 57–67. doi:10.1109/TNN.2004.836241.

[64] R. Neruda, P. Vidnerova, Learning Errors by Radial Basis Function Neural Networks and Regularization Networks, International Journal of Grid and Distributed Computing 1 (2) (2009) 49–58.

[65] K. M. Tao, A closer look at the radial basis function (rbf) networks, 1993 Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers (1993) 401–405.

[66] J. Moody, C. J. Darken, Fast Learning in Networks of Locally-Tuned Processing Units, Neural Computation 1 (1989) 281–294.

[67] N. Cressie, Statistics for spatial data, Terra Nova 4 (5) (1992) 613–617.

[68] G. Matheron, Principles of Geostatics, Economic Geology 58 (1963) 1246–1266.

[69] L. Zhao, K. Choi, I. Lee, D. Gorsich, A metamodeling method using dynamic kriging and sequential sampling, in: The 13th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Fort Worth, TX, Sept, 2010, pp. 13–15.

[70] S. N. Lophaven, H. B. Nielsen, J. Søndergaard, Aspects of the matlab toolbox dace, Tech. rep., Informatics and Mathematical Modelling, Technical University of Denmark, DTU (2002).

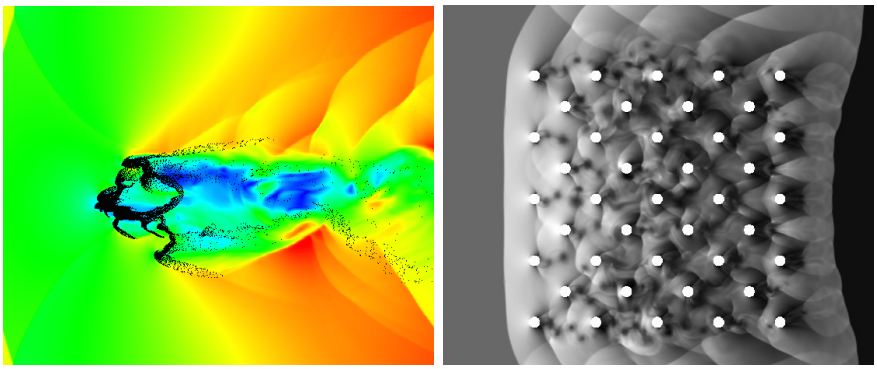
[71] M. Stein, Large sample properties of simulations using latin hypercube sampling, Technometrics 29 (1987) 143–151.

[72] T. Goel, R. Haftka, W. Shyy, L. Watson, Pitfalls of using a single criterion for selecting experimental designs, International Journal of Numerical Methods in Engineering 75 (2008) 127 – 155.

[73] J. Burkardt, M. Gunzburger, J. Peterson, R. Brannon, User manual and supporting information for library of codes for centroidal voronoi placement and associated zeroth, first, and second moment determination, Tech. rep. (2002).

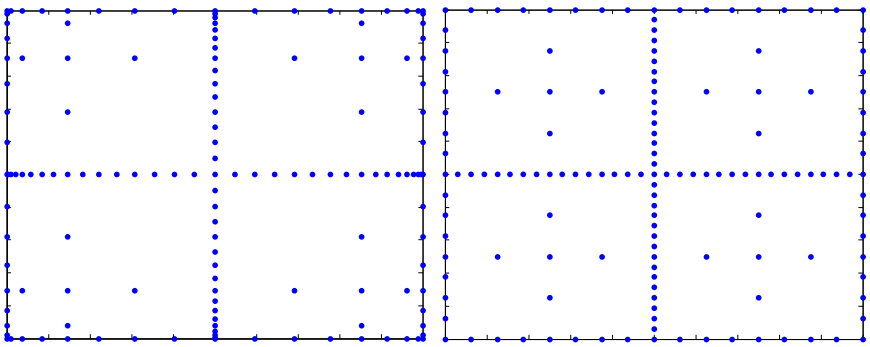
[74] V. Boiko, V. P. Kiselev, S. P. Kiselev, A. Papyrin, S. Poplavsky, V. Fomin, Shock wave interaction with a cloud of particles, Shock Waves 7 (1997) 275–285.

List of Figures



(a) (b)

Figure 1: Two-dimensional examples of the (a) macroscale interaction of a large number of modeled particles with a right running normal shock and (b) full resolution meso-scale computation of a small number of particles interacting with a right running normal shock.



(a) (b)

Figure 2: The distribution of nodes from a two-dimensional fifth level Smolyak Sparse Grid based on (a) Chebyshev polynomial end knots and (b) a uniform Newton-Cotes grid.

