Fall 2016

Multiscale modeling of multimaterial systems using a Kriging based approach

Oishik Sen
University of Iowa

MULTISCALE MODELING OF MULTIMATERIAL SYSTEMS USING A KRIGING BASED APPROACH

by

Oishik Sen

A thesis submitted in partial fulfillment
of the requirements for the Doctor of Philosophy
degree in Mechanical Engineering in the
Graduate College of
The University of Iowa

December 2016

Thesis Supervisor: Professor H.S. Udaykumar

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

_____

PH.D. THESIS

_____

This is to certify that the Ph.D. thesis of

Oishik Sen

has been approved by the Examining Committee for
the thesis requirement for the Doctor of Philosophy degree
in Mechanical Engineering at the December 2016 graduation.

Thesis Committee: _____
H.S. Udaykumar, Thesis Supervisor

_____
K.K. Choi

_____
Jia Lu

_____
Pablo Carrica

_____
Edward Sander

ACKNOWLEDGEMENTS

I would like to thank my thesis supervisor Dr. H.S. Udaykumar for his continuous support, guidance and patience in course of my PhD studies. In addition to helping me in my academic subject of interest, I am grateful to him for the tremendous effort he has put in honing my skills as an overall researcher by constantly working with me on my technical writings and professional presentations. Choosing to come to the University of Iowa and doing a PhD under him was possibly one of the best decisions I have taken till date.

I would also like to thank my master's supervisor, Dr. S. Arjun Tekalur at Michigan State University for teaching me the basics of research and helping me develop my skills of reasoned questioning. I gratefully acknowledge the help and support of our collaborators - Dr. Gustaaf Jacobs at San Diego State University and Dr. K.K. Choi at the University of Iowa for the constant guidance and help provided in course of my PhD. I would like to thank Dr. Nicholas Gaul and Dr. Hyeongjin Song at the University of Iowa and Dr. Sean Davis at the San Diego State University for helping me develop a successful collaboration during my PhD thesis and in spite of their overwhelming schedule, for taking out the time to constantly help me in several parts of my thesis and journal papers. I would like to my other committee members, Dr. Jia Lu, Dr. Pablo Carrica, and Dr. Edward Sander for taking the time to carefully review my research and providing valuable inputs to ensure quality work.

A successful PhD results not merely from five/six years of focused academic research but also because of constant academic guidance during the formative years of education. I am gratefully indebted to the guidance provided by my teachers at my high school – South Point High School (SPHS) and my undergraduate professors at the National Institute of Technology, Calicut (NITC). I would like to thank Dr. Mohammed Ameen and Dr. Vijaykumar at NITC and Mrs. Milly Mukherjee, Mr. Sujit Bose and Mr. Sandeep Chattopadhyay at SPHS for their guidance and support during the early years of my education. My teachers Mr Amal Laha and Mr. Sujoy Bose in India also deserve special mention.

It is by no means an exaggeration to say that I owe everything I am to my parents Mr. Pallav Sen and Mrs. Sudipta Sen, my grandparents, Mr. Arindam Chowdhury, Mrs Nandita Chowdhury, Late Mr. Paramesh Chandra Sen, Mrs Jyostna Sen and my aunt Mrs. Simonti Chakraborty. A couple of lines of grateful acknowledgement is by no means sufficient to acknowledge their roles in my life. Similarly, the roles of my best friends Nirmal Rai, Nisarga Phatak, Avishek Chakravorty

and his wife Nibedita Saha in my life cannot be emphasized enough; they have always been a source of support for me and are no less than family to me. I am also blessed to have the support of their family –Mr. Saikat and Mrs. Baishakhi Chakravorty, Mr. Rakesh and Mrs Meera Rai, Mrs Sunita Pandey, Vishal and Amrita Rai and Mr. Lakhsman and Mrs. Chandralekha Phatak in my life. Each of them have special roles in my life and represent more than mere names to me.

I am thankful to Dr. Jayadipta Ghosh, Dr. Sreeparna Sengupta and Mr. Hironmoy Sengupta and Mrs. Basanti Sengupta for encouraging me to pursue graduate studies. My friends at Michigan State – Parimal Maity, Abhishek Dutta, Swatilekha Bhattacharya, Pavel Roy Paladhi, Rahul and Koyeli Banerjee, my friends in The University of Iowa – Suman Ghorai, Bhakti Bapat, Abhinaba Gupto, Ipshita Bhattacharya, my roommates -Abhay Shah and Aashay Shah, my school friends – Arnab Sengupta, Snigdhadeb Kundu, Ahijit Sarkar, Ankit Roy, Ayan Chakraborty, Pragnajyoti Mukherjee, Rusha Chatterjee, Suvojit Ghosh and my friends in NITC – Ashwani Singal, Gaurav Soni, Harshdeep Singh, Abhishek Chakrabarty deserve special mention for their roles in both my academic and personal lives.

Last, but by no means the least, I would like to thank my past and present labmates – Anil Kapahi, John Mousel, Seth Dillard, Mehrdaad, Fazlolah Mohagegh, Liza Sreshta, Keshav Venkat, Piyusha Gade, Vishwanath Somashekhar, Kayley Lain, Jared Becker, Aaron Goddard, Alexia De Brauer, Pratik Das, Sidhartha Roy, Mike Augspurger, Anas Nassar and Moustafa Ibrahim for always maintaining a academic environment in the lab conducive to productivity and for making me want to come to the lab everyday.

ABSTRACT

The present work presents a framework for multiscale modeling of multimaterial flows using surrogate modeling techniques in the particular context of shocks interacting with clusters of particles. The work builds a framework for bridging scales in shock-particle interaction by using ensembles of resolved mesoscale computations of shocked particle laden flows. The information from mesoscale models is “lifted” by constructing metamodels of the closure terms - the thesis analyzes several issues pertaining to surrogate-based multiscale modeling frameworks.

First, to create surrogate models, 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. 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.

After this, closure laws for drag are constructed in the form of surrogate models derived from real-time resolved mesoscale computations of shock-particle interactions. The mesoscale computations are performed to calculate the drag force on a cluster of particles for different values of Mach Number and particle volume fraction. Two Kriging-based methods, viz. the Dynamic Kriging Method (DKG) and the Modified Bayesian Kriging Method (MBKG) are evaluated for their ability to construct surrogate models with sparse data; i.e. using the least number of mesoscale simulations. It is shown that unlike the DKG method, the MBKG method converges monotonically even with noisy input data and is therefore more suitable for surrogate model construction from numerical experiments.

In macroscale models for shock-particle interactions, Subgrid Particle Reynolds’ Stress Equivalent (SPARSE) terms arise because of velocity fluctuations due to fluid-particle interaction in the subgrid/meso scales. Mesoscale computations are performed to calculate the SPARSE terms and the kinetic energy of the fluctuations for different values of Mach Number and particle volume fraction. Closure laws for SPARSE terms are constructed using the MBKG method. It is found

that the directions normal and parallel to those of shock propagation are the principal directions of the SPARSE tensor. It is also found that the kinetic energy of the fluctuations is independent of the particle volume fraction and is 12-15% of the incoming shock kinetic energy for higher Mach Numbers.

Finally, the thesis addresses the cost of performing large ensembles of resolved mesoscale computations for constructing surrogates. Variable fidelity techniques are used to construct an initial surrogate from ensembles of coarse-grid, relative inexpensive computations, while the use of resolved high-fidelity simulations is limited to the correction of initial surrogate. Different variable-fidelity techniques, viz the Space Mapping Method, RBFs and the MBKG methods are evaluated based on their ability to correct the initial surrogate. It is found that the MBKG method uses the least number of resolved mesoscale computations to correct the low-fidelity metamodel. Instead of using 56 high-fidelity computations for obtaining a surrogate, the MBKG method constructs surrogates from only 15 resolved computations, resulting in drastic reduction of computational cost.

PUBLIC ABSTRACT

This thesis develops numerical framework for bridging scales in problems with wide variety of length and time scales. The calculations involved in such multiscale problems in engineering are expensive and require tremendous computational powers even on supercomputers. A robust method pf performing these classes of computations is described in this work; while the thesis uses a particular example of shocks interacting with debris, the methods presented are generic and may be used for a wide variety of multiscale problems in engineering applications.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

1.1 INTRODUCTION

A wide variety of problems with distinct meso- and macroscales appear in several important engineering applications. Examples of typical multiscale problems are the dynamics of particle-laden gases [1,2], deformation of heterogeneous materials such as bones [3–5], concrete [6,7], heterogeneous explosives[8–11], sediment transport in river beds [12], and meso-scale models of blood flow[13]. In such problems, macroscale computations typically resolve a limited range of length and time scales. The unresolved/subgrid scales are modeled using closure laws in homogenized (volume-averaged/filtered/coarse-grained) macroscale systems of equations. Traditionally, such closure laws were obtained from physical experiments, canonical theoretical constructs or phenomenological arguments. With increasing computational power and improved physical models and algorithms, it is now possible to derive closure laws from highly resolved mesoscale simulations. This work addresses the issue of efficient and accurate closure model construction from highly resolved meso-scale simulations in the context of shocked particle-laden flows. The techniques examined in this work can be applied generally to the construction of closure models in any multi-scale modeling problems where there is a distinct separation of scales.

1.2 CLOSURE LAWS IN MACROSCALE MODELS

1.2.1 Closure Laws in a General Macroscale Model in Problems with Multiple Length and Time Scales

In problems with multiple length and time scales, the macroscale variables can be represented by the vector **A** and satisfy the system of equations given by [14]

$$\mathbf{F}(\mathbf{A}, \mathbf{D}) = \mathbf{0} \tag{1}$$

where **F** is the macroscopic operator. **D** is the information at the macroscale, which depends on the mesoscale interactions and needs to be provided to close Equation (1). Once **D** is known, **A** can be determined using Equation (1)

The mesoscale state variable **a** is a solution of the system of equations given by [14]

$$\mathbf{f}(\mathbf{a})\big|_{\mathbf{b}} = \mathbf{0} \tag{2}$$

where **f** is a mesoscale operator and **0** is the null-vector. In the mesoscale system of equations, **b** represents the constraints (e.g. boundary/initial conditions) on the mesoscale model; it is typically prescribed by local macroscale dynamics. In a multiscale modeling framework, the macroscale supplies the constraints **b** to the mesoscale model and the mesoscale returns the missing information, **D** to the macroscale model in form of a closure law. A closure law is a model of the form

$$\mathbf{D} = D(\mathbf{b}) \tag{3}$$

where the constraints ***b*** define the parameter space for **D**.

1.2.2 Closure Laws in Macroscale Models of Shock Particle Interactions

For the particular problem of shocked gas-particle flows[1,15], the number of particles in the flow is so large that macroscale models cannot explicitly track the dynamics of each such particle. Macroscale computations are performed using Eulerian-Lagrangian [16] and Eulerian-Eulerian [17] approaches. In the Eulerian-Lagrangian (E-L) approach, a popular macroscale model for shock particle interaction is the Cloud-in-Cell (CIC) or the Particle Source in Cell (PSIC) method [16,18–23]. In this method, the fluid phase is computed in the Eulerian frame while particles are tracked in Lagrangian fashion. In the interest of computational tractability, particularly for large ensembles of particles, E-L methods typically track “computational” particles: pseudo-particles that are agglomerates of a large number of physical particles ([16,20,24]).The computational particles are modeled as singular point sources which couple with the carrier fluid through momentum exchange modeled via source terms (*D)* in the macroscopic fluid equations. These exchange terms depend on the local macroscopic conditions, e.g. shock strength and particle

loading, which define the parameter space for obtaining the closure law in the form given by Equation (3).

Recently [Reference], the traditional CIC method has been improved model by adding the effects of velocity fluctuations in the particle and fluid phases within a computational particle. In this improved approach, the instantaneous momentum-equation for a computational particle evolves as [Reference]

$$\frac{dv_i^p}{dt} = f(\bar{a}_i)\bar{a}_i + \frac{df(\bar{a}_j)}{d\bar{a}_j}\overline{a'_\iota a'_J} \tag{4}$$

Here $\bar{a}_\iota = \bar{u}_i - \bar{v}_i^p$ is the mean slip velocity of the fluid, $\bar{u}_i$ and $\bar{v}_i^p$ are the mean velocities of the fluid and the computational particle and $a_i' = u_i' - v_i^{p\prime}$ where u_i' and $v_i^{p\prime}$ are the fluid-phase and particle-phase velocity fluctuations. The first-term in the right-hand side of Equation (4) represents the mean momentum-transfer (drag, lift) between the fluid and the particle phases and arises in typical CIC models. The stress tensor in the second term, $\overline{a'_\iota a'_J}$ is an additional term that is not typical of first-order CIC models. This tensor represents the effect of the velocity fluctuations resulting from the interaction of the gas and particle phases in the subgrid/meso-scale and arise in the macroscale because of Reynolds' averaging over computational particles (hence the name SPARSE, which is an acronym for Subgrid Particle Average Reynolds' Stress). It has been shown that rather than including the drag alone, a macroscale formulation that uses both the SPARSE terms in addition to the drag uses lesser number of computational particle than typical CIC models to trace a cloud of particles interacting with shocks.

1.2.3 Constructing Closure Laws from Mesoscale Simulations of Shock Particle Interactions

To solve the macroscale system of equations, it is necessary to close the drag and the SPARSE terms. There are several semi-empirical drag laws available to close the macroscale gas-particle momentum equations [25–31]. These drag laws are developed via experiments performed in a limited parameter space and extrapolated to other regimes. An alternative, pursued in this and other works [3–5,7,20,32–36] is to construct closure laws using resolved mesoscale simulations. In this approach, mesoscale computations are performed for a range of anticipated flow conditions

*a prio*ri. A numerical model for the closure laws are constructed from these mesoscale experiments. The simulation-based closure model, also known as a metamodel or a "model of a model" [37–39] serves as a surrogate for the meso-scale (subgrid or unresolved) dynamics in the macroscale computations. The macroscale computations utilize the numerical closure models at each time step in each sub-volume of the macroscale domain.

1.3 OBJECTIVES OF THE THESIS

1.3.1 Selecting a Metamodeling Technique for Creating Closure Laws from Numerical Experiments

A key issue in creating surrogate models from mesoscale experiments is choosing an appropriate metamodeling technique from a wide selection of existing methods [39–42]. Because mesoscale computations are expensive, a suitable surrogate modeling technique must contend with the cost of performing large ensembles of numerical experiments. Furthermore, to prevent wastage of computational effort, each additional mesoscale computation should improve the accuracy of the surrogate model – in other words, a metamodeling technique must be monotonically convergent.

The first objective of the thesis is to compare the rates of convergence of different metamodeling methods for different classes of analytical functions and semi-empirical closure laws. In Chapter 2, candidate state-of-the-art surrogate modeling techniques are considered. These are the Stochastic Collocation (PSC) methods [43,44], the Radial Basis Function Method (RBFANN) [45,46] and the Kriging Family of Methods – in particular the Dynamic Kriging Method [37]. These methods are used to create metamodels from different analytical functions and canonical drag laws. The rates of convergence of the metamodeling techniques are quantified for each of these functions and compared with each other. Among the methods evaluated, the Dynamic Kriging method has the highest rate of convergence for all the functions and is identified as a suitable metamodeling method for constructing surrogates from real-time numerical experiments.

1.3.2 Creating Surrogate Models for Drag on Particles from Numerical Experiments of Shock Interacting with Particles

After identifying a suitable metamodeling technique, the second step is to apply the method to create surrogate models from real-time numerical experiments. Noise is inherent to numerical experiments; quantifying the convergence of metamodeling technique in the presence of noise

from real-time numerical experiments is important to create a smooth as well as convergent surrogate modeling framework. Furthermore, mesoscale computations present different length and time scales from the macroscale computations. A quantity of interest to closure laws (such as drag) is a function of the mesoscale length and time scales. However closure laws that are useful for macroscale computations are usually scalar-valued functions, independent of mesoscale length and time scales [26,27,47]. How does one appropriately homogenize quantities like drag over mesoscale length and time scales?

In Chapter 3, numerical computations of shocks interacting with particles are used to construct surrogates for the drag as a function of the shock Mach number, *Ma* and volume fraction, ϕ . A procedure for homogenization of the mesoscale drag – both in time and space is demonstrated. Furthermore, to alleviate the influence of noise in mesoscale experiments, a Modified Bayesian Kriging Method [48] - which is a modification of the Kriging method selected in Chapter 2 - is analyzed for the suitability of creating noise free surrogates from numerical experiments. The MBKG method is used to create surrogates for drag; the convergence of the method is evaluated and the number of mesoscale computations required to create a reasonably accurate surrogate for drag is ascertained.

1.3.3 Creating Surrogate Models for the SPARSE Tensor

To close the macroscale system of equations, both the drag/lift forces as well as the SPARSE tensor $\overline{a'_i a'_j}$ must be quantified in the parameter range corresponding to shock-particle interactions. Furthermore, to evaluate the necessity of the SPRASE terms in the macro-scale model, it is necessary to quantify the contribution of the kinetic energy of the velocity fluctuations to the momentum transfer between the particles and the gas.

Because velocity fluctuations are naturally incorporated in mesoscale simulations, resolved simulations of shock-particle interactions can be used for the above purposes. In Chapter 4, following the procedures for drag, surrogate models for the SPARSE tensor $\overline{a'_i a'_j}$ are obtained from high-fidelity simulations of shock-particle interactions. The surrogates are then used to quantify the kinetic energy of the velocity fluctuations as a function of the shock strength and particle volume fraction.

1.3.4 Construction of Surrogate Models using Variable Fidelity Techniques

In the previous chapters, the closure models for drag and SPARSE terms are constructed for a two dimensional parameter space. It is well known that the number of experiments increases exponentially with the dimension of the parameter space [53]. Mesoscale computations are expensive; each 2D mesoscale computation of interaction of shocks with particles is several hours of compute time in a multiprocessor system. The process becomes even more computationally intensive for more expensive mesoscale computations (e.g. 3D mesoscale computations). Therefore, creating closures based on higher dimensional parameter spaces (e.g. drag as a function of Mach Number, Particle Volume Fraction, Reynolds Number) using ensembles of expensive mesoscale becomes practically impossible [34] because of the high computational cost of performing numerical experiments. In such scenarios, it is useful to explore alternative methods which reduce the computational cost of constructing surrogates from mesoscale numerical experiments.

One such alternative is the variable-fidelity technique [35–42] used commonly in for constructing metamodels in surrogate-based optimization problems. In this approach, ensembles of low-resolution/coarse grid computations are used to create an initial surrogate model; this is subsequently corrected by using only a few high fidelity computations. Chapter 4 evaluates the suitability of different variable fidelity techniques for use in multiscale modeling problems. Three different correction techniques are evaluated based on the computational cost and the error entailed in constructing reasonably accurate surrogates for drag in particles interacting with shock. The rates of convergence of these methods as well as the computational time entailed with performing the high-fidelity computations for correcting the surrogates are compared with each other. The aim of the chapter is to ascertain the suitability of using variable fidelity models for constructing surrogates for multiscale modeling problems.

1.4 CONCLUSION

The thesis evaluates the suitability of using surrogate models for bridging scales in problems with multiple length and time scales. Several state-of-the-art surrogate modeling techniques are evaluated by constructing surrogate models for analytical functions and canonical drag laws. Among these, the Kriging family of methods are found to have the highest rates of convergence and are then used to create surrogates from real-time mesoscale simulations of shock-particle

interactions. It is found that noise is an inherent part of numerical experiments; among the different Kriging methods, the MBKG method is found to contest will noise in experiments. The MBKG method is used to create surrogates for drag as well the SPARSE terms in macroscale models of shocked particle-laden flows. Finally, to reduce the computational cost of constructing surrogates from large ensembles of resolved simulations, the possibility of creating surrogates from relatively inexpensive mesoscale computations is evaluated.

CHAPTER 2. EVALUATION OF CONVERGENCE BEHAVIOR OF METAMODELING TECHNIQUES FOR BRIDGING SCALES IN MULTI_SCALE MULTIMATERIAL SIMULATION

2.1 INTRODUCTION

In multiscale modeling problems, the physics of the micro/meso-scale needs to be represented in macro-scale simulations. This can be achieved by averaging over the heterogeneous micro/meso-scale. In such volume-averaged macro models [19] or homogenized models [30] micro/meso physics appear in the form of closure terms in the macro-scale equations. In general, closure models are obtained in the form of correlations developed in a physical experiment [21,31–33]. 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 can be used as surrogates for physical experiments to obtain closure models connecting the meso-scale physics to the macro-scale.