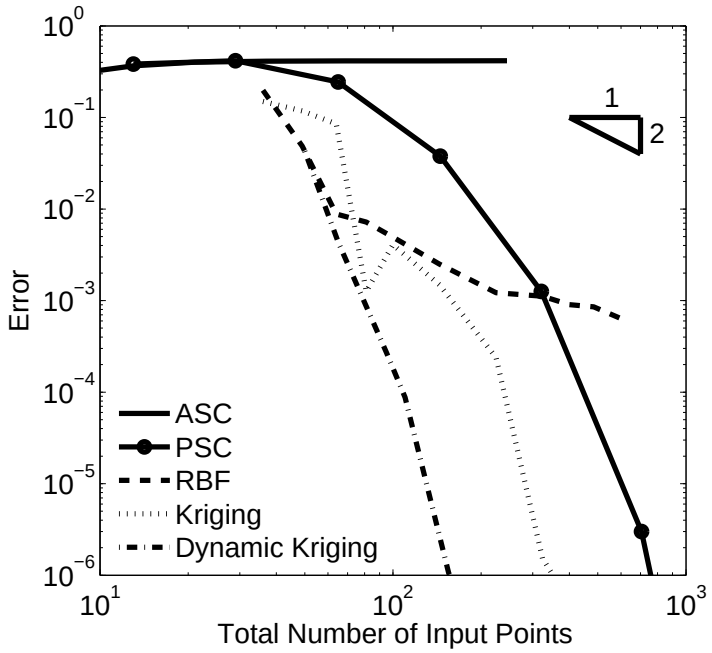


Figure 3: Error plot showing the convergence rates on approximating the smooth harmonic test function from (12).

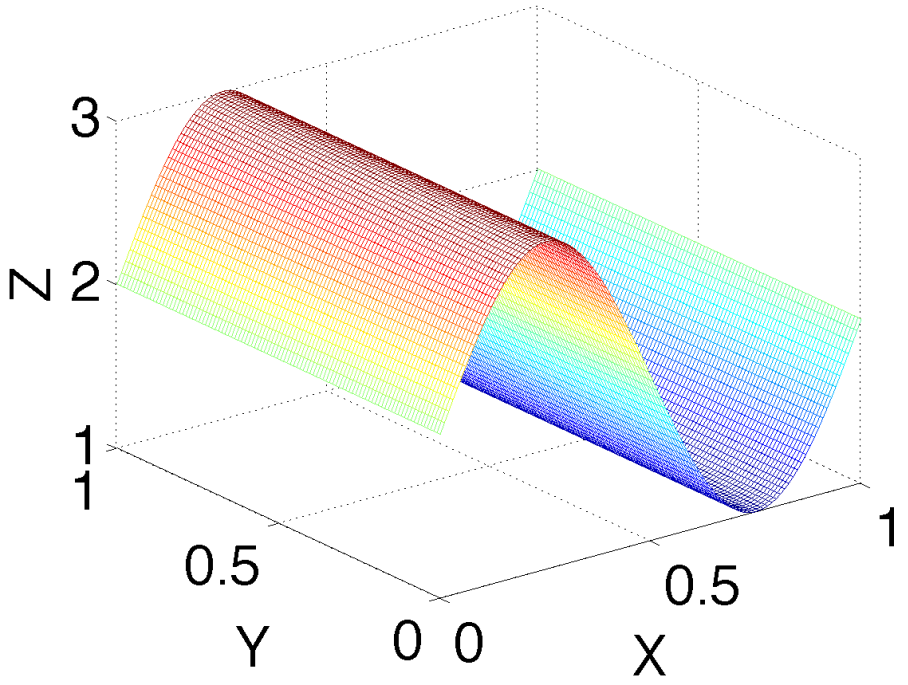


Figure 4: The hypersurface of the function given by (12) as approximated by the ASC method.