2.2 METAMODELING FOR BRINDGING SCALES

A metamodel, or a 'model of a model' [34] 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 [35]. 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. 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. 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 multi-scale 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 metamodeling 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 multi-scale framework:

1. An interpolation method; Stochastic Collocation (PSC) methods [37,38] 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 [39,40] 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 [34] and the Dynamic Kriging Method (DKG) [41]are chosen in this study.

2.3 METAMODELING TECHNIQUES

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 $f(\mathbf{x})$: $\Re^n \rightarrow \Re$ is only known at certain discrete (distinct) points, xj. Here, $\Re^n$ [**0**,**1**] is a bounded subspace of $\mathbf{R}^n$, 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.

2.3.1 Stochastic Collocation Methods

Stochastic Collocation (SC) interpolation methods [37,38] rely on sparse grids generated using the Smolyak algorithm [37] 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, $\tilde{f}(\mathbf{x_0})$ is given by

$$\tilde{f}(\mathbf{x_0}) = \sum_{j_1=1}^{m_{i_1}} \cdots \sum_{j_n=1}^{m_{in}} f\left(x_{j_1}^{i_1}, \ldots, x_{j_n}^{i_n}\right)\left(a_{j_1}^{i_1} \otimes \ldots \otimes a_{j_n}^{i_n}\right) \tag{5}$$

where i is the level of interpolation, m_i is the number of inputs required for level i and a_j are the basis functions used in n dimensions. 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 end knots of a Chebyshev polynomial on a Clenshaw–Curtis grid (Figure 3).

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. The grid is locally refined around points where the hierarchical surplus, defined as,

$$\mathbf{w}_{\mathbf{j}}^{\mathbf{i}} = f\left(x_{j_1}^{i_1},...,x_{j_n}^{i_n}\right) - \tilde{f}\left(x_{j_1}^{i_1},...,x_{j_n}^{i_n}\right) \tag{6}$$

at level i on point j is above a threshold value, ε. 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.

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 is on the order of

$$\left\| f(\mathbf{x_0}) - \tilde{f}^{\,i}(\mathbf{x_0}) \right\|_{\infty} = O\left(N^{-2}\left|\log_2 N\right|^{3(n-1)}\right) \tag{7}$$

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

$$\left\| \tilde{f}^{\,i}(\mathbf{x_0})_{PSC} - \tilde{f}^{\,i}(\mathbf{x_0})_{ASC} \right\|_{\infty} \le N_2 \varepsilon \tag{8}$$

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

2.3.2 Radial Basis Function Neural Network

A Radial Basis Function Artificial Neural Network (RBFANN) is a fitting method, particularly suited to function approximation and pattern recognition [39], 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

$$\tilde{f}(\mathbf{x_0}) = \sum_{k=1}^{M} \lambda_k \phi\left(\mathbf{x_0}, \mathbf{x}_{c_k}; \theta_{c_k}\right) \tag{9}$$

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 $\phi\left(\mathbf{x_0}, \mathbf{x}_{c_k}; \theta_{c_k}\right) = \exp\left(-\frac{d^2_{ck}}{\theta^2_{ck}}\right)$ where $d_{ck} = \left\|\mathbf{x_0} - \mathbf{x}_{c_k}\right\|$, is the Euclidean Norm; this basis function has non-compact support. The weights λ_k are given by solving the least-squares problem

$$\boldsymbol{\lambda} = \mathbf{H}^{\perp}\mathbf{f} \tag{10}$$

where $\mathbf{H}=H_{jk} = \phi\left(\left\|\mathbf{x}_i - \mathbf{x}_{c_k}\right\|; \theta_{c_k}\right)$ j =1, 2, ..., N, k =1, 2, ..., M, with N being the number of inputs and M the number of Gaussians used. The determination of optimal parameters, M, $\mathbf{x}_{ck}$ and θ_{ck} of an RBFANN is a subject of active research [7,40,42–44]. 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 multi-scale 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 $\mathbf{x}_{ck}$, *k =1, 2, ..., M*. The RBFs are initially uniformly distributed in the domain and are updated by a *K*-means clustering algorithm to avoid the possibility of an empty cluster in case of non-uniform inputs.

3. Determination of the shape parameter, θ_{ck}, 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

2.3.3 Kriging Family of Methods

The third class of metamodels studied is derived from Kriging methods, which have their origin in mining and geostatistical applications involving spatially and temporally correlated data[45]. 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 [46]. The resulting approximation interpolates the sampled input points. In a Kriging method, the estimated value of the function is given by

$$\tilde{f}(\mathbf{x_0}) = \sum_{l=0}^{r} \lambda_l p_l(\mathbf{x_0}) + Z(\mathbf{x_0}) \tag{11}$$

where $\tilde{f}(\mathbf{x_0})$ is the approximation to $f(\mathbf{x})$ at $\mathbf{x} = \mathbf{x_0}$, $p_l(\mathbf{x_0})$ denotes the *l*-th order term in a polynomial basis function of maximum order *r* and λ_l is the solution to the set of normal equations $f(\mathbf{x}_j) = \lambda_l p_l(\mathbf{x}_j)$, *j=1,2,…,N*. In Equation **(4)**, $Z(\mathbf{x_0})$ is a Gaussian random process with zero mean and a covariance structure $E[Z(\mathbf{x}_j)Z(\mathbf{x}_q)]$ *j,q* = 1,2,…,*N*, with the process variance σ^2. The correlation model R_{jq} of the process is of the form:

$$\mathbf{R} = R_{jq} = R(\theta, \mathbf{x}_j, \mathbf{x}_q) = \prod_{k=1}^{n} \gamma_k(\theta_k, d_k) \tag{12}$$

with a shape parameter θ where $d_k = x_{kj} - x_{kq}$, $k = 1,2,\ldots,n$, where n is the dimension of the vector **X**. The value of θ is selected by maximizing the following log-likelihood function of the model parameters

$$l = -\frac{N}{2}\ln\left[2\pi\sigma^2\right] - \frac{1}{2}\ln\left[|\mathbf{R}|\right] - \frac{1}{2\sigma^2}(\mathbf{f} - \mathbf{P}\boldsymbol{\lambda})^T \mathbf{R}^{-1}(\mathbf{f} - \mathbf{P}\boldsymbol{\lambda}) \tag{13}$$

where $\mathbf{f} = \lfloor f(\mathbf{x_1}) \quad f(\mathbf{x}_2) \quad \ldots \quad f(\mathbf{x}_N) \rfloor^T$, $\boldsymbol{\lambda} = \lfloor \lambda_0 \quad \lambda_1 \quad \ldots \quad \lambda_r \rfloor^T$ 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 [34].

An improved version of Kriging algorithm called the Dynamic Kriging Method with adaptive sampling (DKG) [41,47] has been implemented. The key aspects of the algorithm are summarized here.

1. The DACE code uses a modified Hooke and Jeeves algorithm [34] 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 [47].

2. The choice of the correlation model and the mean structure depends on the hypersurface to be approximated 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, 1and 2, using a Cross-Validation (CV) error estimate. The method also evaluates seven different correlation models, 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 or Latin Centroidal Voronoi Tessellation (LCVT) method. 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.

2.4 EVALUATION TECHNIQUES FOR METAMODELS

In order to be used for the generation of closure models in a multi-scale 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 reuse*: Since the metamodeling 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 metamodeling 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 meta-model filter noisy data to build a smooth approximation without adding excessive filtering errors.

2.5 ANALYSIS OF CONVERGENCE BEHAVIOR FOR METAMODELING TECHNIQUES

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 ×100uniformly 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 In order to quantify the accuracy of approximation at these points, a normalized sum-of-squares error is calculated:

$$\varepsilon_{L^2} = \sqrt{\frac{\sum_{j=1}^{S}\left(f\left(\mathbf{x}_j\right) - \tilde{f}\left(\mathbf{x}_j\right)\right)^2}{\sum_{j=1}^{S}\left(f\left(\mathbf{x}_j\right)\right)^2}} \tag{14}$$

where $f(\mathbf{x}_j)$ is the exact value of the function at the point $\mathbf{x}_j$ and $\tilde{f}\left(\mathbf{x}_j\right)$ is the value approximated by a metamodel.

4. *Local error estimation:* Since the error measure given by Equation (4) is a global measure of approximation errors, a normalized local error field, $\delta(x_j)$, defined as

$$\delta(\mathbf{x}_j) = \frac{\left| f(\mathbf{x}_j) - \tilde{f}(\mathbf{x}_j) \right|}{\frac{1}{S}\sum_{i=1}^{S} f(\mathbf{x}_j)} \tag{15}$$

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.

2.5.1 Harmonic Test Function

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

$$f(x, y) = \sin(2\pi x)\cos(4\pi y) + 2 \tag{16}$$

where x and y range from 0 to 1. As the convergence rates of the errors shown in Figure *2* demonstrate, 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 decreases as the number of input points increases for the RBFANN method. The system of equations defined by (10) 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 Figure *2*. 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 *1* .

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 1 : Correlation Parameters Estimated by the Kriging Method for approximating the function given by **(4)**

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 *2* when the rate of convergence of the Kriging method also changes. In comparison, the shape parameter estimated by the DKG method decreases uniformly (Table *1*) and the error in the DKG method also decreases uniformly (Figure *2*). 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 Figure *2,* 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 Figure *3*. 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 3). 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 are 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,

$$f(x, y) = \sin(2\pi x + 0.25)\cos(4\pi y + 0.5) + 2 \tag{17}$$

where *x* and *y* range from 0 to 1. The aliasing error seen in Figure *3* 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 3 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 *4* .

2.5.2 Hypersurface Constructed 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, LCVT method 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 multi-scale 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, 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 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, while the standard deviation of the error for the input distributions (for a fixed number of input points) is used to obtain *a prima facie* measure of the confidence interval. The average error is then compared to the error calculated from

the approximation of a structured grid of training points. *Figure 5* show the rate of convergence of the error for the RBFANN and Kriging methods respectively.

The convergence rates shown in *Figure 5* 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.

2.5.3 Reconstruction of a Hypersurface from a 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 (4) 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 (4). The contour plots for each of the reconstructed hypersurfaces are shown in Figure *6*

The hypersurface predicted using the RBFANN method is noise-free. Because an RBFANN filters out all frequencies beyond a certain limit, it can filter out the noise components most effectively. As seen in Figure *6*, 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 *6*, the noise from the training data is retained in the hypersurface predicted by these

methods. This section thus demonstrates that there may be cases of noisy input data which may not be satisfactorily reconstructed by the SC methods and the Kriging family of methods. Unless otherwise mentioned, the training data in the subsequent sections of the work is noise-free.

2.5.4 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,

$$f(x,y)=\begin{cases}\dfrac{C_1}{2\sqrt{0.4}}\ln\left|\dfrac{\sqrt{x^2+y^2}+\sqrt{0.4}}{\sqrt{x^2+y^2}-\sqrt{0.4}}\right| & \text{if } \sqrt{x^2+y^2}\le 0.3\\ \dfrac{C_2}{2\sqrt{0.2}}\ln\left|\dfrac{\sqrt{x^2+y^2}-\sqrt{0.2}}{\sqrt{x^2+y^2}+\sqrt{0.2}}\right| & \text{if } \sqrt{x^2+y^2}>0.3\end{cases} \tag{18}$$

where

$$C_1=\frac{16\sqrt{0.4}}{\ln\left|\dfrac{\sqrt{0.3}+\sqrt{0.4}}{\sqrt{0.3}-\sqrt{0.4}}\right|};C_2=\frac{16\sqrt{0.2}}{\ln\left|\dfrac{\sqrt{0.3}+\sqrt{0.2}}{\sqrt{0.3}-\sqrt{0.2}}\right|}$$

and x and y range from 0 to 1. This function is an integral of Eq.(65) from [38]. The contours of *f(x,y)=constant* are radially symmetric with respect to the origin. Steep gradients arise along the arc $x^2+y^2=0.3$; while the hypersurface varies more slowly as the distance from the arc increases. The exact hypersu rface is shown in Figure *10* (a).

Figure *7* shows the convergence of the errors of the metamodels with respect to the number of input points. In the convergence plot, 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 $x^2 + y^2 = 0.3$ decreases resulting in the corresponding higher local errors seen in Figure 9 (a) and (b).

Figure *6* 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 (>103), 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>103) is shown in Figure *8*.

2.6 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 multi-scale modeling framework.

2.6.1 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. [32], and is given by

$$C_D = \left(0.38 + \frac{24}{\mathrm{Re}} + \frac{4}{\mathrm{Re}^{1/2}}\right)\left(1 + e^{-\frac{0.43}{M^{4.67}}}\right) \tag{19}$$

This function includes the effects of particle Reynolds number, $Re=|vf-vp|dp/\nu$, and relative Mach number, $M=|vf-vp|/\_Tf$, in the drag coefficient equation. The model is limited to relative Mach numbers of $M \leq 1.2$and Reynolds numbers of $Re \leq 1 \times 10000$. However, for the present study, the model is considered to apply for $0.1 \leq M \leq 3$and $100 \leq Re \leq 10000$. The macro-scale EL code in [25] 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 $0.5<M<1.5$ and at $Re<200$ 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 *11*(a).

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 3. The sharpest transition region in the hypersurface in Figure *11* lies along the low *Re* 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 *11* 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. 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, 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 *14*). 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 *13* is that unlike the SC methods, the hypersurface obtained from the other metamodels is most erroneous 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*. 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.

2.6.2 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.*[33]*. This model also corrects for high particle Mach and Reynolds numbers but over a wider range, Re≤1 $\times 10^5$ and M≤5,

$$C_D = \frac{24}{\mathrm{Re}}\left(1 + 0.25\,\mathrm{Re}^{0.687}\right)\left(1 - \frac{0.258C}{1+514G}\right) + \frac{0.42C}{1 + 4.25\times 10^4\,\mathrm{Re}^{-1.16}\,G} \tag{20}$$

where C and G are defined as

$$C = \begin{cases} \frac{5}{3}\tanh\left(3\ln\left(Ma + 0.1\right)\right) \text{if Ma} \le 1.45 \\ 2.044 + 0.2\mathrm{e}^{-1.8\left(\log\left(\frac{\mathrm{Ma}}{2}\right)\right)^2} \text{ if Ma} > 1.45 \end{cases} \tag{21}$$

$$G = \begin{cases} 1 - 1.525Ma^4 \text{ if Ma} < 0.89 \\ 0.0002 + 0.0008\tanh\left(12.77\left(\mathrm{Ma} - 2.02\right)\right) \text{if M} > 0.89 \end{cases} \tag{22}$$

The function produces very steep gradients arranged in a series of steps in the interior of the domain. The contour of the hypersurface of (18) is shown in Figure *14* (a).

As shown in Figure *15*, 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 Boiko, many of the complex characteristics of Loth's model lie in the interior of the domain as can be seen in Figure *14*. But from Figure 3, 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 *16*. 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 *14*. The oscillations in the DKG method is more localized than the Kriging method Figure *16*. To investigate this, the correlation 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 using a General Exponential correlation 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 *16*. 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 *16*.

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

Tong et. al. *[48]* have extended Loth's model to include variations of particle shape and particle volume fraction, α,

$$C_D = C_{da} + 0.5048\alpha\left(1+\frac{34.8}{\mathrm{Re}^{0.5707}}\right)^4 + 0.9858\alpha\left(1+\frac{34.8}{\mathrm{Re}^{0.5707}}\right) \tag{23}$$

where C_{da} is the drag coefficient calculated using Equation (22). This model illustrates the complex dependence of the particle drag coefficient on many different parameters, including the Knudsen number (set equal to 10 here), *Ma* (set to1 in here), *Re* (varied) and α (varied). The hypersurface is shown in Figure 17 (a)

The convergence of the metamodels with respect to the number of input points is shown in Figure 18. The PSC method and the DKG method approximate the hypersurface most accurately.

Despite the higher concentration of input nodes in the low *Re*, 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 17, oscillations arise when the RBFANN method is used to build the approximation. Similar to the case of the harmonic function in Section4, 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 2 . 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 θ=20past 400, where a non-monotonic jump is seen in the convergence plot in Figure 18. 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 (see also Figure 19).

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	9.0073

Table 2 : Correlation parameters estimated by the Kriging method and the DKG method to approximate the hypersurface given by **(4)**

2.7 CONCLUSION

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 and the local errors.

For a large number of training points, the SC methods generally approximate most of the hypersurfaces most accurately. 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 meso-scale 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 Section4.3. 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.

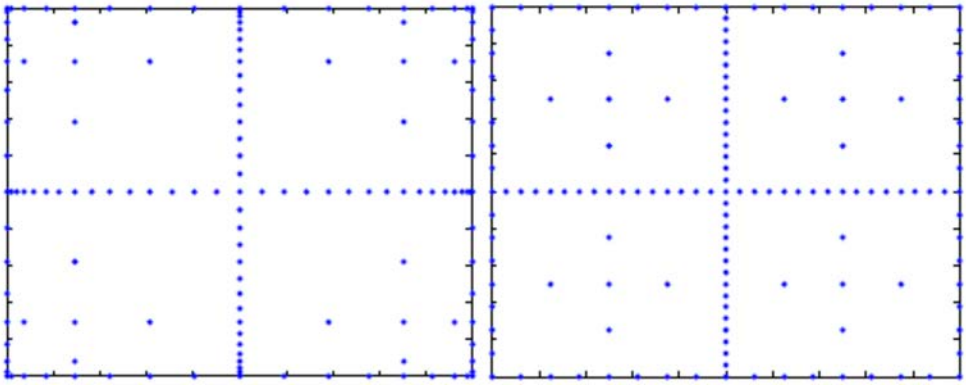


Figure 1 : The grid in an SC method using (a) the zeroes of Chebyshev Polynomial (b) Newton-Cotes quadrature.

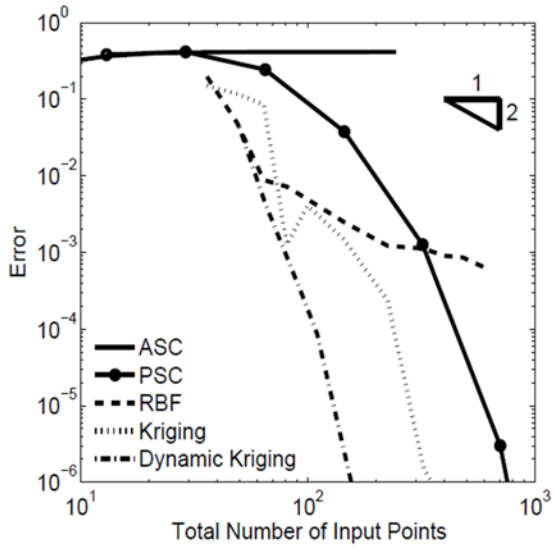


Figure 2 : Error plot showing the convergence rates on approximating the smooth harmonic test function.

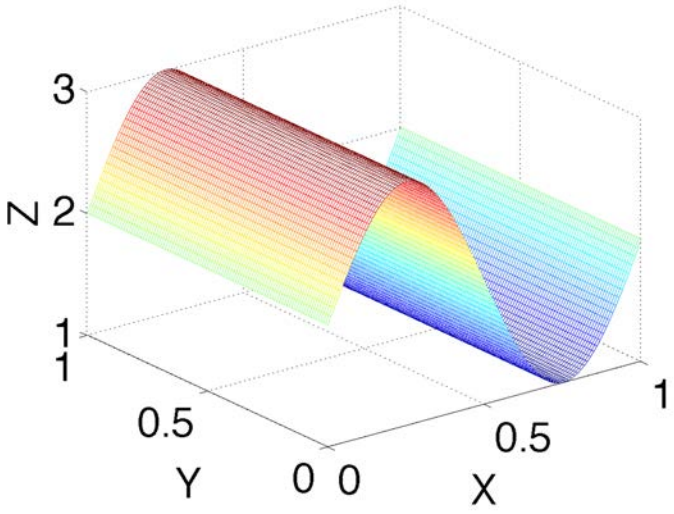


Figure 3 : The Hypersurface of the function **(4)** as approximated by the ASC method.

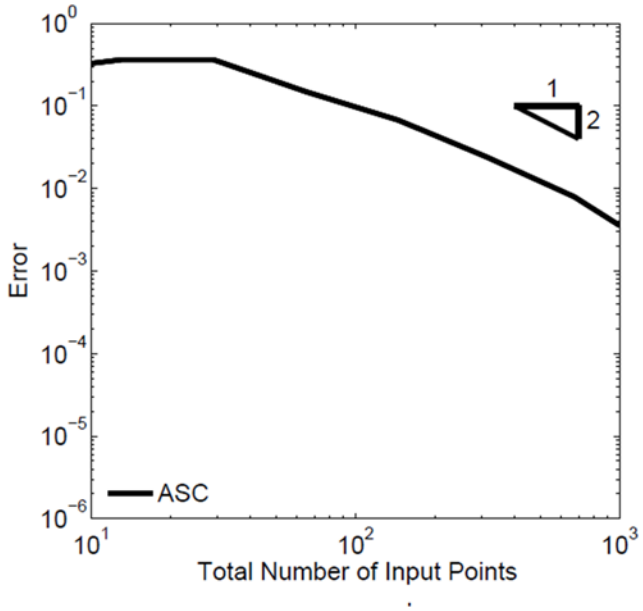


Figure 4 : Error Convergence of the approximation of the Shifted Harmonic Test Function by the ASC Method.

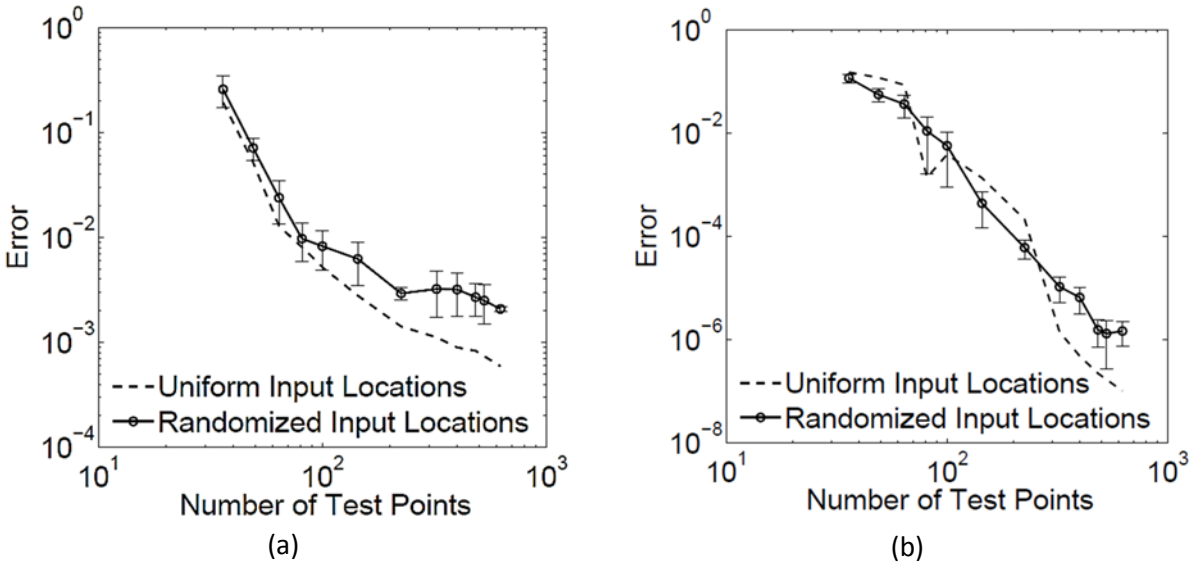


Figure 5 : Error plot showing the convergence rates on interpolating the harmonic test function from using (a) an RBF method and (b) a Kriging method. The error bars show the standard deviation from the mean error for randomly distributed input points.

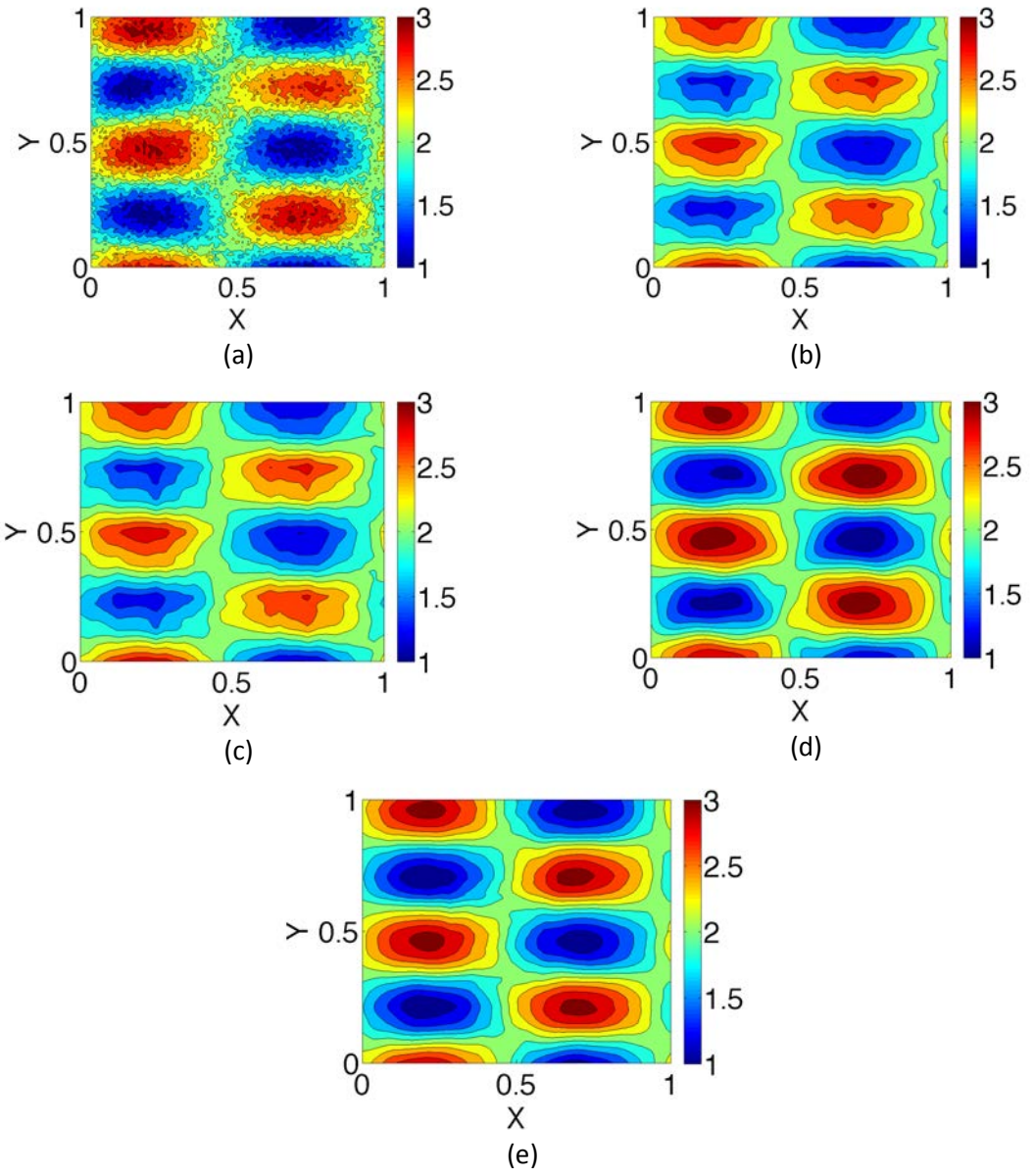


Figure 6 : Representative Hypersurfaces for the noisy shifted harmonic function using (a) a PSC method, (b) an ASC method, (c) an RBF ANN, (d) a Kriging and (e) a DKG method.

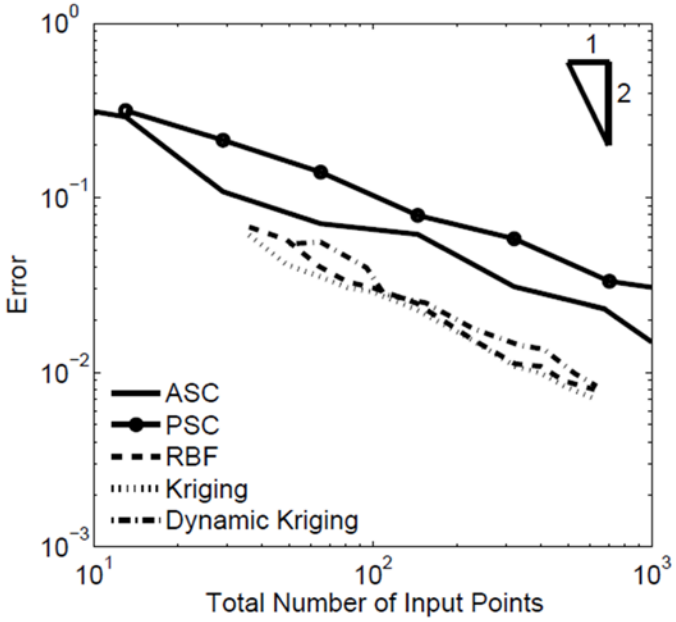


Figure 7 : Error plot showing the convergence rates on approximating the radially symmetric test function using a PSC method, an ASC method, an RBFANN, a Kriging method and the DKG method.

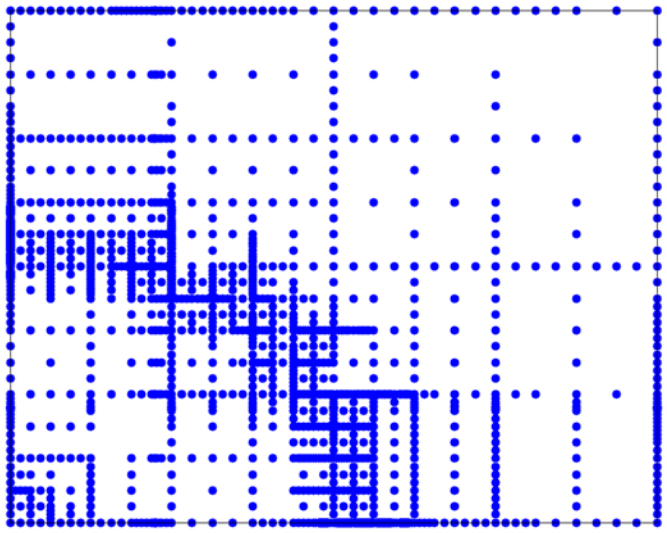


Figure 8 : Locations of nodes in a 12th level Smolyak sparse grid, refined adaptively using hierarchical surpluses with a maximum error of ε=0.01, interpolating the function.

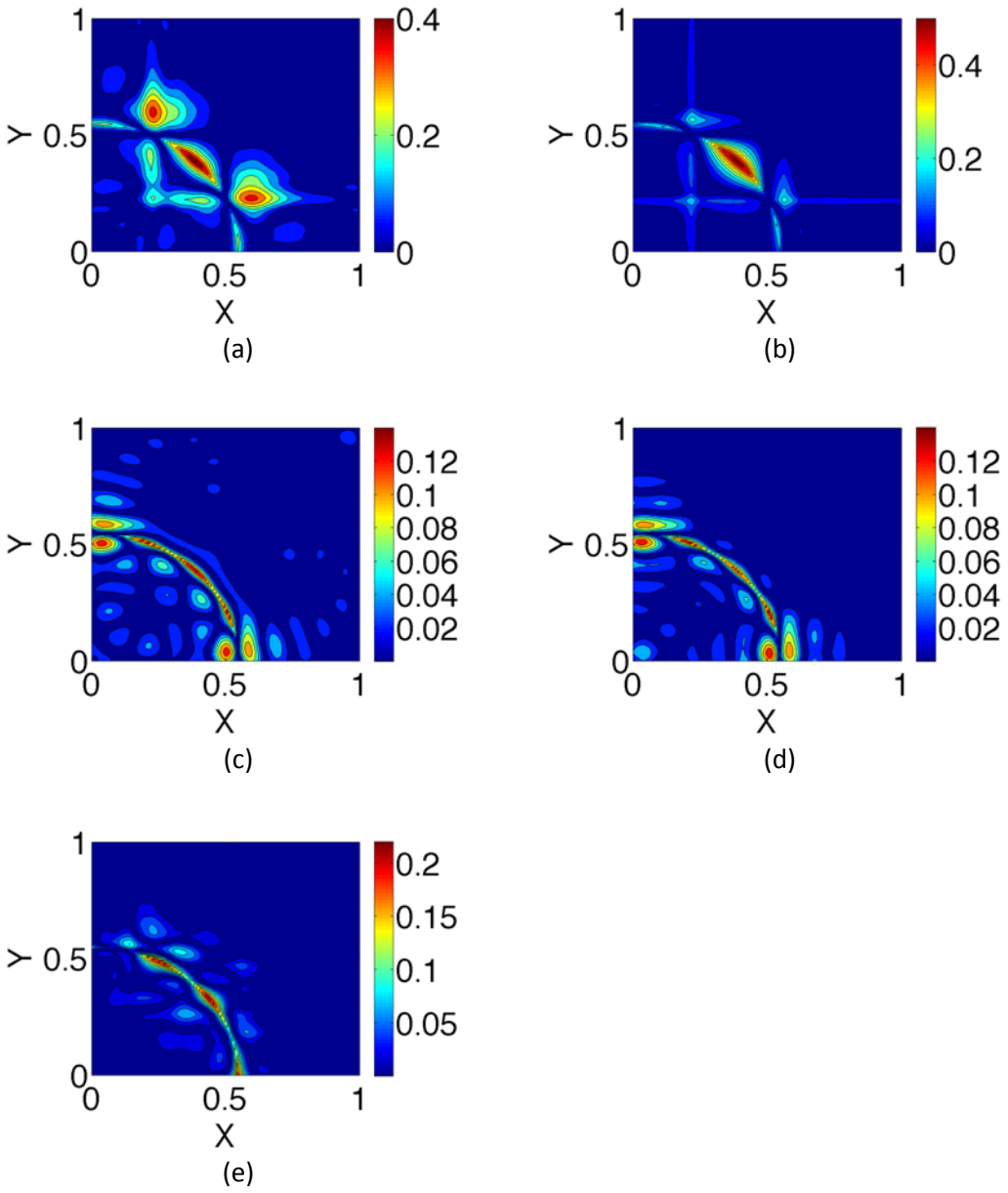


Figure 9 : Local error plots for the approximation of the radially symmetric test function using (a) a PSC method, (b) an ASC method, (c) an RBF ANN, (d) a Kriging method, and (e) the DKG method.

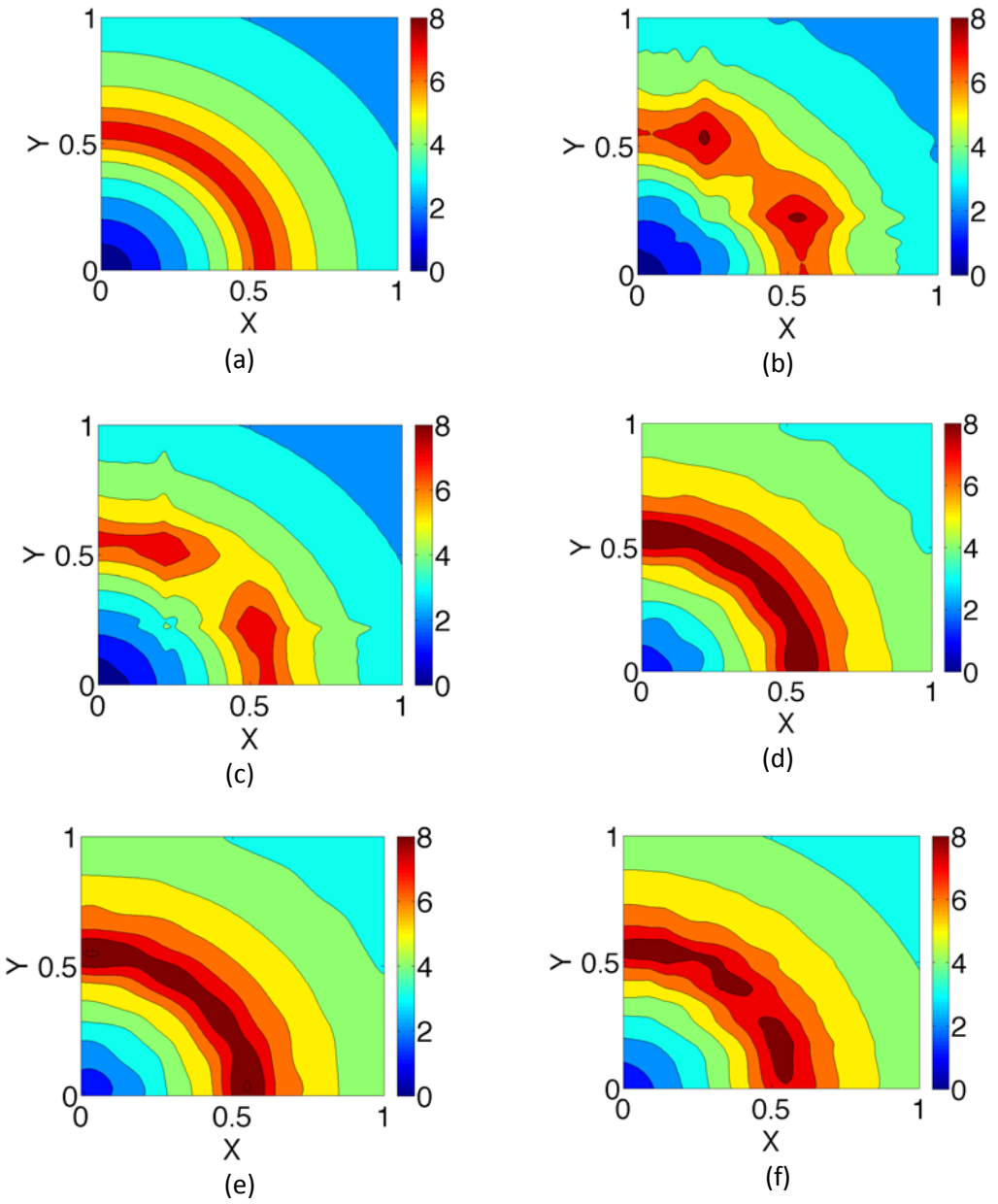


Figure 10 : The (a) exact contours and representative contour plots for the test function from(14)using (b)a PSC method, (c)an ASC method, (d)an RBF ANN, (e)a Kriging method and (f)the DKG method.

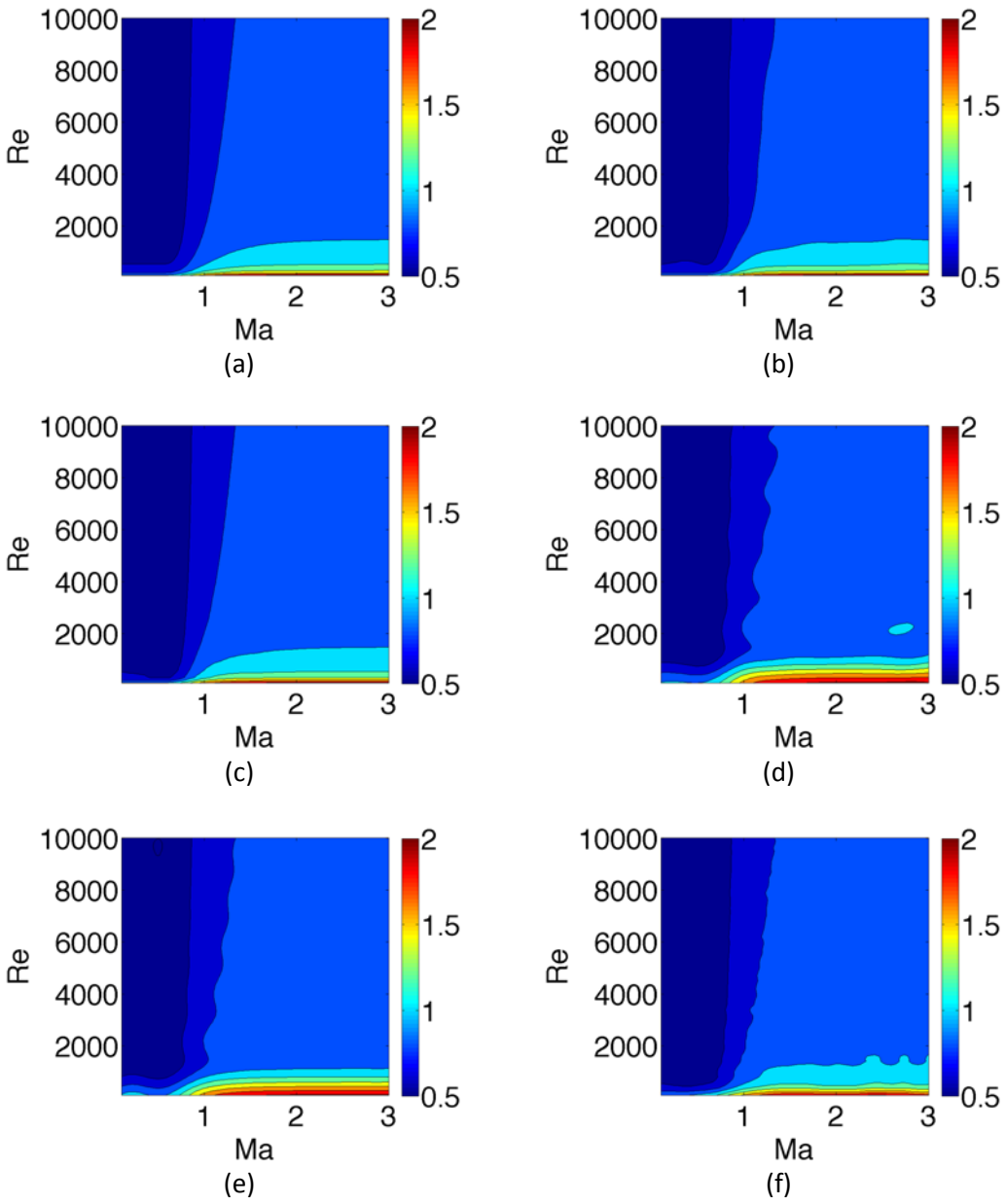


Figure 11 : The (a) exact contours and representative contour plots for the drag coefficient of [32] using (b)a PSC method, (c)an ASC method, (d)an RBF ANN, (e)a Kriging method and (f)the DKG method.