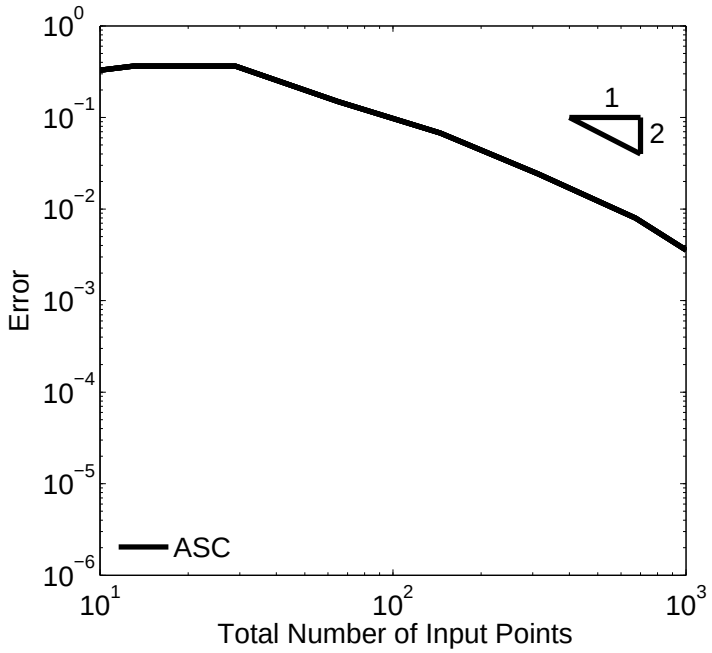
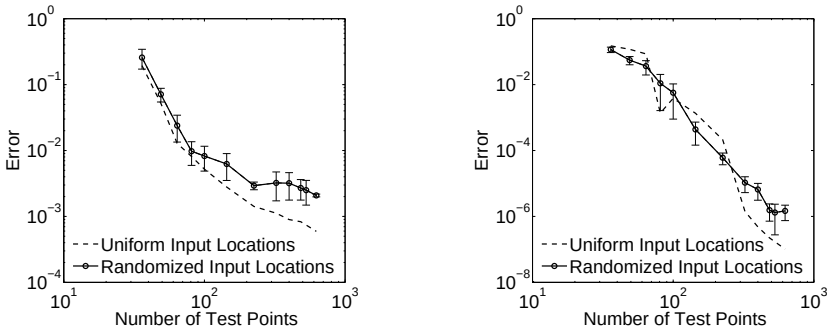
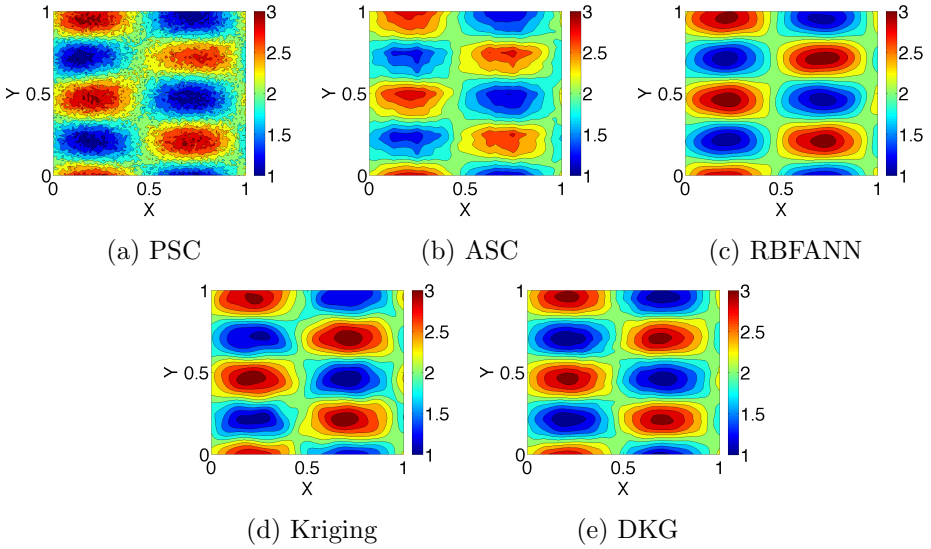


Figure 5: Error convergence of the approximation of the shifted harmonic test function from (13) using an ASC method.



(a) (b)

Figure 6: Error plot showing the convergence rates on interpolating the harmonic test function from Eqn. 12 using (a) an RBF method and (b) a Kriging method.



(a) PSC (b) ASC (c) RBFANN

(d) Kriging (e) DKG

Figure 7: Representative hypersurfaces for the noisy shifted harmonic function using (a) a PSC method, (b) an ASC method, (c) an RBF ANN, (d) a Kriging and (e) a DKG method.

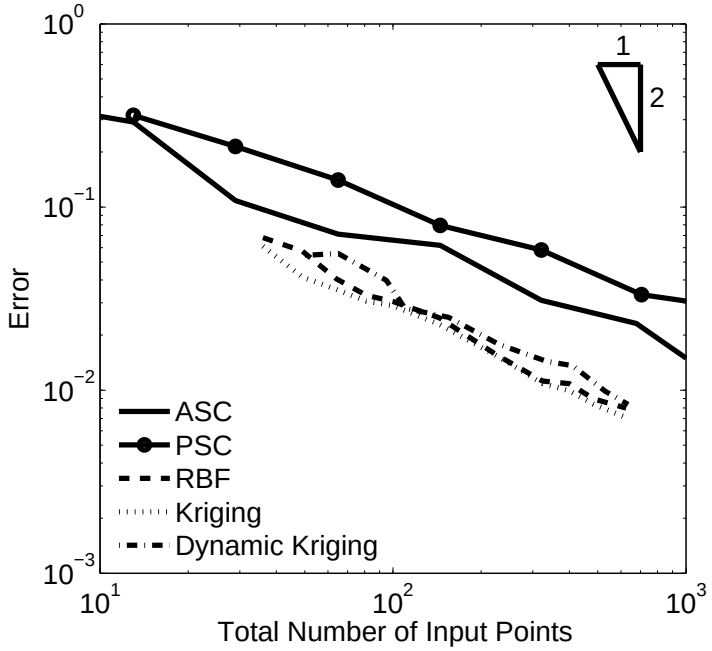


Figure 8: Error plot showing the convergence rates on approximating the test function from (14) using a PSC method, an ASC method, an RBFANN, a Kriging method and the DKG method.