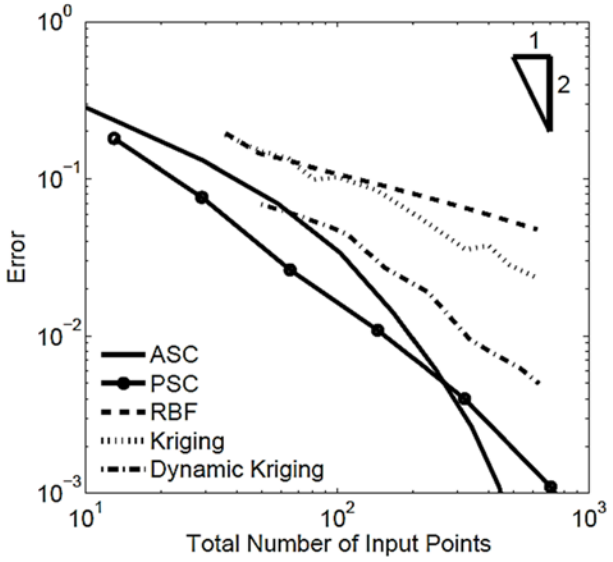


Figure 12 **:** Error plot showing the convergence rates on approximating the drag model of Boiko et.al. [32]] using a PSC method, an ASC method, an RBFANN, a Kriging method and the DKG method.

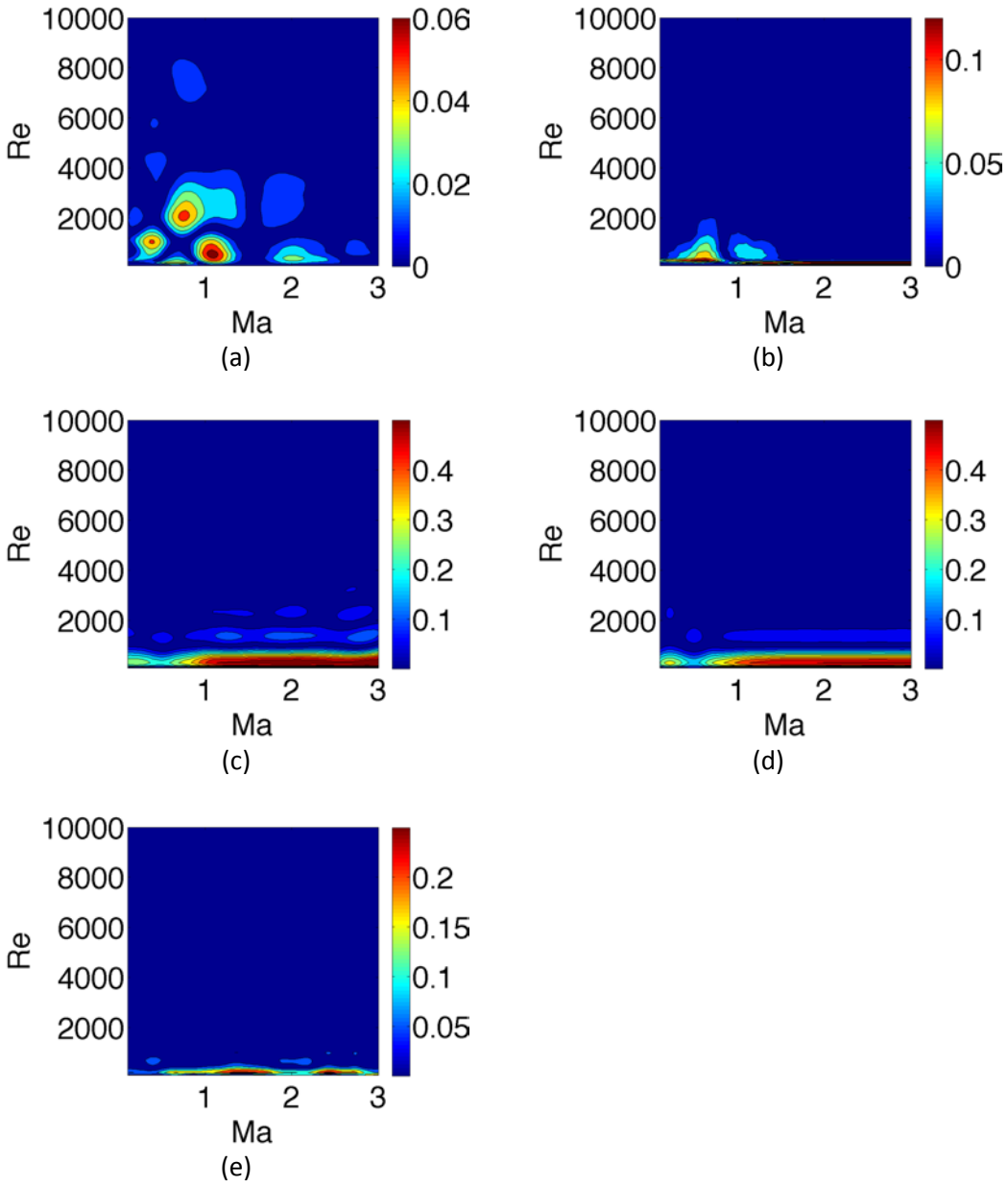


Figure 13 : Local error plots for the approximation of drag model of Boiko et al.[32]]using (a)a PSC method, (b)an ASC method, (c)an RBF ANN, (d)a Kriging method and (e)the DKG method.

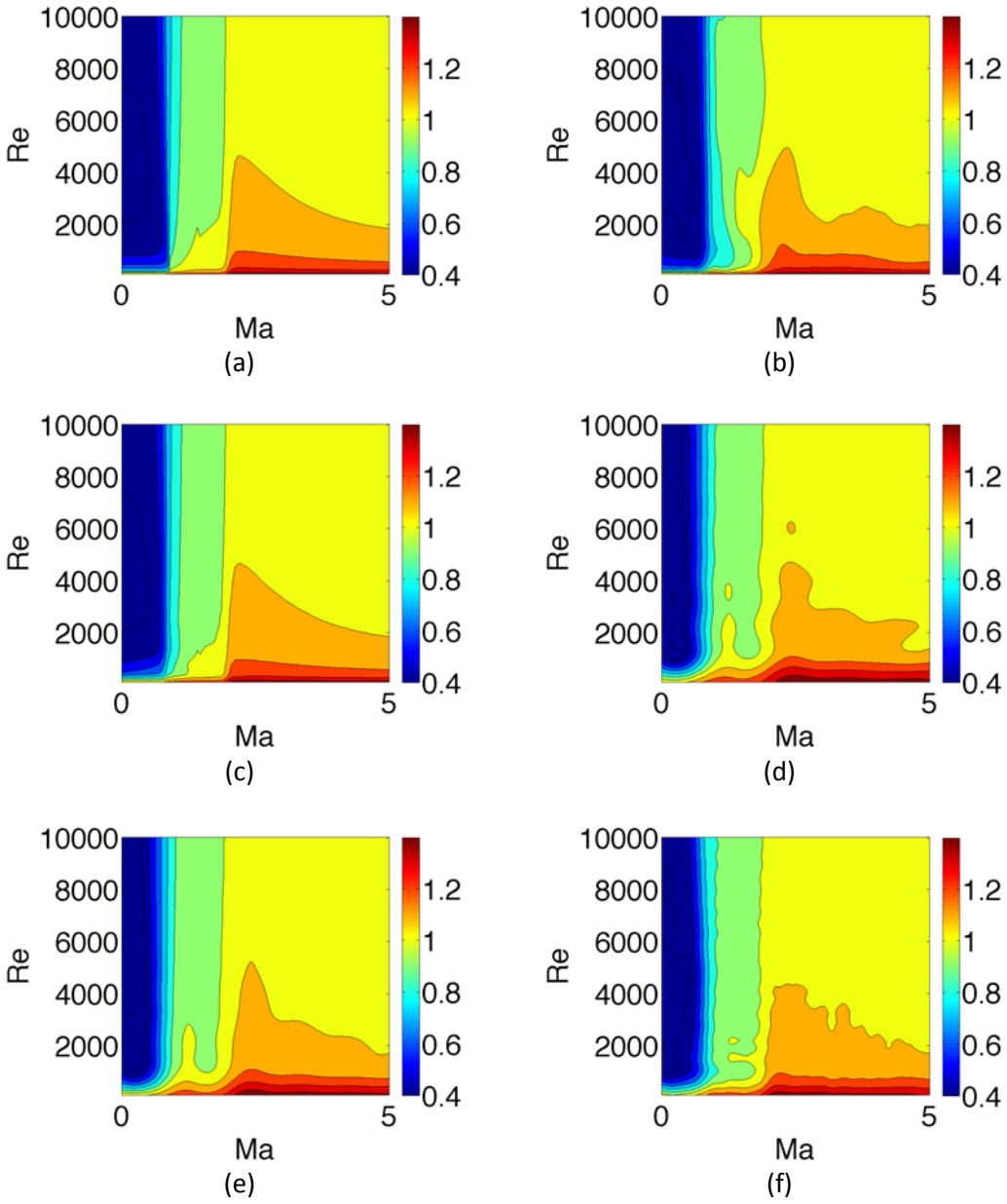


Figure 14 : The (a) exact contours and representative contour plots for the drag coefficient of [33] using (b)a PSC method, (c)an ASC method, (d)an RBFANN method, (e)a Kriging method and (f) the DKG method.

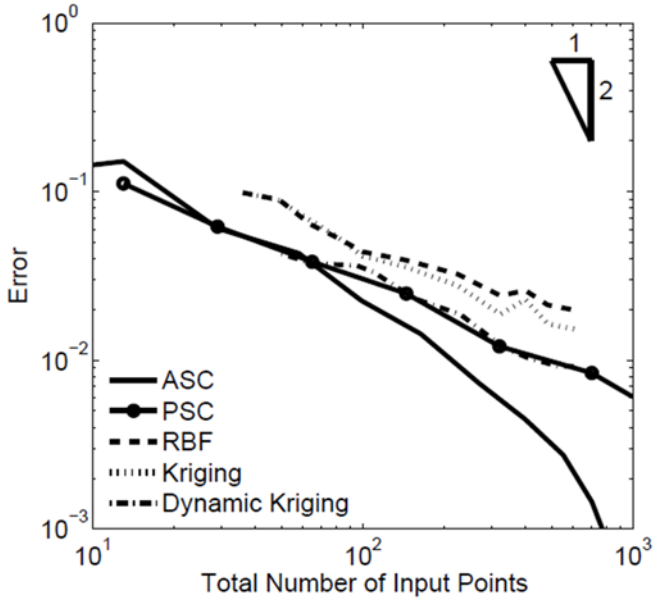


Figure 15 : Error plot showing the convergence rates on approximating the drag model of Loth [33] using a PSC method, an ASC method, an RBFANN, a Kriging method and the DKG method.

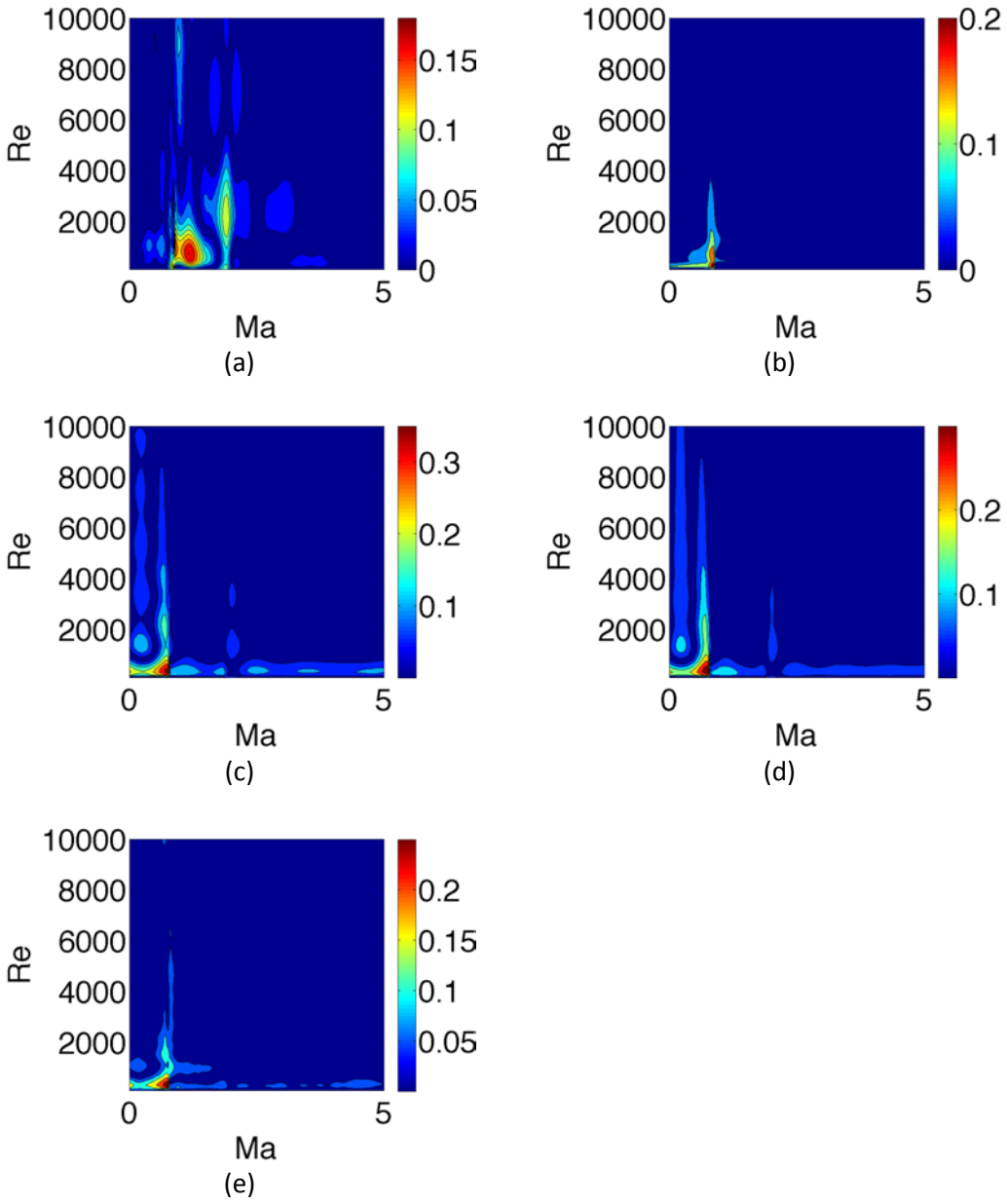


Figure 16 : Local error plots for the approximation of drag model of Loth using (a) a PSC method, (b) an ASC method, (c) an RBFANN method, (d) a Kriging method and (e) the DKG method.

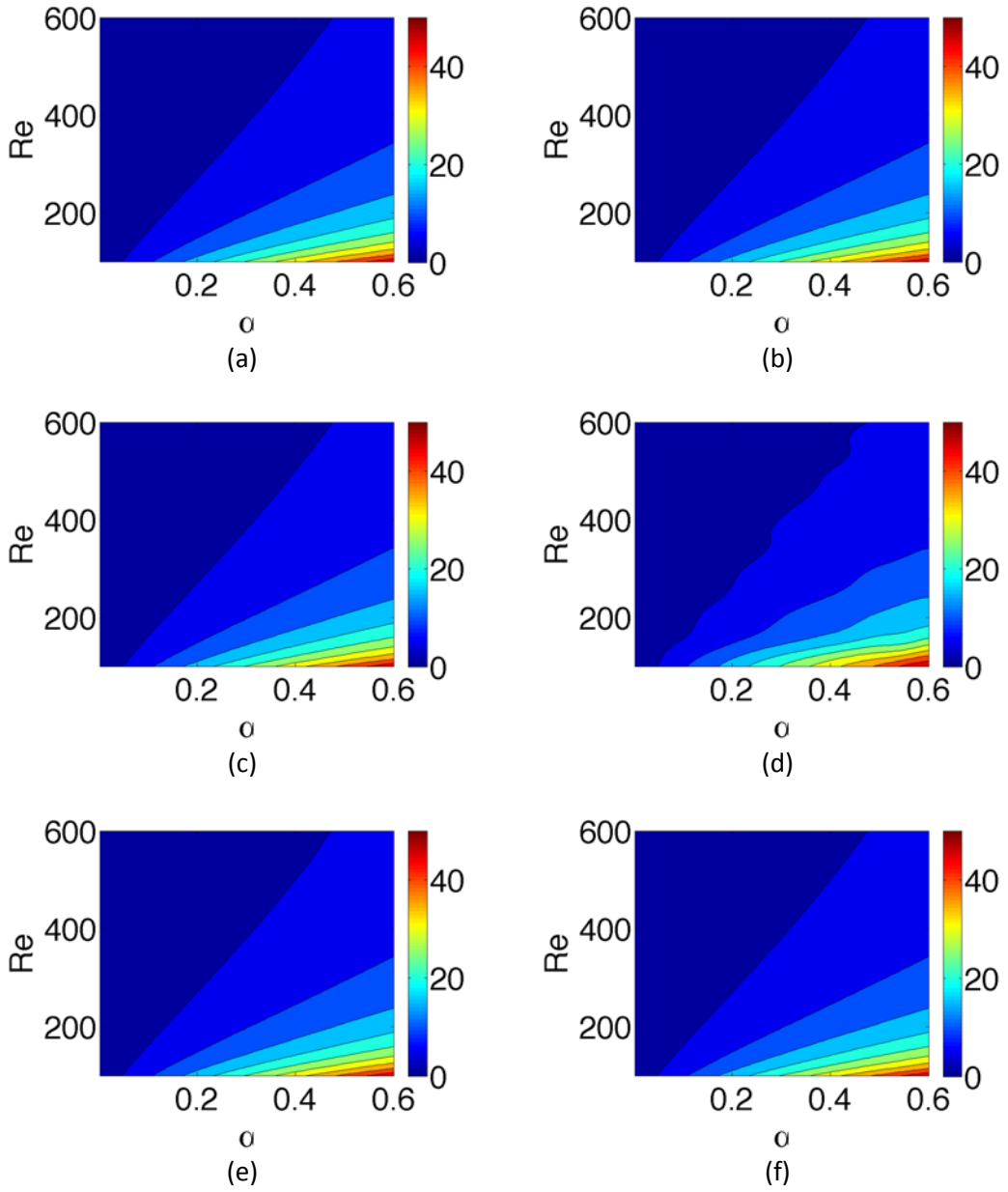


Figure 17 : The (a) exact contours and representative contour plots for the drag coefficient of Tong using (b) a PSC method, (c) an ASC method, (d) an RBFANN, (e) a Kriging method and (f) the DKG method.

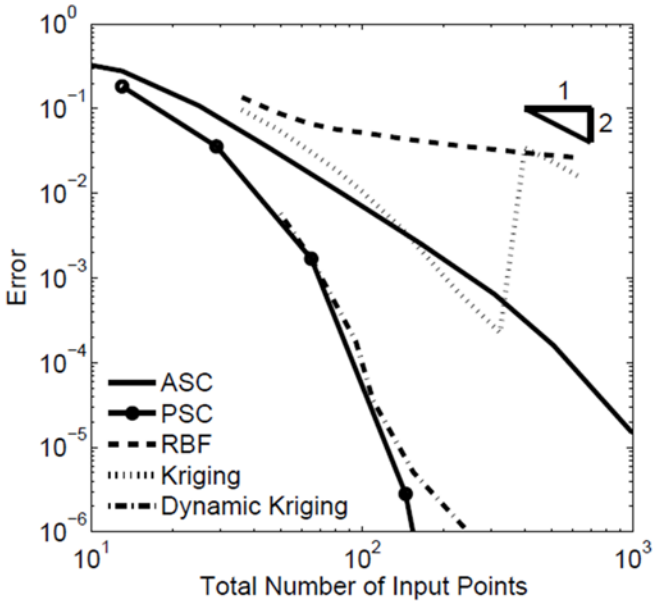


Figure 18 : Error plot showing the convergence rates on approximating the drag model of Tong et al. using a PSC method, an ASC method, an RBFANN, a Kriging method and the DKG method.

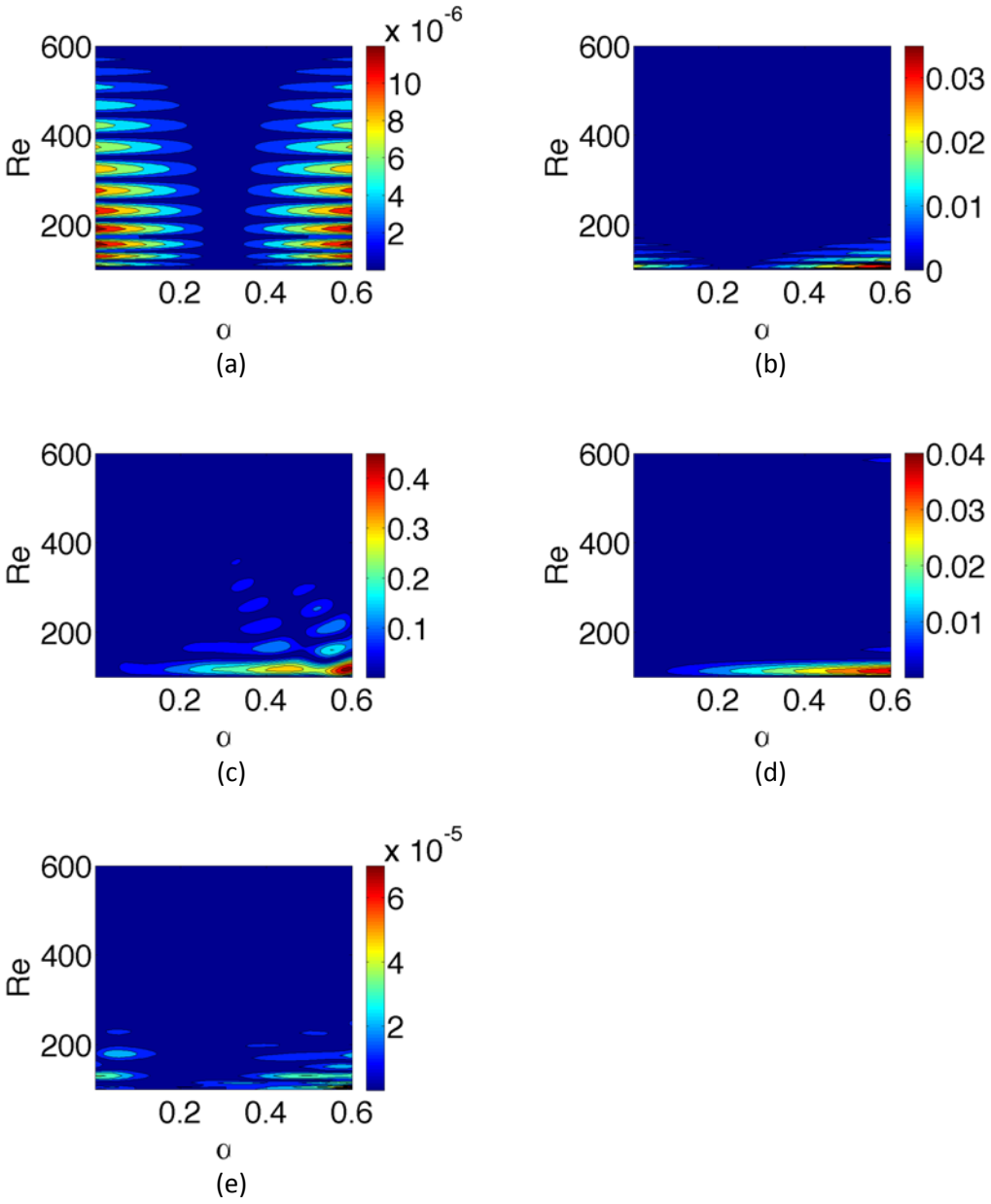


Figure 19 : Local error plots for the approximation of drag model of Tong at. al. [48] using (a) a PSC method, (b)an ASC method, (c) an RBF ANN, (d) a Kriging method and (e)the DKG method.

CHAPTER 3. EVALUATION OF KRIGING BASED SURROGATE MODELS CONSTRUCTED FROM MESOSCALE COMPUTATIONS OF SHOCK INTERACTION WITH PARTICLES

3.1 INTRODUCTION

3.1.1 Motivation and Relevance

Multiscale problems with distinct meso- and macroscales appear in several important engineering applications, e.g. the dynamics of particle-laden gases [1,2], deformation of heterogeneous materials such as bones [3–5], concrete [6,7], heterogeneous explosives[8–11], sediment transport in river beds [12], and meso-scale models of blood flow[13]. In such problems, computational approaches can typically resolve only a limited range of length and time scales. The unresolved or subgrid scales are coupled to the macroscale using a variety of multiscale modeling frameworks such as patch-dynamics [14,15], heterogeneous multiscale modeling [16–19], multigrid methods [20,21], wavelet-based methods [22–24] and others [25]. In most multiscale modeling frameworks, the subgrid scale physics is modeled using "closure laws" in homogenized (volume averaged/ filtered/coarse-grained) macroscale systems of equations. Traditionally, such closure laws were obtained from physical experiments, canonical theoretical constructs or phenomenological arguments. With increasing computational power and improved physical models and algorithms, it is now possible to derive closure laws from highly resolved mesoscale simulations. This work addresses the issue of efficient and accurate closure model construction from highly resolved meso-scale simulations in the context of shocked particle-laden flows. The techniques examined in this paper can be applied generally to multi-scale modeling problems where there is a distinct separation of scales and the macro-scale and meso-scale can be computed using continuum thermo-mechanical descriptions.

3.1.2 Closure Models for Bridging Scales in Multi-scale Problems

A generalized framework for solving multiscale problems is the heterogeneous multiscale modeling (HMM) approach [16–19]. In HMM, the macroscale variables are defined by the vector **A** and satisfy the system of equations given by [19]

$$\mathbf{F}(\mathbf{A}, \mathbf{D}) = \mathbf{0} \tag{24}$$

where **F** is the macroscopic operator. **D** is the information at the macroscale, which depends on the mesoscale interactions and needs to be provided to close Equation (1). Once **D** is known, **A** can be determined using Equation (1)

The mesoscale state variable **a** is a solution of the system of equations given by [19]

$$\mathbf{f}(\mathbf{a})\big|_{\mathbf{b}} = \mathbf{0} \tag{25}$$

where **f** is a mesoscale operator and **0** is the null-vector. In the mesoscale system of equations, **b** represents the constraints (e.g. boundary/initial conditions) on the mesoscale model; it is typically prescribed by local macroscale dynamics. The HMM framework allows two-way coupling between the meso and the macroscales, i.e., the macroscale supplies the constraints **b** to the mesoscale model and the mesoscale returns the missing information, **D** to the macroscale model in form of a closure law. A closure law is a model of the form

$$\mathbf{D} = D(\mathbf{b}) \tag{26}$$

where the constraints ***b*** define the parameter space for **D**. Closures can be obtained through physical experiments which are expensive and may only cover limited regions in parameter space. An alternative, pursued in this and other works [3–5,7,30–35] is to construct **D** using resolved mesoscale simulations. Theoretically, large ensembles of meso-scale simulations can be used to collect quantitative information that can be used to construct closure laws. In practice such simulations are also limited by computing resources and fidelity of models. Therefore construction of closure from numerical experiments must contend with sparse datasets.

To compute **D** from mesoscale computations, either a concurrent coupling or a sequential coupling strategy for connecting the scales can be used. In a concurrent strategy [19, 26,36] the macroscale computations are performed for one time step at all the spatial computational sub-volumes of the discretized macroscale model. The macroscale field then supplies boundary conditions to mesoscale computations. Mesoscale computations are performed — perhaps in selected sub-volumes — to compute the closure terms. The macroscale model uses these closure terms to advance the computation to the next time level. A bottleneck in this approach is that the highly resolved mesoscale computations are expensive. It is generally not possible to perform mesoscale simulations within each macroscale control volume for each time step. Alternatively, in a sequential strategy [7,30] mesoscale computations are performed for a range of anticipated flow conditions *a prio*ri. A numerical model for the closure laws are constructed from these mesoscale experiments. The simulation-based closure model, also known as a metamodel or a "model of a model" [36–38] serves as a surrogate for the meso-scale (subgrid or unresolved) dynamics in the macroscale computations. The macroscale computations utilize the numerical closure models at each time step in each sub-volume of the macroscale domain.

3.1.3 Construction of Surrogate Models for Shocked Particle-laden Flows

For the particular problem of shocked gas-particle flows[1, 31], the number of particles in the flow is so large that macroscale models cannot explicitly track the dynamics of each such particle. Macroscale computations are performed using Eulerian-Lagrangian [40] and Eulerian-Eulerian [41] approaches. In the Eulerian-Lagrangian (E-L) method, the fluid phase is computed in the Eulerian frame while particles are tracked in Lagrangian fashion. In the interest of computational tractability, particularly for large ensembles of particles, E-L methods typically track "computational" particles: pseudo-particles that are agglomerates of a large number of physical particles ([35,40,42]).The computational particles are modeled as singular point sources which couple with the carrier fluid through momentum exchange modeled via source terms (*D)* in the macroscopic fluid equations(Figure 20a). These exchange terms depend on the local macroscopic conditions, e.g. shock strength and particle loading, which define the parameter space for obtaining the closure law in the form given by Equation (3).

There are several semi-empirical drag laws available to close the macroscale gas-particle momentum equations [43–49]. These drag laws are developed via experiments performed in a

limited parameter space and extrapolated to other regimes. In previous work [33,35,43,50] closure models were constructed and used in the multiscale framework for gas-solid flows. In [43] an Artificial Neural Network (a multilayer perceptron) was used to construct the surrogate model. In [50] it was shown that depending on the surrogate model to be constructed, ANN-based techniques require an unduly large number of mesoscale simulations to construct surrogates. The burden of computational cost in surrogate based multiscale modeling approaches lie in: (a) the number of mesoscale computations needed to construct a reasonably accurate surrogate model and (b) construction of the surrogate model from the computed data. Of these two, for typical metamodeling techniques, the computational cost of the latter is negligible compared to the cost of performing computationally intensive mesoscale simulations. Recent work [50] showed that Kriging based methods provide accurate, efficient approaches for constructing surrogate models from modest numbers of meso-scale simulations.

3.1.4 Focus and Novelty of this Work

Developing metamodeling techniques that provide robust surrogates using sparse data sets is an active field of research [38,51–53]. In previous work [50], the performance of several candidate metamodeling techniques was analyzed. This was done for several canonical drag laws in the particular context of shocked particle laden gases. Among the different metamodeling techniques tested, viz. stochastic collocation techniques [54,55], radial basis functions [56–60] and the Kriging family of methods, the Dynamic Kriging Method [36,37] (DKG) emerged as the best approach. DKG was shown to converge monotonically for several model closure laws even when the number of input points were low. However, in the previous analysis, the study of the convergence of metamodeling techniques was restricted to analytical functions, including canonical drag laws. The current work extends the effort in [50]. Instead of analytical functions, the construction of surrogate models is carried out from resolved mesoscale computations using state-of-the-art [36] Kriging-based methods. But Kriging methods that rely on interpolation, including DKG are known to experience difficulties with noisy data [50,61,62]. Because high frequency noise in the input data is inherent to numerical experiments, a fitting method called the Modified Bayesian Kriging Method (MBKG) [61,62] is evaluated alongside the DKG method to construct surrogate models from numerical experiments.

This is the first attempt, to the authors' knowledge, at constructing surrogate models for closure using the DKG and MBKG methods. In this paper, the relative merits and weaknesses of the two methods are analyzed. Furthermore, while metamodeling in the presence of noise in the input data has been studied previously, a large number of the studies assume particular nature of the noise [50,61]. In the current work, no prior assumption is made about the presence of noise in the numerically calculated input data. This work is a key step towards a full multiscale modeling of particle-laden flows using surrogate based closure laws obtained from mesoscale computations.

3.1.5 Outline of the Paper

The paper is organized as follows. In Section 2, the governing equations for the mesoscale computational model and the equations for the Kriging methods are outlined. To obtain surrogate models from mesoscale computations numerical experiments are performed. The description of the mesoscale computational set up, along with the algorithm for creating surrogates using the DKG and the MBKG methods is presented in Section 6. The mesoscale computational algorithm and Kriging methods are validated in Section 7.1. Section 7.2 describes the results of numerical experiments. The surrogates created from mesoscale computations using the Kriging methods are shown in Sections 7.3. The rate of convergence of the surrogates generated with respect to the number of inputs using the DKG methods and the MBKG methods are compared and key observations are presented in Section 7.3.3.

3.2 COMPUTATIONAL MODELS

There are two ingredients required to construct surrogate models from numerical experiments:

1. A mesoscale computational framework to compute the drag around particles from resolved numerical computations.
2. A metamodeling technique to construct a numerical closure law for the drag in the parameter space (defined by macro-scale conditions).

In Section 3.2.1 , the computational framework for performing high-fidelity numerical experiments is described, while in Section 3.2.2 , techniques for constructing surrogate models from numerical experiments are outlined.

3.2.1 The Mesoscale Model

3.2.1. 1 The Governing Equations

In the mesoscale computations, the gas flow is modeled by the compressible Euler equations:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_i)}{\partial x_i} &= 0 \\ \frac{\partial (\rho u_i)}{\partial t} + \frac{\partial (\rho u_i u_j + p\delta_{ij})}{\partial x_j} &= 0 \\ \frac{\partial (\rho E)}{\partial t} + \frac{\partial (\rho u_i E + p u_i)}{\partial x_i} &= 0 \end{aligned} \tag{27}$$

where ρ, u_i, p are the density, velocity components and the pressure of the fluid respectively, while $E = e + 1/2 u_i u_i$ and e are the specific total internal energy and the specific internal energy of the fluid. The equations are closed by the ideal gas equation of state given by

$$p = \rho e(\gamma - 1) \tag{28}$$

where the specific heat ratio $\gamma = 1.4$. A well tested Eulerian code SCIMITAR3D [63–69] is used to solve Equations (27) and (28) and is described in the next section.

3.2.1. 2 Numerical Framework

The governing equations are spatially discretized on a fixed Cartesian mesh using a 3rd order Essentially Non-Oscillatory (ENO) [70] scheme and are marched in time explicitly using a 3rd order Runge-Kutta (RK) scheme. The procedure for tracking the interfaces of the solid particles and the application of the boundary conditions at the interface is summarized as follows.

1. To define embedded objects in the flow a narrow-band level-set [71] method is used; this allows tracking the object interfaces in a sharp manner. The level set field, ϕ_l at any point is the signed normal distance from the l^{th} immersed object with $\phi_l < 0$ inside the

immersed object and $\phi_l > 0$ outside. The interface is implicitly determined by the zero level set fields, i.e. $_l = 0$ contour represents the l^{th} immersed boundary.

2. Once the object interfaces are identified by $\phi_l = 0$ contour levels, the no-penetration boundary conditions are applied using a variant of the ghost fluid method (GFM) [72]. GFM relies on the definition of a band of ghost points corresponding to each phase of the interacting materials. The ghost points for the fluid phase are the points lying inside the embedded object (defined by $\phi_l > 0$) and real points are those which are outside.
3. Once these ghost points are identified, the next step is to populate the ghost field for the fluid. The ghost field is obtained by constructing least-squares interpolation of the field variables of real material points and by imposing the appropriate interfacial conditions [68] Once the ghost points are identified and the values of the flow variables (e.g. ρ, u_i, p and E) at these points are populated with the least-squares field, the two-material problem can be converted to two single-material problems consisting of real fields and their corresponding ghost fields.

   The above approach, discussed in detail in [63–69], is used to compute the flow variables and the drag on the particles in the flow field. Once the drag is calculated, surrogate models are constructed using the metamodeling techniques described below.

3.2.2 Metamodeling Techniques:

Metamodeling involves the estimation of an unknown function $f(\mathbf{x})$ which is known only at certain discrete and distinct points $\mathbf{x}_j$ (j=1,2,…,N). The points $\mathbf{x}_j$ (j=1,2,…,N) where the values of the function are known span the parameter space of the surrogate model. The set of the known values of the function and their locations, i.e. the set ($\mathbf{x}_j$, $f(\boldsymbol{x}_j)$) are known as the *inputs* of the surrogate model and are used to construct the unknown function $f(\mathbf{x})$ in the parameter space. Often, it is of interest to obtain the value of the function at selected locations in the parameter space, say $\boldsymbol{x}_0$. The point $\boldsymbol{x}_0$ is called the "*probe point*" of the surrogate model, while the value of the function at the probe points, $f(\boldsymbol{x}_0)$ is called the *output* of the surrogate model.

3.2.2. 1 The Kriging Methods

Kriging methods for constructing surrogates combine a global (polynomial) model which fits to the given input points and a semi-variogram model which represent the local departures from the

global fit [73]. The output of the surrogate model is the function, denoted by $\tilde{f}(\mathbf{x}_0)$ and expressed as [36,74]

$$\tilde{f}(\mathbf{x_0}) = \sum_{l=0}^{r} \lambda_l p_l(\mathbf{x_0}) + Z(\mathbf{x_0}) \qquad (29)$$

where $\tilde{f}(\mathbf{x}_0)$ is the approximation to $f(\mathbf{x})$ at $\boldsymbol{x} = \boldsymbol{x_0}$, $p_l(\boldsymbol{x}_0)$ denotes the l-th order term in a polynomial basis function 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; $\tilde{f}(\mathbf{x}_j)$ represents the overall trend or the mean structure [75] of the input points in the parameter space. The specific choices of r = 0 or r = 1 are called Blind Kriging or Universal Kriging, respectively [37]. In general, while constructing a Kriging model, the mean structure (i.e. the value of r) needs to be specified *a priori*.

The second term in Equation (29), $Z(\boldsymbol{x}_0)$ represents the correlation of nearby points with each other and is modeled as a Gaussian random process with zero mean and a covariance structure $E[Z(\mathbf{x}_j)Z(\mathbf{x}_q)] = \sigma^2 R_{jq}$ j,q = 1,2,…,N, where σ^2 is the process variance. The correlation model R_{jq} is of the form:

$$\mathbf{R} = R_{jq} = R(\theta, \mathbf{x}_j, \mathbf{x}_q) = \prod_{k=1}^{n} \gamma_k(\theta_k, d_k) \qquad (30)$$

with a shape parameter θ, where $d_k = (x_{kj} - x_{kq})$, k = 1,2,…,n, n being the dimension of the vector **x**. Commonly used models of the correlation functions are listed in Table 3 [76]. Similar to the mean structure, construction of a Kriging approximation assumes a known covariance model specified *a priori*. Once a correlation function is specified, the value of θ may be determined by maximizing the following log-likelihood function of the model parameters [37]

$$l = -\frac{N}{2}\ln\left[2\pi\sigma^2\right] - \frac{1}{2}\ln\left[\left|\mathbf{R}\right|\right] - \frac{1}{2\sigma^2}(\mathbf{f}-\mathbf{P}\boldsymbol{\lambda})^T\mathbf{R}^{-1}(\mathbf{f}-\mathbf{P}\boldsymbol{\lambda}) \quad (31)$$

where $= \lfloor f(\boldsymbol{x}_1) \quad f(\boldsymbol{x}_2) \quad ... \quad f(\boldsymbol{x}_N)\rfloor$, $\boldsymbol{\lambda} = \lfloor \lambda_0 \quad \lambda_1 \quad ... \quad \lambda_T \rfloor^T$ and $\mathbf{P} = P_{jl} = p_l(\boldsymbol{x}_j)$. For a given mean structure and correlation function, Kriging provides an interpolation through the input points, i.e. $\tilde{f}(\mathbf{x}_j) = f(\mathbf{x}_j), \forall \mathbf{x}_j, j = 1,2,...,N.$ In addition to estimating $\tilde{f}(\mathbf{x}_0)$, the predicted variance at a point $\mathbf{x}_0$, $\sigma_p{}^2(\mathbf{x}_0)$ can also be computed using the following equation [36]

$$\sigma_p^2(\boldsymbol{x}_0) = \sigma^2(1 + \mathbf{w}_0^T\mathbf{R}\mathbf{w_0} - 2\mathbf{w_0}\mathbf{r_0}) \quad (32)$$

where $\mathbf{r}_0 = \lfloor R(\theta,\mathbf{x}_0,\mathbf{x}_1) \quad R(\theta,\mathbf{x}_0,\mathbf{x}_2) \quad ... \quad R(\theta,\mathbf{x}_0,\mathbf{x}_j) \quad ... \quad R(\theta,\mathbf{x}_0,\mathbf{x}_N)\rfloor^T$ represents the correlation vector between the input points and the probe point and $\mathbf{w_0}$ is the weight vector of the linear predictor equation given by $\tilde{f}(\mathbf{x_0}) = \mathbf{w_0}\mathbf{f}$. More extensive discussions on the choice of correlation functions and metamodeling using Kriging methods can be found in [36,76–78].