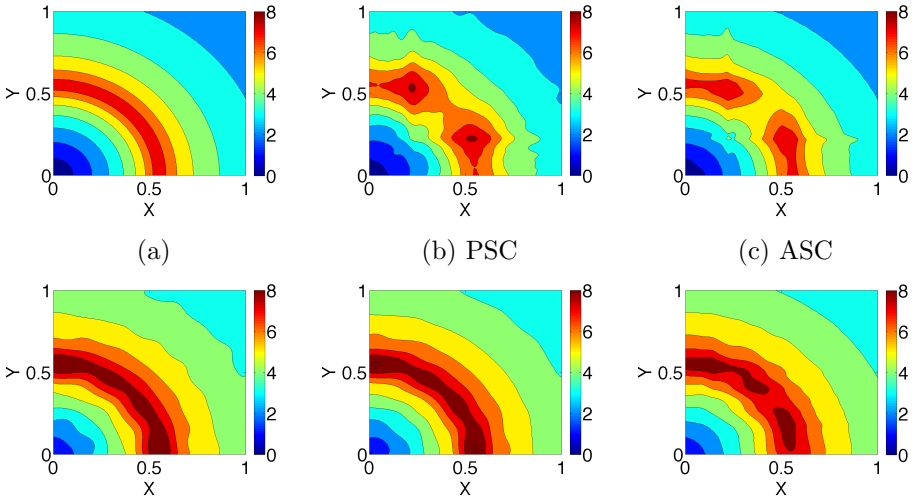
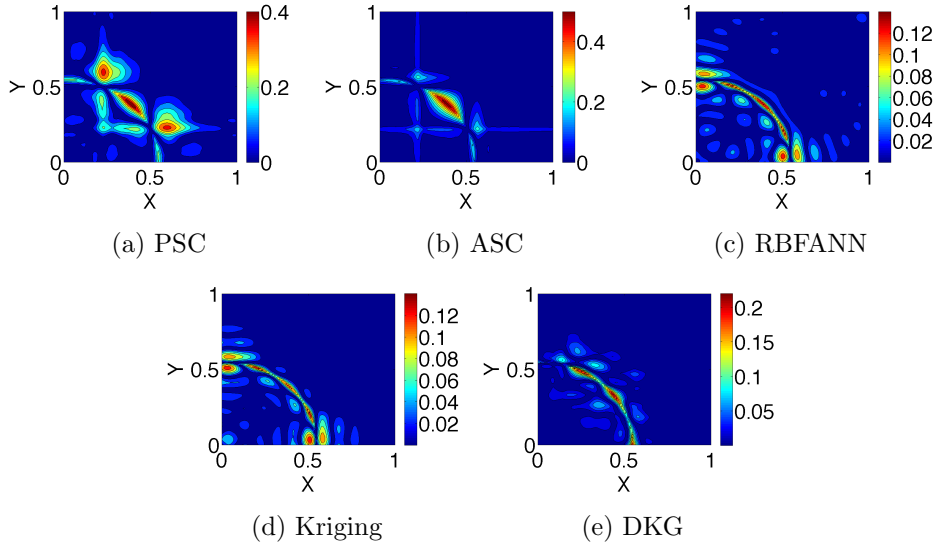


Figure 9: The (a) exact contours and representative contour plots for the test function from (14) using (b) a PSC method, (c) an ASC method, (d) an RBF ANN, (e) a Kriging method and (f) the DKG method.



(a) PSC (b) ASC (c) RBFANN

(d) Kriging (e) DKG

Figure 10: Local error plots for the approximation of the test function from (14) using (a) a PSC method, (b) an ASC method, (c) an RBF ANN, (d) a Kriging method, and (e) the DKG method.

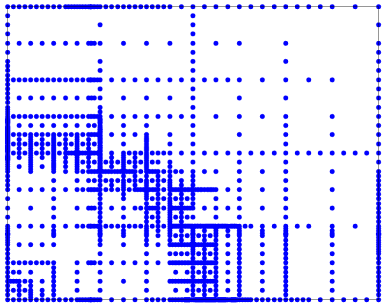


Figure 11: Locations of nodes in a 12th level Smolyak sparse grid, refined adaptively using hierarchical surpluses with a maximum error of $\epsilon = 0.01$, interpolating the function in (14).