Correlation Function	$\gamma_k(\theta, d_k)$
Exponential	$\exp(-\theta_k \lvert d_k \rvert)$
General Exponential	$\exp(-\theta_k \lvert d_k \rvert^{\theta_{n+1}})\ ;\ 0 < \theta_{n+1} \le 2$
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	$1-3\xi_k^{\,2} + 2\xi_k^{\,3},\ \xi_k = \min\{1, \theta_k \lvert d_k \rvert\}$
Spline	$1-1.5\xi_k^{\,2} + 30\xi_k^{\,3}$ for $0 \le \xi_k \le 0.2$ $1.25(1-\xi_k)^3$ for $0.2 < \xi_k < 1\ ;\ \xi_k = \theta_k \lvert d_k \rvert$ 0 for $\xi_k \ge 1$

Table 3 : Common Correlation Models used in the Kriging Family of Methods

3.2.2. 2 The Dynamic Kriging Method

The Dynamic Kriging Method (DKG) was developed by Choi et al. [36,37] as an improvement over a general Kriging method; the features of the algorithm are:

1. Adaptive selection of the mean structure: Because a general Kriging model assumes a prior specification of the mean structure (i.e. the value of *r*), the accuracy in constructing a surrogate model hinges on a conjecture about the mean structure of the input data. In the DKG algorithm, three different polynomial models (models corresponding to r = 0, 1 and 2) are constructed. For each of the three models, a cross-validation approach is used to calculate the residue at the input points. The value of r which gives the minimum value of a cross-validation error, $\sum_{j=1}^{N}\left[\tilde{f}(\mathbf{x}_j) - \lambda_l p_l(\mathbf{x}_j)\right]^2, l = 0,...,r$, is used for constructing a surrogate model using the given inputs. Therefore, the algorithm obviates prior knowledge of mean structure of the input points and adapts to the behavior of the hypersurface in the vicinity of the input points.

1. Adaptive selection of the correlation model: Similar to the mean structure, construction of a general Kriging model assumes a known correlation model. In the DKG method, for a given

value of *r*, seven different Kriging models are constructed using the correlation functions listed in Table 3. Among the 7 possible choices, the correlation model which maximizes the log-likelihood function in Equation (31) for the given input points is used for calculating $\tilde{f}(\mathbf{x}_0)$. Therefore the algorithm does not require the use of a pre-specified correlation model.

2. Sequential Adaptive Sampling Strategy: In general in Kriging methods, to construct an initial surrogate model, a set of initial inputs are required. The subsequent input points for further computations are typically chosen by the user based on the regions of interest in the parameter space. Examples of sampling strategies include choosing samples at regular intervals, Latin Hypercube techniques [79], Latin Centroidal Voronoi Tessellation (LCVT) methods [80], etc. A majority of the sampling techniques are unsupervised in nature, i.e. the locations of the subsequent input points in the parameter space are independent of the regions of complexity of the hypersurface to be approximated. The DKG algorithm, on the other hand, is coupled with a semi-supervised adaptive sequential sampling strategy. This strategy computes the predicted variance, $\sigma_p^{\,2}$, at a large number of locations in the parameter space using Equation (32). Among these, the locations with the maximum values of $\sigma_p^{\,2}$ are selected as new input locations for performing subsequent computations. The advantage of this sampling strategy is that the method circumvents performing large numbers of computations in regions of low gradient in the parameter space. Because the sampling strategy guides the user to provide inputs at highly non-linear regions of the surrogate, this feature is particularly useful in cases where numerical computations required to generate input data are expensive.

In many cases, high-frequency noise is inherent in outcomes from physical/numerical experiments. Because the Kriging Family of methods (including DKG) interpolate through the data, surrogate models constructed retains the noise in the input data. A correction to high-frequency noise in DKG-based surrogates motivates the use of a Modified Bayesian Kriging Method (MBKG) for metamodeling. This method is presented in the next section.

3.2.2. 3 The Modified Bayesian Kriging Method (MBKG)

The MBKG method [61,62] assumes the inputs come from a stationary Gaussian random process, with a mean value of $\mathbf{P}\boldsymbol{\lambda} + \mathbf{Z}$ and variance $\sigma^2\beta$, i.e.

$$\tilde{f}(\boldsymbol{x}_0) \sim MVN(\mathbf{P}\boldsymbol{\lambda} + \mathbf{Z}, \sigma^2\beta\mathbf{I}) \tag{33}$$

where, as in the DKG method, $\mathbf{P}\boldsymbol{\lambda}$ represents the mean structure, and $\mathbf{Z}$ is modeled as a Gaussian random process with zero mean and covariance $E[Z(\mathbf{x}_j)Z(\mathbf{x}_q)] = \sigma^2\mathbf{R}$, $\mathbf{R}$ being the spatial correlation of the input points, given by Equation (30). When compared to DKG the new term in the MBKG method is $\sigma^2\beta$ which is the variance of the multivariate normal distribution. The unknown parameters in the MBKG model are $\boldsymbol{\lambda}$, σ^2, $\boldsymbol{\theta}$ and β.

The construction of a surrogate model using the DKG method is based on maximum likelihood techniques, i.e. the maximum likelihood function given by Equation (31) is used to determine the model parameters such as θ. The MBKG method considers a probability distribution function over the weight space, representing the relative degrees of belief in different values for the model parameters [81]. The function is initially set to a prior distribution. The observations i.e., the input data are used to convert the prior distribution to a posterior distribution using Bayes' theorem. Bayes' theorem states that the posterior probability distribution of model parameters, $f(Parameter|Data)$ is proportional to the likelihood, $f(Parameter)$ and prior probability distribution $f(Data|Parameter)$ [82–85], i.e.

$$f(Parameter|Data) \propto f(Parameter) \times f(Data|Parameter) \tag{34}$$

The posterior distribution, which expresses the current belief about model parameters, can be used to obtain desired values of the model parameters, i.e., the mean of the posterior may be used to obtain a point estimate for the unknown parameter value. The posterior may also be used to reveal the amount of uncertainty that remains in the parameter value, i.e., probability intervals of the posterior may be used to define credible sets, which are believed to contain the true parameter value with a specified probability.

Priors may be chosen such that they are conjugate priors wherever possible, i.e., the prior distribution may be from a parametric family that takes on the same functional form as the likelihood function. However, there may be scenarios in which a conjugate prior does not exist for

a given problem. In such cases, any distribution that reflects the prior knowledge about the unknown parameters may be used as priors. Bayes' rule may still be used in such cases, but this has to be done using a numerical method. Markov Chain Monte Carlo (MCMC) is a numerical method that can be used to draw samples from high-dimensional and non-standard probability distributions. Under certain regularity conditions, MCMC may be shown to converge in distribution to samples drawn from the posterior distribution [83–85]; the current investigation uses the MCMC method to estimate the parameters for the prior distribution for the MBKG method.

For the three unknown model parameters, $\boldsymbol{\lambda}$, σ^2 and β, semi-conjugate prior distributions are used to fit the MBKG surrogate model. For the parameter $\boldsymbol{\lambda}$, the conjugate prior is a multivariate normal distribution and is expressed as

$$\boldsymbol{\lambda} \sim MVN(\boldsymbol{\mu}_{\boldsymbol{\lambda}}, \boldsymbol{\Sigma}_{\boldsymbol{\lambda}}) \tag{35}$$

where $\boldsymbol{\mu}_\lambda$ and $\boldsymbol{\Sigma}_\lambda$ are the prior mean vector and the covariance matrix for the distribution. Similarly, the conjugate prior distribution for σ^2 and θ are Inverse-Gamma distributions expressed as

$$\sigma^2 \sim InverseGamma(\alpha_\sigma, \beta_\sigma) \tag{36}$$

$$\beta \sim InverseGamma(\alpha_\beta, \beta_\beta) \tag{37}$$

where α_σ, β_σ, α_β, β_β are the prior parameters of the respective distributions. The parameters $\boldsymbol{\theta}$ are embedded in the correlation matrix, $\mathbf{R}$, and there is no known conjugate distribution type that may be used as prior distributions. The prior distribution for $\boldsymbol{\theta}$ is chosen to be a uniform distribution and is expressed as

$$\theta_j \sim \mathrm{U}\left(a_{\theta_j}, b_{\theta_j}\right) \tag{38}$$

where θ_j is the j^{th} correlation function parameter and $a_{\theta j}$ and $b_{\theta j}$ are the prior parameters for θ_j. Using Equations (35) through (38), the joint posterior distribution for the MBKG formulation given in Equation (34) may be expressed as

$$\begin{aligned} f\left(\boldsymbol{\lambda}, \sigma^2, \boldsymbol{\theta}, \beta, \mathbf{R} \middle| f(\mathbf{x})\right) \propto & \prod_{j=1}^{k}\left[\mathrm{U}\left(a_{\theta_j}, b_{\theta_j}\right)\right] \times MVN\left(\boldsymbol{\mu}_{\lambda}, \boldsymbol{\Sigma}_{\lambda}\right) \times \\ & \text{Inverse Gamma}\left(\alpha_{\sigma}, \beta_{\sigma}\right) \times \text{Inverse Gamma}\left(\alpha_{\beta}, \beta_{\beta}\right) \\ & \times MVN\left(\mathbf{0}, \sigma^2 \mathbf{R}\right) \times MVN\left(\mathbf{P}\boldsymbol{\lambda} + \mathbf{Z}, \sigma^2 \beta \mathbf{I}\right) \end{aligned} \tag{39}$$

The full conditionals for each of the model parameters, shown in [61], may be used to estimate the values of the model parameter for each MCMC sample. Because the MBKG method is evaluated using posterior distributions, the MBKG surrogate model is not a deterministic surrogate model but rather a surrogate that produces posterior distributions for the MBKG parameters. Therefore, a predicted response value does not have one deterministic value but rather has a distribution that gives the probability of the predicted response value being in any interval. The MCMC samples drawn from the predictive distribution of the response variable may be used to estimate any desired characteristics of the distribution, e.g., the mean, standard deviation and credible sets. The larger the standard deviation and the wider the credible sets, the more uncertainty there is in the predicted value. The credible sets come closer to the mean value with larger number of samples, i.e. the uncertainty in the true parameter value reduce remarkably with increasing values of *N*, the number of samples. Unless otherwise mentioned, in the current work, the term MBKG hypersurface is used to denote the *mean response* of the probability distribution of the predictions.

The DKG method is an interpolation technique, while the MBKG method is a fitting method. The choice between the two classes of approximation is a trade-off between the bias and the variances of approximation [81]. For example, if the number of input points is too large or if the input data is noisy, the DKG method reduces the bias, but may overfit through the input points resulting in high variance. On the other hand, there may be cases where the MBKG method creates too "smooth" an approximation from the input points. In such cases, the variance may be low, but the bias may be considerably high. In the context of bridging scales in a multiscale-modeling framework, the general behavior of the closure terms in the parameter space is unknown.

Therefore, it is useful to examine both the methods as candidate metamodeling techniques and perform a critical analysis of the hypersurface approximated by both these methods. To create hypersurfaces for closure terms, mesoscale computations need to be performed; the method for performing mesoscale computational experiments is described in the next section.

3.3 MESOSCALE COMPUTATIONS TO CONSTRUCT SURROGATE MODELS

The mesoscale computational model consists of the gas phase and the particle phase. The particles are modeled as rigid cylinders immersed in a flow at various Mach numbers. For sufficiently high fluid velocities and small particles the viscous time scales are larger than the shock propagation time scales and drag on the particles is dominated by pressure forces. Therefore, the Euler equations are solved in the fluid phase. The mesoscale computational domain is illustrated in Figure 22. The setup consists of an array of 41 particles of diameter d, immersed in a uniform flow with ($\mathbf{u}$,ρ,p) = (**0.0**,1.0,1.0). The particles are arranged in a square of 1.0×1.0 units and the local particle volume fraction ϕ is given by

$$\phi = 41\frac{\pi}{4}d^2 \tag{40}$$

To vary the particle volume fraction, the diameter of the particle d is varied to achieve a target volume fraction. The simulation is initiated with an imposed shock of Mach number, Ma, located some distance upstream of the particles. The initial thermo-mechanical properties of the shocked gas (air) are computed from the Rankine-Hugoniot jump conditions. Slip boundary conditions are specified at the top and bottom surfaces of the computational domain while inflow and outflow boundary conditions are maintained at the left and right of the domain respectively.

In the mesoscale computations, the dimensions of the square in which the particle cluster is arranged is selected as the reference length scale, l_{ref} and is set to l_{ref} = 1.0. Corresponding to l_{ref}, a reference time scale, t_{ref} is defined as $t_{ref} = l_{ref} / u_s$, such that an incident shock of speed u_s traverses the particle cluster in t_{ref} =1 unit of time. Throughout the paper, the drag forces on particles and other transient quantities are presented as a function of a scaled time t^* which is defined as the ratio of the non-dimensional time t (as in Equation (27)) to the reference time t_{ref}, i.e. $t^* = t/ t_{ref}$.

Since the shock traverses the particle cluster in time of order t_{ref} =1 using the scaled time t^* allows for comparison of the dynamics across the range of parameters (ϕ, *Ma)*.

Once the flow field is computed for given (ϕ, *Ma)* the drag on an immersed object is computed by integrating the pressure *p* over the surface of the object. The drag force is computed individually for each particle in the mesoscale domain. A homogenized mesoscale drag $\tilde{F}_D(Ma, \phi)$ is used as an input to the surrogate model. Because mesoscale computations are expensive, numerical experiments can only be performed at selected values (*Ma*, ϕ*)* in the parameter space. The locations for performing mesoscale simulations in the parameter space are systematically selected in the following way:

1. First, the limits for the parameter space are specified. Numerical experiments are designed to be conducted for *1.1<Ma<3.5* and *1.28%<*ϕ*<20%*. To start with N = N_1 numerical experiments are conducted at uniformly spaced locations in the parameter space to obtain $\tilde{F}_D(Ma, \phi)$ values. An initial surrogate model for $\tilde{F}_D$ is constructed using the DKG and MBKG methods.
2. The predicted variance σ_P^2 , given by Equation (32) is computed at 100x100 locations, uniformly distributed in the parameter space. A new set of N_2 candidate locations with the highest values of σ_P^2 in the parameter space are selected for performing the next set of numerical experiments. An improved surrogate model is then constructed based on the new values of $\tilde{F}_D$ at all the $N = N_1+N_2$ locations.
3. This process is continued to a desired number of times *k*, each time adding N_k new points until a surrogate model of acceptable accuracy is constructed. Note that the number of points N_k is not fixed *a priori* and the user is free to choose the number of new mesoscale simulations based on the computational resources available.

Surrogate models are constructed for N_k = 9, 24, 40 and 56 mesoscale computations for *k = 1,2,3 and 4* respectively. The location of the mesoscale computations at each step *k* are shown in Figure 31. It may be noted that the bounds of the parameter space are determined *a priori* based on the anticipated flow conditions in a typical macroscopic computation. If the limits are found to be too restrictive, the parameter space can be augmented by performing new numerical experiments in an expanded parameter space. Once a spatio-temporally averaged drag is obtained at selected points in the parameter space, the next step is to use this information to create surrogate models

using metamodeling techniques. Both the DKG and the MBKG methods are examined for this purpose. The reason for the preference of one method over the other is not clear *a priori* and in general depends on the characteristics of the surrogate model to be generated. The decision of choosing one method over the other is based primarily on which method uses the least number of numerical experiments to construct a surrogate of desired accuracy. In other words, among the two methods, the one with a higher rate of convergence provides a more robust technique of constructing surrogates from mesoscale numerical experiments.

3.4 RESULTS

3.4.1 Validation of the Mesoscale Computational Model and the Metamodeling Techniques

The accuracy of a surrogate model for the effective drag force acting on a cluster of particles depends on the accuracy of mesoscale computations as well as that of the metamodeling techniques. These two components are validated as follows.

To study the accuracy of the mesoscale computations, the problem of interaction of a shock with a single cylinder is studied. The results are compared with benchmarks [86,87] in Section 7.1.1

The accuracy of the metamodeling methods are ascertained by studying the error in approximating an analytical function. The hypersurfaces created by the DKG and the MBKG methods are compared with the exact hypersurface in Section 7.1.2 to quantify the error in metamodeling.

3.4.1. 1 Validation of the Mesoscale Model: Interaction of a Shock with a Single Cylinder

To validate the calculations of SCIMITAR3D [63–69] in the present context the interaction of a shock with a single cylinder is studied. The diameter of the cylinder is selected to be 0.1 non-dimensional units and the center of the cylinder is placed at (1.1,1.0) inside a domain of size 3.0 x 1.0. A planar shock of Ma = 2.6 is initially placed at $x_1 = 1.0$. Slip boundary conditions are enforced at the top and bottom faces, while inflow and outflow conditions are maintained at the west and east faces of the domain respectively. The study is conducted for different mesh sizes of $\Delta x = \Delta y =$ 0.004, 0.0025, 0.001, 0.0009 and 0.00075.

The drag on a cylinder is compared with the calculations of [86]. Figure 23 shows that the peak drag coefficient agrees well with the benchmark [86]. The initial part of the drag corresponds to

the pressure forces on the particle resulting from the passage of the shock and is in good agreement. The disagreement in the decreasing part of the drag curve is because the present calculations are inviscid while [86] used the Navier-Stokes equations. To further validate the computational model, the case of a shock of Ma=2.81 interacting with a cylinder is studied. The locus of the upper and lower triple-points of the shock-system created after the onset of Mach reflections are compared with experimental observations of [88] and calculations of [86]. Figure 24 shows that the current calculations are in excellent agreement with the experimental observations as well as the calculations of [86]. While more extensive validation of the code SCIMITAR3D is shown in [63–69], the two problems chosen in this section validate the computations in the context of shock interactions with particles.

3.4.1. 2 Validation of the Metamodeling Techniques

In this section, the DKG and the MBKG methods for metamodeling are assessed for accuracy and convergence in representing an analytical function. For this purpose, the following function is used:

$$f(x,y) = sin(2\pi x)\cos(4\pi y) + 2; 0 \leq x, y \leq 1 \quad (41)$$

Values of the function at a set of *N* input points are supplied to the DKG and MBKG algorithms. Figure 5 shows the input points (for $N = 64$) in the parameter space. As can be seen in the figure the distribution of input points is non-uniform. These input points were selected by the DKG approach using the adaptive sampling strategy described in Section 5.2.2. Both the DKG and the MBKG methods are then used to reconstruct the hypersurface of the function given by (41) using the same set of *N* input points.

To calculate the error in prediction, a set of $M = 100\text{x}100$ probe points, distributed regularly inside the domain, are selected. The values of the function, $\tilde{f}$, as approximated by the DKG method/MBKG method at these *M* probe points are determined by interrogating the surrogate model. Contours of the hypersurfaces of $\tilde{f}$ as predicted using the DKG method and the MBKG method are shown in Figure 25b-c respectively. Both the DKG method and the MBKG method reconstruct the function reasonably well. The contours at $(x,y) = (0.5, 0.7)$ are smeared out (Figure 25c) for the MBKG method while the DKG approach produces sharper representation. This is

because the MBKG method fits the approximated function through the input points. This leads to diffusion of the contours around the inflection points. Since the DKG approach interpolates through the data it produces better representation around the inflection points. The error in approximating the hypersurface by the two metamodeling techniques is calculated as:

$$\varepsilon_{L^2} = \frac{\sqrt{\sum_{i=1}^{M}\left(f(x_i, y_i) - \tilde{f}(x_i, y_i)\right)^2}}{\sqrt{\sum_{i=1}^{M}\left(f(x_i, y_i)\right)^2}} \tag{42}$$

where $f(x_i, y_i)$ is the exact value of the function given by Equation (41), $\tilde{f}(x_i, y_i)$ is the value approximated by the DKG/MBKG method; *M* refers to the 100x100 probe points at which the error is evaluated. The error for both the methods converge monotonically (Figure 25 d). However, the rate of convergence of the error for the DKG method is higher than that of the MBKG method. which indicates that the DKG method is a better metamodeling technique in this case. However, this assessment only holds for the case where the input data is devoid of any high frequency noise. It is well known that Kriging approaches that rely on interpolation through data experience difficulties in the presence of noise [61,62]. Since the MBKG method is formulated as a fitting method, it is expected to reconstruct a smooth hypersurface from noisy input data. It is premature to pick the DKG method over the MBKG method based solely on their ability to represent analytical functions. Therefore, for the current paper both methods, DKG and MBKG, will be examined to construct surrogate models from possibly noisy simulation-derived data.

3.4.2 Mesoscale Numerical Experiments

Mesoscale computational experiments are performed to provide inputs to the surrogate model for drag. Key aspects of the physics of shock interactions with particle clusters that need to be captured by a surrogate model are described for Ma = 3.5 and ϕ = 20% in the next section. The particle clusters are modelled as an array of cylinders; for the purpose of this paper, the terms particle and cylinder is used interchangeably in the ensuing sections, unless otherwise mentioned.

3.4.2. 1 Interaction of a Shock (Ma = 3.5) with a Cluster of Particles at the Mesoscale

While modeling the interaction of shock with particle clusters, the drag on individual particles are often modeled by correlations which are developed for isolated spheres [43–49]. However the dynamics of the interaction of a shock with a particle cluster is different from that of an isolated sphere. An ensemble of particles is characterized by reflections of shock within the ensemble, which leads to constructive/destructive interference of shock waves. For example for a shock of Ma = 3.5 interacting with an array of particles of ϕ = 20%, the incident shock is reflected from each of the first column of particles and individual bow shocks are formed (Figure 7a). At t^* = 0.25, the Mach stems of the first column of particles obliquely impact the second column of particles. Thus, particle 26 (see Figure 22) is impacted sideways by the respective Mach stems of particles 1 and 6 (Figure 22) on one hand and is also impacted by a part of the growing bow shocks reflected from particles 1 and 6 on the other hand. Because of the resulting shock focusing, the peak drag for particle 26 is higher than that for particle 1 (Figure 8a). As the flow further evolves, the reflected bow shocks from the first column of particles coalesce to form a single reflected shock, RS1 (Figure 7(b)). The reflected shock from each subsequent column of particles interact with the adjacent column of particles upstream (Figure 7b). This leads to the development of a system of reflected shocks (denoted by RS2, RS3 and RS4 in Figure 7b). The flow is characterized by a series of slip lines (indicated as SL1 and SL2 in Figure 7b). As the shock traverses further through the array of particles, the downstream arrays of particles are impacted more obliquely compared to the particles upstream (Figure 7b-c). The flow is characterized by vortices of different sizes that are generated via the baroclinic mechanism. The smallest size of vortices correspond to those of a single particle. Several of these vortices merge to form larger vortices. In the absence of viscosity, the vortices are convected unhindered with the flow. At t^* = 1.62, the incident shock system leaves the array of particles completely (Figure 7c). As the flow evolves, the dominant shock system can be observed to be similar to the shock around a single bluff body [89]. A major reflected shock, RS_{eff} , an effective Mach Stem, MS_{eff} , and an effective Triple Point TP_{eff} are formed at t^* >1.62 (Figure 7d). At this stage the particle cluster acts like an effective bluff body with the flow features resembling that of a single cylinder as shown in [89].

3.4.2. 2 Selecting an RVE to Extract Data from the Particle Cluster

Closure laws for drag on particles rely on homogenization of the momentum exchanged between the particle and fluid within an RVE. The selection of a suitable RVE in a general mesoscale

simulation is not a settled issue [90–95]. Typical RVEs chosen for regular arrays in steady smooth flows exploit the symmetry or periodicity of the meso-structure [96]. In shock-particle mesoscale computations, it is not *a priori* clear how to select RVEs. The transient structures (shock systems, slip-lines, vortical wakes) arise at different time and spatial scales because of shock-particle interactions in the computational domain. The shock strength is attenuated along the length of the particle array and a part of the incident shock is also reflected by the array. Therefore, the RVE needs to be carefully chosen to extract data to be passed through a closure model to a macroscale homogenized description of the system. In other words, the question to be addressed is: what is the shock seeing when it passes through an array of particles?

In a macroscale computation, a physical particle contained within the "computational" particle is surrounded by other physical particles. The model for drag that is sought from the mesoscale model is one that applies to a "typical" particle embedded within the cluster. In mesoscale computations, the particles on the periphery of the cluster experience very different forces from those in the interior of the cluster. For example, unlike particles located in the outer periphery of the cluster (particles 1, 2, 3 etc.), the transverse shocks reflected from the boundaries of the domain (RS1, RS2, RS3 etc.) do not impact the five central particles (numbered as 13, 31, 32, 35, 36) directly (Figure 7). Also, the drag on the peripheral particles (Figure 8a) differ widely compared to the five central particles (Figure 8b). The shock profiles and the drag on the peripheral particles are an artefact of the finite size of the mesoscale geometry. To mitigate the influence of the edge effects in the cluster, a square subdomain circumscribing the central five particles is chosen as the RVE at the mesoscale. This allows data to be extracted for a representative set of particles embedded deep inside a cluster of particles.

3.4.2. 3 Characteristics of a Homogenized Drag in an RVE

To obtain the drag for a representative particle in the selected RVE, the transient drag $\overline{F_D}(t^*)$, is defined as the mean of $F_D(t^*)$ of the five central particles (particles 13, 31, 32, 35 and 36). Because $\overline{F_D}(t^*)$ is a homogenized drag in an RVE, it can be interpreted to be the average effective force on a computational particle. The variation of drag on an individual particle $F_D(t^*)$is compared with $\overline{F_D}(t^*)$, in Figure 8(c).

For an imposed shock of Ma = 3.5, two primary differences between $F_D(t^*)$ and $\overline{F_D}(t^*)$ are noticed. First, the peak value of the average drag $\overline{F_D}(t^*)$ is observed to be higher than that of a

single particle $F_D(t^*)$. This is because the definition of $\overline{F_D}(t^*)$ relies on the individual drag of five cylinders interacting with each other. Each of the cylinders is impacted by both the far-field incident shock, as well as shocks reflected from adjoining particles. Second, unlike $F_D(t^*)$ for a single particle, $\overline{F_D}(t^*)$ decreases with time. Shock attenuation through a system of particles has been observed previously [97–99]. Attenuation reflects the fact that $\overline{F_D}(t^*)$ represents the momentum exchange between the fluid and the cluster of particles. As expected, $\overline{F_D}(t^*)$ shows signatures of the shock-interactions due to neighboring particles as well as attenuation as the shock passes through the cluster. The average drag therefore captures the dynamics of shock-particle interactions in the cluster.

3.4.2. 4 Selection of a Characteristic Mesoscale Time Scale for Averaging Transient Forces

In constructing a surrogate model to supply closure terms to the macroscale the quantity of interest is the time-average of the transient drag $\overline{F_D}(t^*)$. $\overline{F_D}(t^*)$ is averaged over a mesoscale time, $\bar{t}^*$ as follows:

$$\tilde{F}_D(Ma,\phi) = \frac{\int_{t^*=0}^{t^*=\bar{t}^*} \overline{F}_D(t^*)dt^*}{\int_{t^*=0}^{t^*=\bar{t}^*} dt^*} \tag{43}$$

where $\tilde{F}_D(Ma,\phi)$ denotes the drag averaged over both space and time. One key issue in obtaining the time-averaged drag is the selection of the time sample, i.e. the value of $\bar{t}^*$. $\bar{t}^*$ is selected such that $\tilde{F}_D(Ma,\phi)$ does not vary for $t^*>\bar{t}^*$. This is based on the assumption of equilibration inside each computational particle in the macroscale. Within the computational (i.e. effective/homogenized) particle, shock reflections from neighboring physical particles equilibrate faster compared to the passage time of the shock over the computational particle.

In the particular example shown in Figure 7, it can be seen that at t t^* = 1.62, the incident shock system clears the array of particles. The effective Mach stem and the effective triple point (shown in the figure as MS_{eff} and TP_{eff}) are located outside the array of particles. After t^* = 1.62 the particles are subject only to shocks reflecting from adjacent particles, i.e. the influence of the

incident shock ceases to act on the particles. The flow around the five particles inside the array is seen to become steady for $t^* > 1.62$ (Figure 7c and Figure 7d). Hence, for computing the time-averaged drag the value of t^* for which the incident shock system leaves the array of particles is selected as the characteristic mesoscale time $\overline{t^*}$.

A spatio-temporally averaged drag on a representative particle $\tilde{F}_D(Ma, \phi)$ which is useful for macroscale computations depends on *Ma* and ϕ. Thus *Ma* and ϕ are the independent variables defining the parameter space for the surrogate model. For an RVE consisting of the five central particles, the effect of *Ma* and ϕ on $\overline{F_D}(t^*)$ and $\overline{t^*}$ are shown in the next section.

3.4.2. 5 Effect of Parameters Ma and ϕ on $\overline{F_D}(t^*)$

To investigate the effect of Ma and ϕ on $\overline{F_D}(t^*)$, the flow as a function of time t^* is compared for different values of *Ma* and ϕ. The results in this section indicate the range of physical phenomena that need to be captured in a surrogate model as *Ma* and ϕ are varied.

The Effect of ϕ on $\overline{\boldsymbol{F_D}}(\boldsymbol{t^*})$

This section examines the influence of particle volume fraction ϕ on the averaged drag in the RVE. The particle cluster is subjected to a shock of *Ma* 3.5 and the volume fraction ϕ is varied by varying the diameter d of the particles.

The flow physics shows several distinctive characteristics when the particle volume fraction is varied. In Section 7.2.1, for ϕ = 20% (d= 0.04) shocks impacted the second column of particles sideways (Figure 7a). However, for ϕ = 1.3% because the particles are located further away from each other, normal shock systems impact the second column of particles (Figure 9a) as opposed to oblique shocks for ϕ = 20% (Figure 7a). Each cylinder develops its own system of reflected shocks and Mach stems (Figure 9a). At $t^* = 1.08$, the reflected shocks from individual particles coalesce with each other and form a single reflected shock. Each subsequent column of particles develops its system of reflected shocks. The flow is characterized by the formation of several vortices corresponding to each cylinder in the flow field. Because the separation distance between the particles is large compared to the diameter of the particles, almost all particles are characterized by individual vortical wakes, which do not interact with vortices from adjoining particles. Unlike in the case of ϕ = 20% (Figure 7 c), for ϕ = 1.3% (Figure 9c), at time t^* = 1.62, the effective Mach Stem only slightly lags behind the incident shock. Furthermore, while for ϕ = 20%, t^* = 1.62 is

the characteristic mesoscale time, for ϕ = 1.3%, the shock systems leaves the array of particles before t^* = 1.62.

To compare $\overline{F_D}(t^*)$ for ϕ = 1.3% and ϕ = 20%, a non-dimensional averaged drag on the cluster $\overline{C_D}(t^*)$ is defined as follows.

$$\overline{C}_D(t^*) = \frac{\overline{F}_D(t^*)}{\frac{1}{2}\rho_i u_s{}^2 D^{eq}} \tag{44}$$

where u_s is the velocity of the incident shock of strength Ma and $D^{eq} = \sqrt{\frac{4}{\pi} \times A \times \frac{\phi}{100}}$ is the equivalent diameter of the particle cluster. The peak value of $\overline{C_D}(t^*)$is lower for ϕ = 1.3% compared to that for ϕ = 20%. (Figure 30). This is because, for ϕ =1.3%, particles are separated from each other compared to ϕ = 20% and shock reflections from neighboring particles do not strengthen the incident shock for ϕ = 1.3%. This demonstrates that similar to the normalized shock passage time, the averaged drag also depends on volume fraction of particles. Therefore, for particle clusters the volume fraction will serve as one of the independent variables of the parameter space.

<u>The Effect of Ma on $\overline{\boldsymbol{F_D}}(\boldsymbol{t}^*)$</u>

This section examines the physics to be captured during the interaction of shocks of varying strengths with an array of particles with fixed volume fraction ϕ = 20%. The study is performed for in incident shock of *Ma* = 1.1. There are several features which distinguishes the case of *Ma* = 1.1 with that of *Ma* = 3.5 discussed in Section 7.2.1. For *Ma* = 1.1, the flow behind the shock is subsonic, while for *Ma* = 3.5, the flow behind the shock is supersonic. The reflected shock emanating from the particle cluster travels back faster for *Ma* = 1.1 case compared to the *Ma* = 3.5 case (Figure 7a and Figure 10a). The flow around each particle is more homogenous (Figure 9 b) for *Ma* = 1.1 than for *Ma* = 3.5 (Figure 7 b). The Mach stem lags behind the incident shock to a lesser extent for *Ma* = 1.1 than for *Ma* = 3.5. The flow for *Ma* = 1.1 is characterized by very weak vortical structures and slip lines (Figure 10b). At t^* = 1.62, the incident shock system travels well beyond the array of particles (Figure 10c).Therefore, $\overline{t^*}$ is less than 1.62 for *Ma* = 1.1. In summary, the drag experienced by representative particles in a cluster depends on both *Ma* and ϕ. The next

section describes the procedure for constructing surrogate models for drag in a 2-dimensional parameter space defined by Ma and ϕ.

3.4.3 Surrogate Models from Numerical Experiments

To construct surrogate models using metamodeling techniques, mesoscale computations are performed to calculate a unique value of $\tilde{F}_D(Ma, \phi)$ for a given value of *Ma* and ϕ. At each of the locations, the set of values (Ma, ϕ, $\tilde{F}_D(Ma, \phi)$) comprise the inputs to the surrogate model. The inputs are used to train (i.e. to determine the unknown parameters in) the surrogate model. Once the parameters of the metamodeling techniques are defined, it can be used to construct a surrogate model for $\tilde{F}_D(Ma, \phi)$ as a hypersurface in the parameter space.

3.4.3. 1 Surrogate Modeling using the DKG Method

The surrogate model for $\tilde{F}_D(Ma, \phi)$ constructed using the DKG method is shown in Figure 33. The value of $\tilde{F}_D(Ma, \phi)$ is minimum for $Ma = 1.1$ and ϕ =1.3%. Because $\tilde{F}_D(Ma, \phi)$ increases with increasing *Ma* and ϕ, the value of $\tilde{F}_D$ is maximum for $Ma = 3.5$ and ϕ =20%. Furthermore, the variation of $\tilde{F}_D(Ma, \phi)$ with *Ma* for a given value of ϕ increases with increasing ϕ. Similarly, the variation of $\tilde{F}_D(Ma, \phi)$ with ϕ for a given value of *Ma* increases with increasing *Ma.*

The local properties of the hypersurface depend on the value of *N*. The surrogate model constructed using N_1 = 9 mesoscale computations is shown in Figure 33a. Mesoscale computations are performed at 8 points on the boundary of the parameter space and 1 point inside the parameter space (Figure 31).The hypersurface has a second-order mean-structure. Because only nine input points are used for the hypersurface (Figure 33a), localized departures from the mean structure are observed in the vicinity of the input data. These departures can be seen as kinks in the hypersurface in Figure 15a. These local deviations are alleviated as inputs are supplied to the DKG algorithm. For N = 24 (Figure 33(b)), the input points are located in closer proximity to each other (Figure 31). The overlap between the support domains of the correlation models located at the input points is sufficiently high. Therefore, local peaks and troughs in the surrogate model for $\tilde{F}_D(Ma, \phi)$ are absent in Figure 33(b). However, a high frequency oscillation is seen in to arise in the hypersurface near the ϕ = 10% line. In fact, as the number of inputs are increased, more high frequency oscillations are observed in the DKG-generated hypersurface (Figure 33(c) and Figure 33(d)).

The presence of high-frequency oscillations in the hypersurfaces constructed by the DKG method has also been observed in previous work [50]. Because DKG is an interpolation technique with no smoothing parameter, if the input data for the DKG method is noisy, then the resulting hypersurface retains the high frequency noise. There are multiple possible sources for noise in the input data supplied to DKG. First, the discretization schemes introduce errors in the computed mesoscale solutions, which decay asymptotically with grid spacing and time step size. Second, the selection of sampling time $\bar{t}^*$ also introduces fluctuations in the input data. For example, in Figure 30, the time varying response of the $\bar{C}_D(t^*)$ is seen to be oscillatory. For $Ma = 3.5$ and ϕ = 20%, the value of $\bar{t}^*$ was selected to be 1.62. If the value of $\bar{t}^*$ is selected to be $1.62 \pm \varepsilon$, where ε is an arbitrarily small number, the value of $\tilde{F}_D(Ma, \phi)$ for Ma =3.5, ϕ =20% will also be slightly altered. In general for any given Ma and ϕ, $\bar{t}^*$ is heuristically determined. Any uncertainty in determining $\bar{t}^*$ manifests as a noise in the input data, which translates to a noisy hypersurface using the DKG method. Additional numerical filters may alleviate noise from the hypersurface, but the filtering process may also entail further modeling errors.

In bridging scales in a multiscale modeling framework, noise in a surrogate model hypersurface changes the local convexity properties of the closure terms. This motivates the use of a metamodeling approach that behaves well in the presence of noisy data. In the Kriging family of method, the Bayesian approach used in the MBKG method offers such insensitivity to noise.

3.4.3. 2 Surrogate Modeling using the MBKG Method

In this section, surrogate models for $\tilde{F}_D(Ma, \phi)$ are constructed using the MBKG method. Unlike the DKG method, the MBKG method does not entail a deterministic point approximation. Instead of a unique value $\tilde{F}_D(Ma, \phi)$, the MBKG method constructs a distribution of $\tilde{F}_D(Ma, \phi)$ with a mean and variance, as described in Section 5.2.3. The mean hypersurface and those corresponding to 95% credible sets are presented in Figure 34 (a) through (d) for different value of N. Like the DKG method (Figure 33), the global properties of the hypersurfaces constructed using the MBKG method (Figure 34) are the same irrespective the value of N.

An additional feature of the MBKG hypersurface is that, in addition to the closure terms expressed by the mean hypersurface shown, the surrogate models can also supply the value of uncertainty arising from limited input points. The uncertainty arises in determining the model parameters of the MBKG method. Thus, a 95% credible set implies that there is a 95% likelihood of the

parameters of the probability distributions in Equation (39) lying in the interval defined by the credible sets.

The characteristics of the mean hypersurfaces as well as those of the credible sets for $\tilde{F}_D(Ma, \phi)$ depend on the value of *N*. For *N* = 10, the credible sets are far apart from the mean hypersurface except at the input points (Figure 34(a)). In addition, as shown in Figure 34(a), the lower 95% credible set has a negative value of $\tilde{F}_D$ almost everywhere in the parameter space. A negative value of $\tilde{F}_D(Ma, \phi)$ implies that momentum is transferred into the flow field by the solid phase. Therefore, the closure terms generated using *N* = 10 can be physically unrealistic. However, as observed in Figure 34(b) through (d), when *N* is increased, the mean hypersurface and the credible sets are non-negative throughout the parameter space. The upper and lower credible sets for $\tilde{F}_D(Ma, \phi)$ are also seen to rapidly approach the mean hypersurface with increasing values of *N* (Figure 34(b) through (d)).

A key observation from Figure 34 is that unlike the DKG method, for a given value of *N*, the hypersurface constructed by the MBKG method does not have high frequency oscillations. A detailed comparison of the hypersurfaces constructed using the DKG method and the MBKG method is presented in the next section.