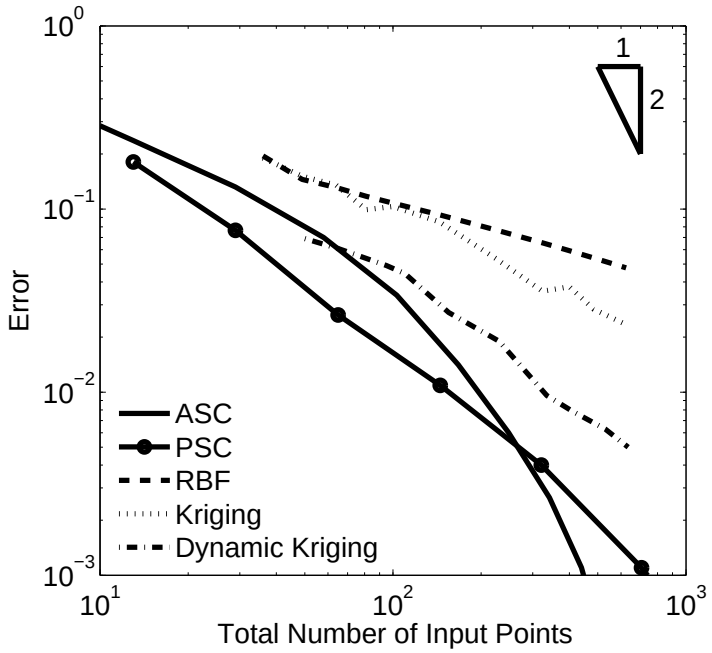


Figure 12: Error plot showing the convergence rates on approximating the drag model of Boiko et. al. [74] using a PSC method, an ASC method, an RBFANN, a Kriging method and the DKG method.

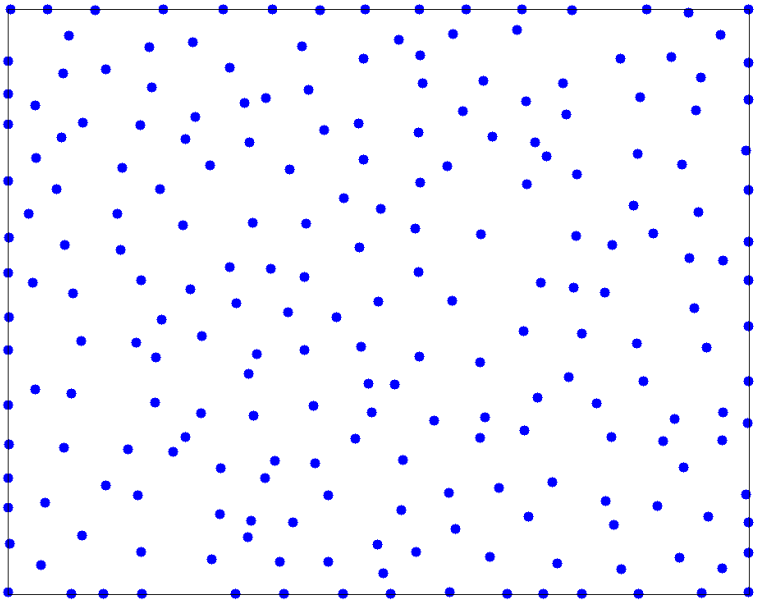
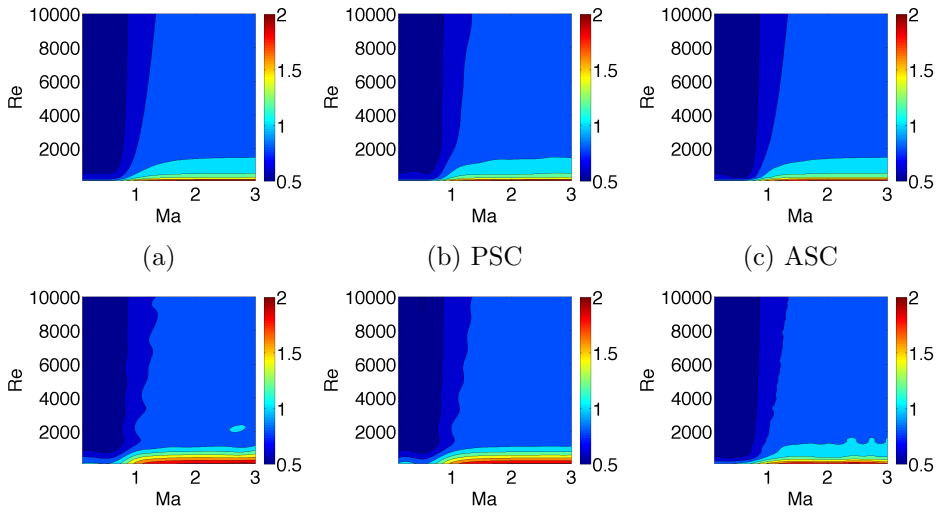


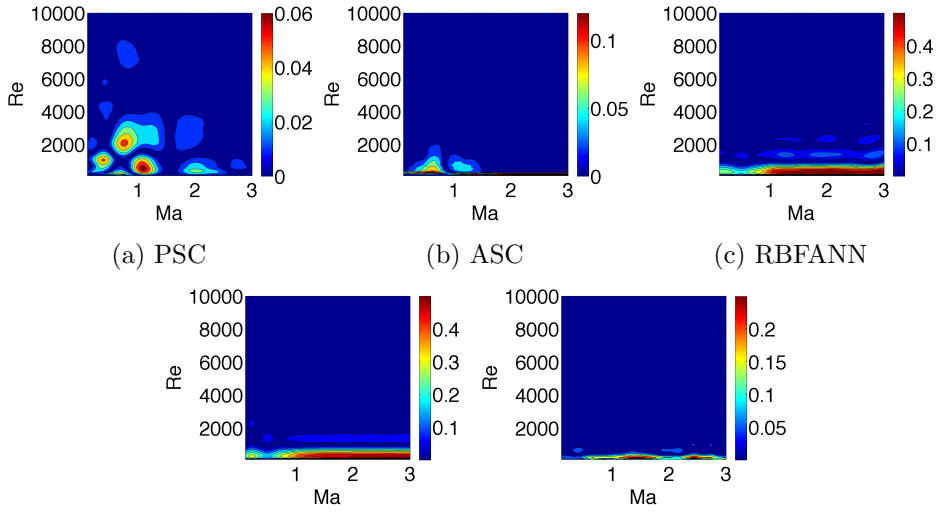
Figure 13: The distribution of training points in the DKG method



(a) (b) PSC (c) ASC

(d) RBFANN (e) Kriging (f) DKG

Figure 14: The (a) exact contours and representative contour plots for the drag coefficient of Boiko et. al.[74] using (b) a PSC method, (c) an ASC method, (d) an RBF ANN, (e) a Kriging method and (f) the DKG method



(a) PSC (b) ASC (c) RBFANN

(d) Kriging (e) DKG

Figure 15: Local error plots for the approximation of drag model of Boiko et. al. [74] using (a) a PSC method, (b) an ASC method, (c) an RBF ANN, (d) a Kriging method and (e) the DKG method.

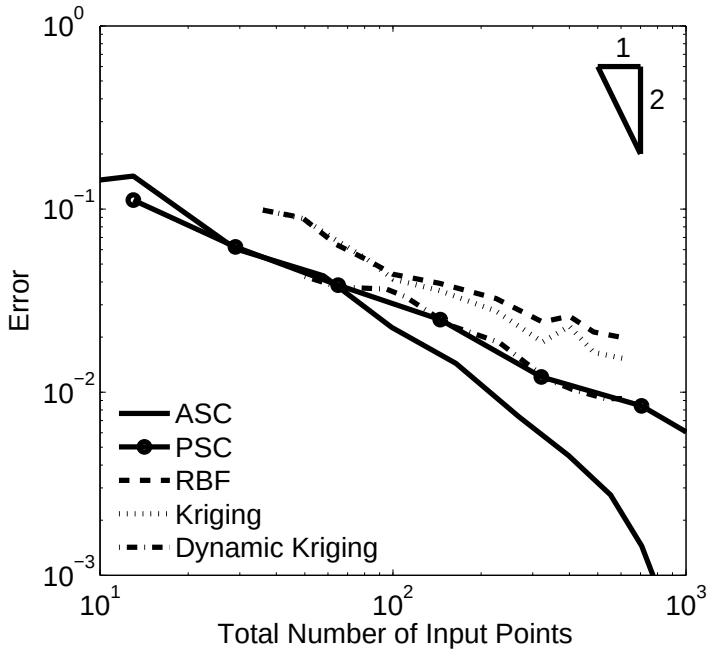
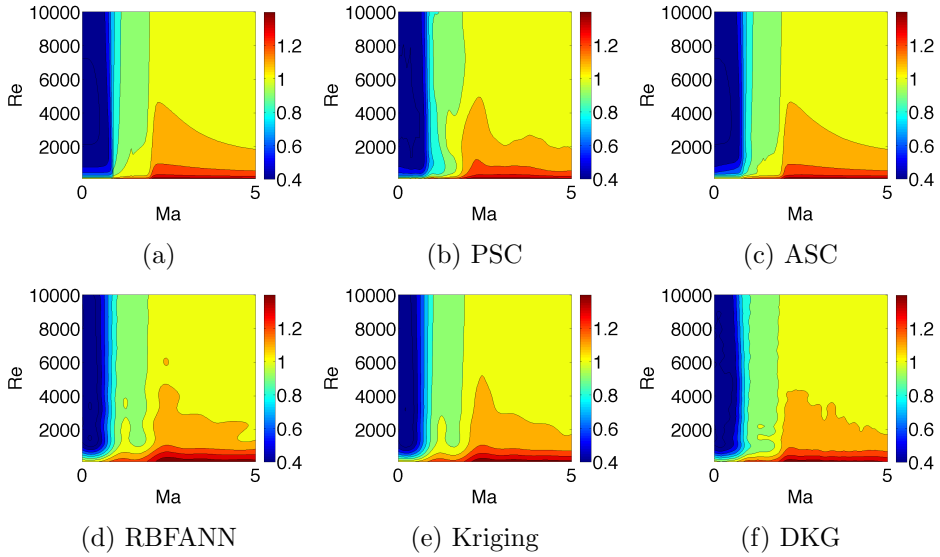


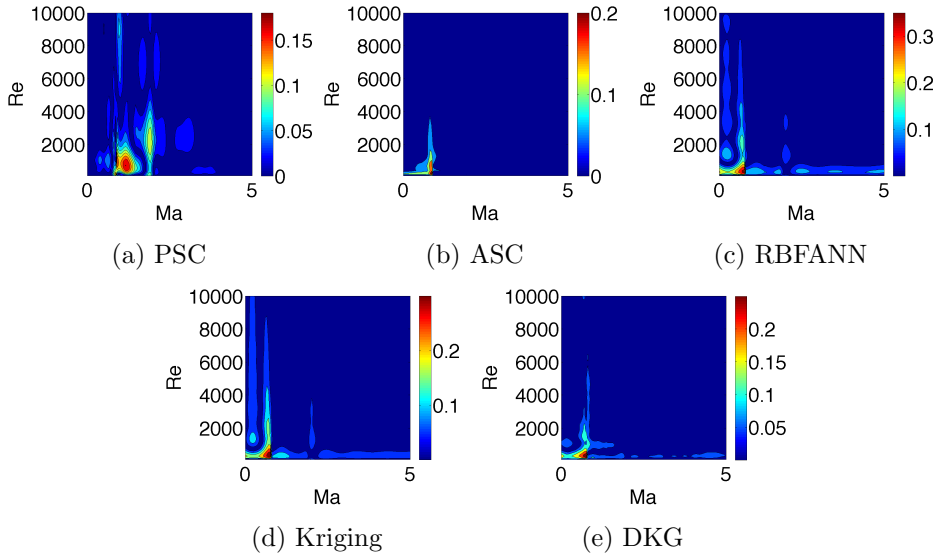
Figure 16: Error plot showing the convergence rates on approximating the drag model of Loth [24] using a PSC method, an ASC method, an RBFANN, a Kriging method and the DKG method.



(a) (b) PSC (c) ASC

(d) RBFANN (e) Kriging (f) DKG

Figure 17: The (a) exact contours and representative contour plots for the drag coefficient of Loth [24] using (b) a PSC method, (c) an ASC method, (d) an RBFANN method, (e) a Kriging method and (f) the DKG method.



(a) PSC (b) ASC (c) RBFANN

(d) Kriging (e) DKG

Figure 18: Local error plots for the approximation of drag model of Loth [24] using (a) a PSC method, (b) an ASC method, (c) an RBFANN method, (d) a Kriging method and (e)the DKG method.

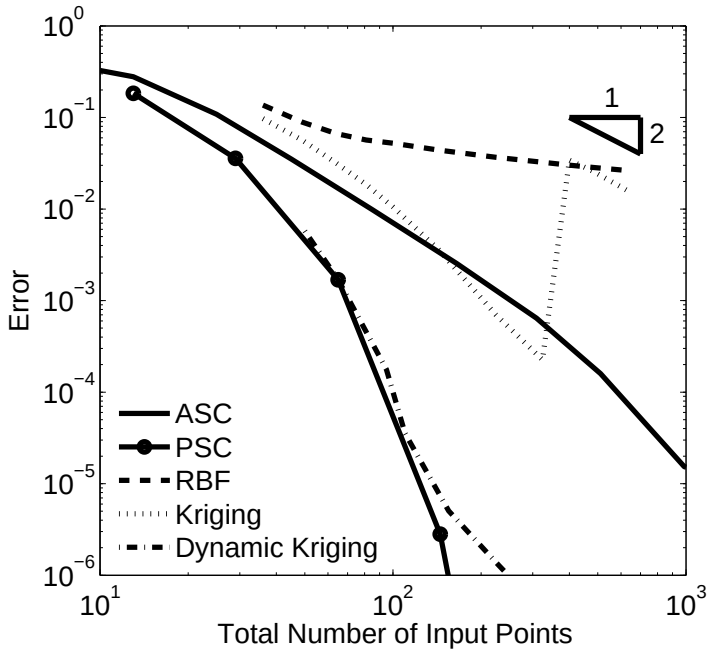
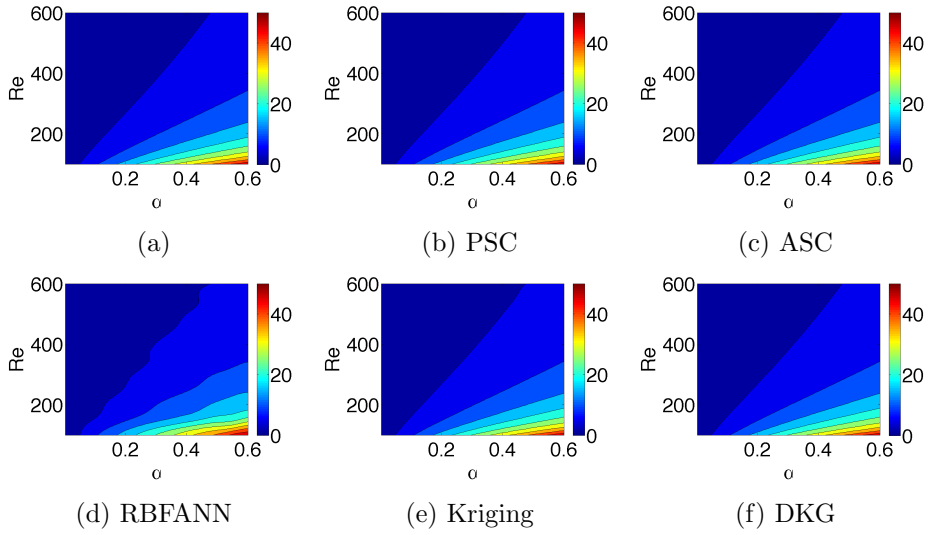


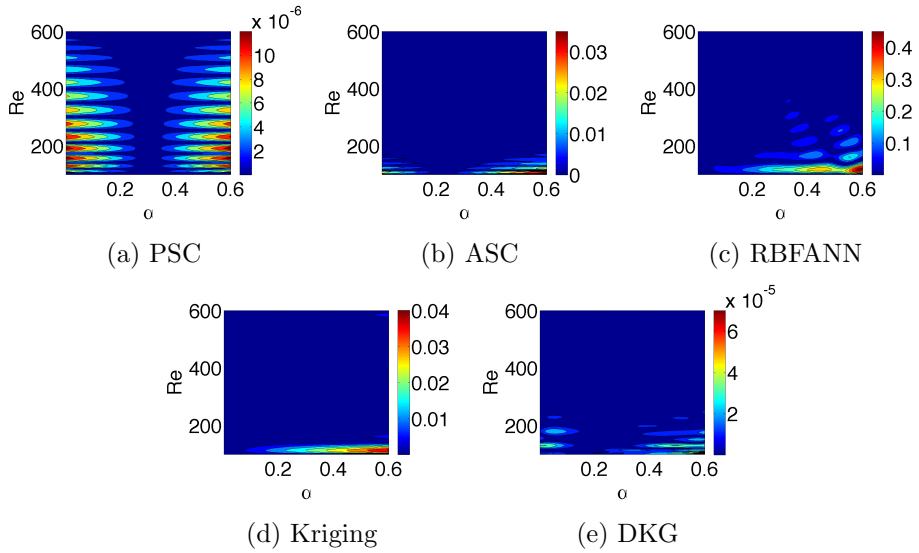
Figure 19: Error plot showing the convergence rates on approximating the drag model of Tong et. al. [26] using a PSC method, an ASC method, an RBFANN, a Kriging method and the DKG method.



(a) (b) PSC (c) ASC

(d) RBFANN (e) Kriging (f) DKG

Figure 20: The (a) exact contours and representative contour plots for the drag coefficient of Tong [26] using (b) a PSC method, (c) an ASC method, (d) an RBFANN, (e) a Kriging method and (f) the DKG method.



(a) PSC (b) ASC (c) RBFANN

(d) Kriging (e) DKG

Figure 21: Local error plots for the approximation of drag model of Tong et. al. [26] using (a) a PSC method, (b) an ASC method, (c) an RBF ANN, (d) a Kriging method and (e) the DKG method.

List of Tables

Correlation Function	$\gamma_k(\theta, d_k)$
Exponential	$exp(-\theta_k \lvert d_k \rvert)$
General Exponential	$exp\left(-\theta_k \lvert d_k \rvert^{\theta_{n+1}}\right);\ 0 < \theta_{n+1} \leq 2$
Gaussian	$exp\left(-\theta_k d_k^2\right)$
Linear	$\max\{0, 1 - \theta_k \lvert d_k \rvert\}$
Spherical	$1 - 1.5\xi_k + 0.5\xi_k^3,\ \xi_k = min\{1, \theta_k \lvert d_k \rvert\}$
Cubic	$1 - 3\xi_k^2 + 2\xi_k^3,\ \xi_j = min\{1, \theta_k \lvert d_k \rvert\}$
Spline	$\begin{array}{ll} 1 - 1.5\xi_k^2 + 30\xi_k^3 & \text{for } 0 \leq \xi_k \leq 0.2 \\ 1.25(1 - \xi_k)^3 & \text{for } 0.2 < \xi_k < 1 \\ 0 & \text{for } \xi_k \geq 1 \end{array}\ ;\ \xi_k = \theta_k \lvert d_k \rvert$

Table 1: List of correlation functions

Number of Inputs	θ(Kriging)	θ(DKG)
36	6.13	0.8898
64	6.13	0.4523
81	0.322	0.3741
100	0.161	0.3683
144	0.161	0.2863

Table 2: Correlation parameters estimated by the Kriging method to approximate the hypersurface given by 12

Number of Inputs	θ(Kriging)	θ(DKG)
36	0.76655	0.9913
64	0.76655	1.3370
81	0.76655	2.7159
100	0.76655	2.5089
144	0.76655	2.8116
225	0.76655	3.5402
324	0.76655	5.8234
400	20	6.6144
529	20	8.3136
625	20	9.0773

Table 3: Correlation parameters estimated by the Kriging method and the DKG method to approximate the hypersurface given by 19