3.4.3. 3 Comparison of the DKG and MBKG Techniques for Constructing Surrogate Models from Numerical Experiments

Mesoscale computations are expensive; each resolved computation takes several hours of compute time on a multiprocessor system. An ideal metamodeling technique should be capable of constructing surrogate models using as few mesoscale computations as possible. This is particularly the case when the dimensions of the parameter space increase or the mesoscale model is computed in 3DThe key issue is to determine how many mesoscale computations are needed by the DKG and MBKG methods to construct a surrogate model with a given error tolerance. This section analyzes the errors generated in the DKG and the MBKG methods in constructing surrogate models. In particular, the following aspects of the two metamodeling techniques are emphasized in this section.

Local Approximation in the Parameter Space: A metamodeling technique should converge monotonically with increasing values of *N* at every point in parameter space. To compare the DKG

and the MBKG methods in this respect, the variation of $\tilde{F}_D(Ma,\phi)$ with ϕ for a fixed Ma = 2.1 and for different values of N is shown in Figure 35. Figure 21(a) shows the behavior of DKG and Figure 21 (b) shows that for MBKG. Unlike the value of $\tilde{F}_D(Ma,\phi)$ estimated using the MBKG method, the value estimated using DKG decreases to a minimum for N = 56 and then increases for increasing N. This indicates non-monotonic convergence at a given point in parameter space; this oscillation occurs because of the sensitivity of the DKG approach to high frequency noise in the input. In contrast, as shown in Figure 35 (b), when surrogate models are constructed using the MBKG method, $\tilde{F}_D(Ma,\phi)$ decreases monotonically with increasing values of N (as seen in Fig. 21(b).

Convergence of the Surrogate model: In the previous section, the DKG method was shown to oscillate about a mean value at points in the parameter space. In theory, an oscillatory sequence may still be convergent if the oscillations asymptotically approach a fixed point. Because mesoscale computations are expensive, it is desirable that the global error in the surrogate model converges monotonically with respect to the number of input data. Broadly, there are two error measures that can be used for studying the convergence. They are as follows.

Relative Error Norm, $\varepsilon_{L_2}^{rel}$: The relative error norm uses the approximation of a surrogate model constructed using a (relatively) large number of input points, (say N_{max}), as the benchmark. The predictions of surrogate models constructed from $N < N_{max}$ input points are compared with the prediction of a surrogate model constructed from $N = N_{max}$. For N_{max} = 64, $\varepsilon_{L_2}^{rel}$ is defined as follows.

$$\varepsilon_{L_2}^{rel} = \frac{\sqrt{\sum_{i=1}^{M}\left(\tilde{F}_D\left(Ma_i,\phi_i \mid N_{\max}\right) - \tilde{F}_D\left(Ma_i,\phi_i \mid N\right)\right)^2}}{\sqrt{\sum_{i=1}^{M}\left(Ma_i,\phi_i \mid N_{\max}\right)^2}} \tag{45}$$

where i = 1,2, …,M corresponds to the number of locations in the parameter space at which the predictions are compared (points marked in red in Figure 31). Because $\varepsilon_{L_2}^{rel}$ is defined using the surrogate model corresponding to $N = N_{max}$ as benchmark, no additional mesoscale computation needs to be performed to study the convergence of the metamodeling techniques.

Absolute Error Norm, ε_{L_2}: The absolute error norm involves performing additional mesoscale computations at selected locations in the parameter space. These extra computations are not used as inputs to the surrogate models but are set aside for determining the error in metamodeling. The value of $\tilde{F}_D(Ma,\phi)$ computed at these *M* locations in the parameter space is compared with the value of $\tilde{F}_D(Ma,\phi)$ obtained from the surrogate models. Let $\breve{F}_D$ be the value of $\tilde{F}_D(Ma,\phi)$ computed by mesoscale simulations at the location (Ma_i,ϕ_i), i = 1,2,…, *M*. ε_{L_2} is then defined as follows

$$\varepsilon_{L^2} = \frac{\sqrt{\sum_{i=1}^{M}\left(\breve{F}_D\left(Ma_i,\phi_i\right)-\tilde{\tilde{F}}_D\left(Ma_i,\phi_i \mid N\right)\right)^2}}{\sqrt{\sum_{i=1}^{M}\left(\breve{F}_D\left(Ma_i,\phi_i\right)\right)^2}} \tag{46}$$

Although ε_{L_2} involves extra computations, convergence for $\varepsilon_{L_2}$with *N* is more demanding than convergence of $\varepsilon_{L_2}^{rel}$ with *N*. In fact, as will be shown below a surrogate model that converges in the $\varepsilon_{L_2}^{rel}$ sense does not necessarily converge in the ε_{L_2} sense.

The relative errors of the surrogate models constructed using both the DKG method and the MBKG method converge monotonically with *N (*Figure 35(c)). The rate of convergence is slightly higher for the MBKG method. This implies that the sequence of $\varepsilon_{L_2}^{rel}$ created for different values of *N* approaches a converged solution. However, $\varepsilon_{L_2}^{rel}$ is misleading in the context of constructing surrogate models from numerical experiments. As will be demonstrated below, the converged solution for the DKG method is not the true solution, i.e. the converged value of $\tilde{F}_D(Ma,\phi)$ predicted by the DKG method for increasing values of *N* does not approach the value computed from mesoscale computations.

To demonstrate this, *M* additional mesoscale computations are performed at the locations marked in red in Figure 31. The error, ε_{L_2}, is computed for both the DKG methods and the MBKG methods and is shown in Figure 35(d). For the MBKG method, ε_{L_2} is found to converge monotonically with *N*. Because both ε_{L_2} as well as $\varepsilon_{L_2}^{rel}$ converge for the MBKG method, the predicted value of $\tilde{\tilde{}}_D(Ma,\phi)$ approaches the solutions obtained from mesoscale computations with increasing values of *N*. For *N* > 24 for DKG, the predicted value of $\tilde{F}_D(Ma,\phi)$ does not converge to the value obtained from the mesoscale computations. This is in contrast to the convergence properties

obtained using $\varepsilon_{L_2}^{rel}$. Therefore, additional computations beyond $N = 24$ do not improve the approximation of $\tilde{F}_D(Ma, \phi)$ using the DKG method.

The above behavior is in contrast to the convergence behavior of ε_{L_2} for DKG studied in a previous work [50] and in Section 7.1.2. In Section 7.1.2 and in [44] the convergence properties of DKG was studied using analytical functions. In the absence of noise, for DKG, ε_{L_2} was found to converge monotonically with N to the true solution of the analytical functions. However, for constructing surrogate models in this work, DKG is trained with realizations from mesoscale experiments, where the input data is noisy. The amplitude of noise is sufficiently high to affect the convergence of ε_{L_2} for DKG. The MBKG method, on the other hand, is found to converge to the "true" solution irrespective of the presence of noise in the input data. Therefore, for MBKG ε_{L_2} converges both in the case of input data obtained from analytical functions as well as those obtained from realizations of mesoscale experiments. However, the convergence rate for MBKG is lower than for DKG for smooth functions.

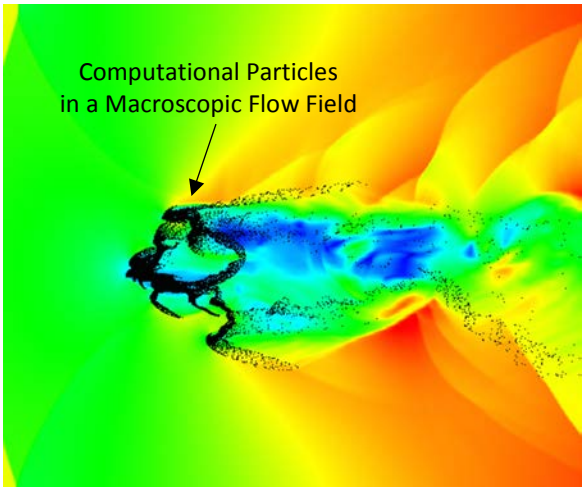
3.5 CONCLUSIONS

This work demonstrates a method for the construction of closure models for macroscale computations of shocks interacting with particle clusters. The closure laws or surrogate (meta-) models are derived from resolved mesoscale computations by using Kriging based metamodeling techniques. In the current work, two variants of the Kriging Method, viz. the DKG method and the MBKG method are tested as candidate metamodeling techniques. The input for the surrogate models are generated using mesoscale computations for an array of cylindrical particles, each with diameter d, subjected to a shock of strength, Ma. It is shown that the parameter space appropriate for the metamodel is spanned by the Mach number and the volume fraction. The SCIMITAR3D code is used to perform numerical experiments for various values of Ma and volume fraction, ϕ. A representative drag, obtained from the drag on particles averaged over space (using a suitable RVE) and time (using a characteristic mesoscale time scale) is computed for each combination of Ma and ϕ. This drag is used to compute a surrogate model for the momentum transferred by the mean flow to a computational particle in a macroscopic computation.

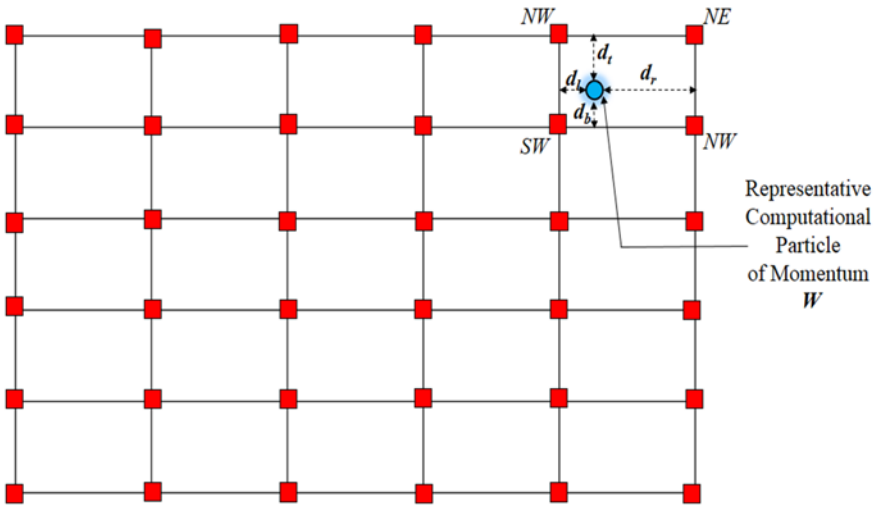
Because mesoscale computations are expensive, to keep the computational cost low, a surrogate model needs to be constructed from as few training points as possible. In this regard, the choice

between DKG and the MBKG methods for constructing surrogates is found to be a trade-off between bias and variance in approximation. In the absence of noise in the training data, the DKG method converges faster than MBKG. However when inputs are generated from mesoscale computations, the input data may contain high frequency noise. In such cases, the MBKG method not only produces a hypersurface that is devoid of high frequency oscillations but also converges monotonically as opposed to the DKG method. Therefore, to construct surrogate models from numerical experiments in a multiscale modeling framework, Modified Bayesian Kriging is recommended.

In ongoing and future work several extensions of the work presented here are being pursued. First, the investigation of appropriate choice of RVEs and time-averaging to extract data from the meso-scale was commenced in this work; more extensive studies of this issue has been conducted and will be reported in a future publication. Second, the process of constructing surrogates is being accelerated by using a multi-fidelity approach, where “rough” surrogate models are constructed using coarse grid calculations and then corrected using small numbers of fine grid calculations. Third, additional physics at the meso-scale, for example the energy removed from the imposed flow by the fluctuating components of the field in the particle cloud is being included to take the closure term to higher order. Finally, three-dimensional and moving particle calculations at the meso-scale are required to obtain a more comprehensive closure model.



(a)



(b)

Figure 20 : (a) An illustration of the macroscopic computational model of supersonic flow over an cloud of particles [50]. The computational particles, each of which is an agglomerate of physical particles, are represented as dimensionless points coupled to the surrounding flow field. (b) Schematic representation of a computational particle in an ENO Stencil in a PSIC Method [40]. Based on a first order weighting function, the source term in the governing equations for the grid point denoted by SW is $S_{SE} \propto [\boldsymbol{d_l} \quad \boldsymbol{d_b}]$W, where W is the momentum of the computational particle.

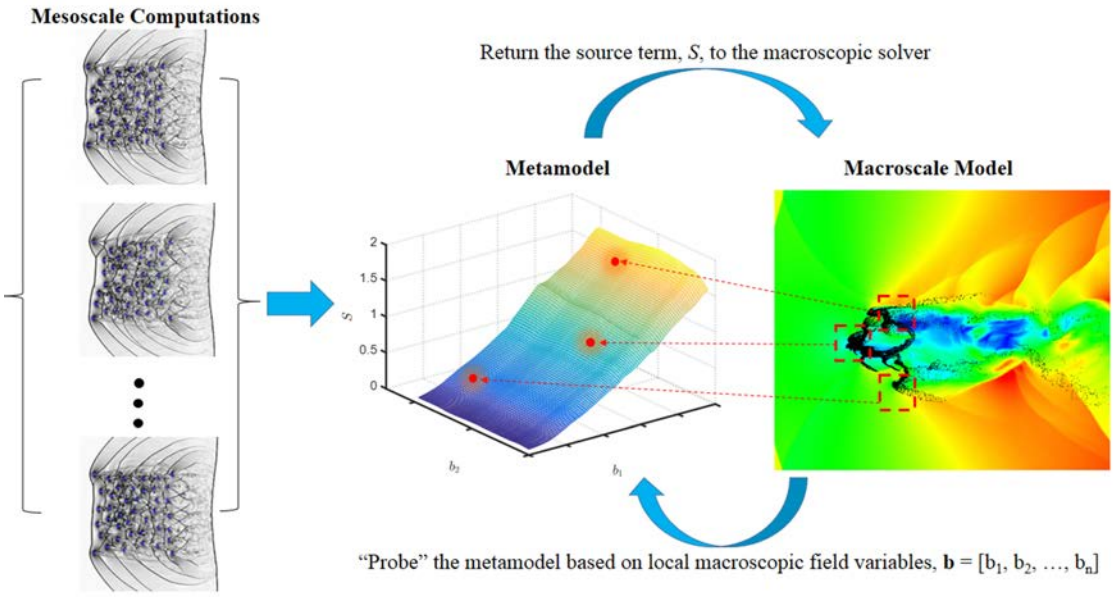


Figure 21 : Schematic representation of a multiscale modeling framework using a surrogate modeling framework. Several mesoscale numerical computations are performed *a priori* ; these are used to construct surrogate model(s) that relate the unknown source term(s) in the macroscale solver to the local macroscopic field variables In course of macroscopic computations, the surrogate model is "probed" on the fly and based on local macroscopic field variables the surrogate model returns the unknown source term(s) to the macroscopic solver, serving as surrogates to high resolution mesoscale simulations in concurrent coupling multiscale modeling approaches.

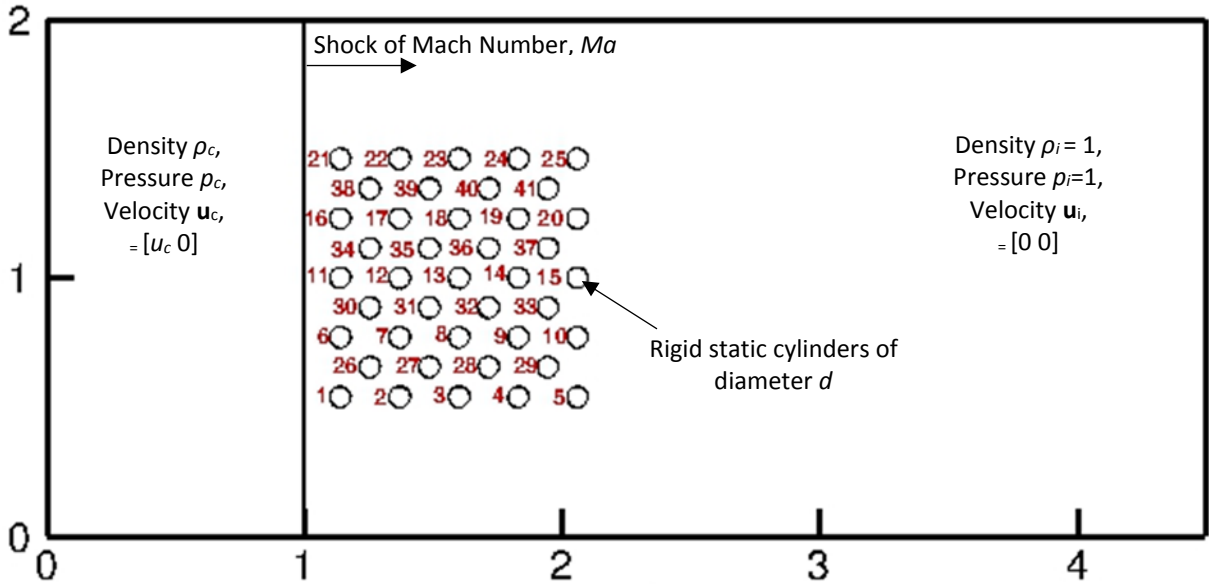


Figure 22 : Description of the set up for numerical experiments; the domain of interest comprises a right moving shock of Mach Number, Ma. To the right of the shock is an array of 41 particles of equal diameter d inscribed in an imaginary unit square (of area A = 1). The volume fraction, ϕ, of the array of particles is given by $\boldsymbol{\phi = 41\pi d^2/4}$. The particles are numbered 1 through 41 as indicated in the figure.

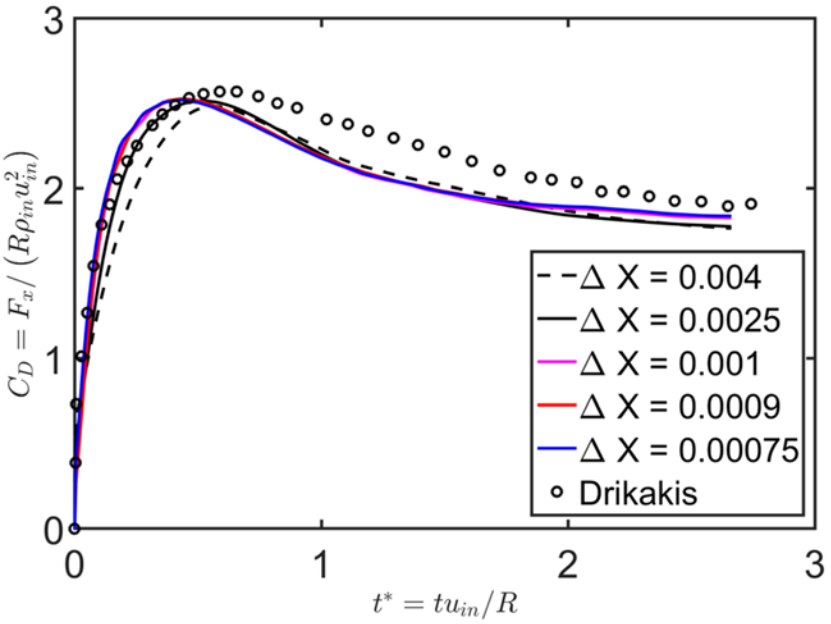


Figure 23 : Comparison of the coefficient of drag force, C_D , on a cylinder as obtained from SCIMITAR3D with the computations of [86]; the shock Mach Number, Ma is 2.6.

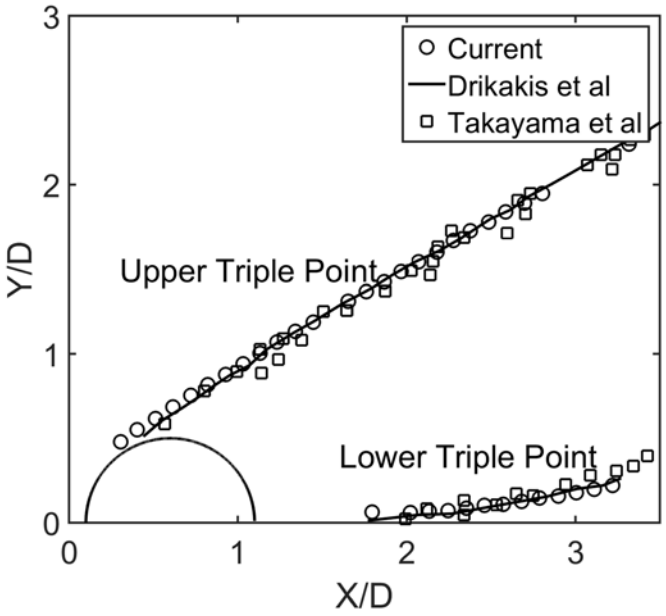


Figure 24 : Comparison of the trajectory of the lower and upper triple points as calculated from SCIMITAR3D with the computations of [86] and the experiments of [88]; the shock Mach number, Ma = 2.81; the mesh size used is $\Delta x=\Delta y= 0.0025$.

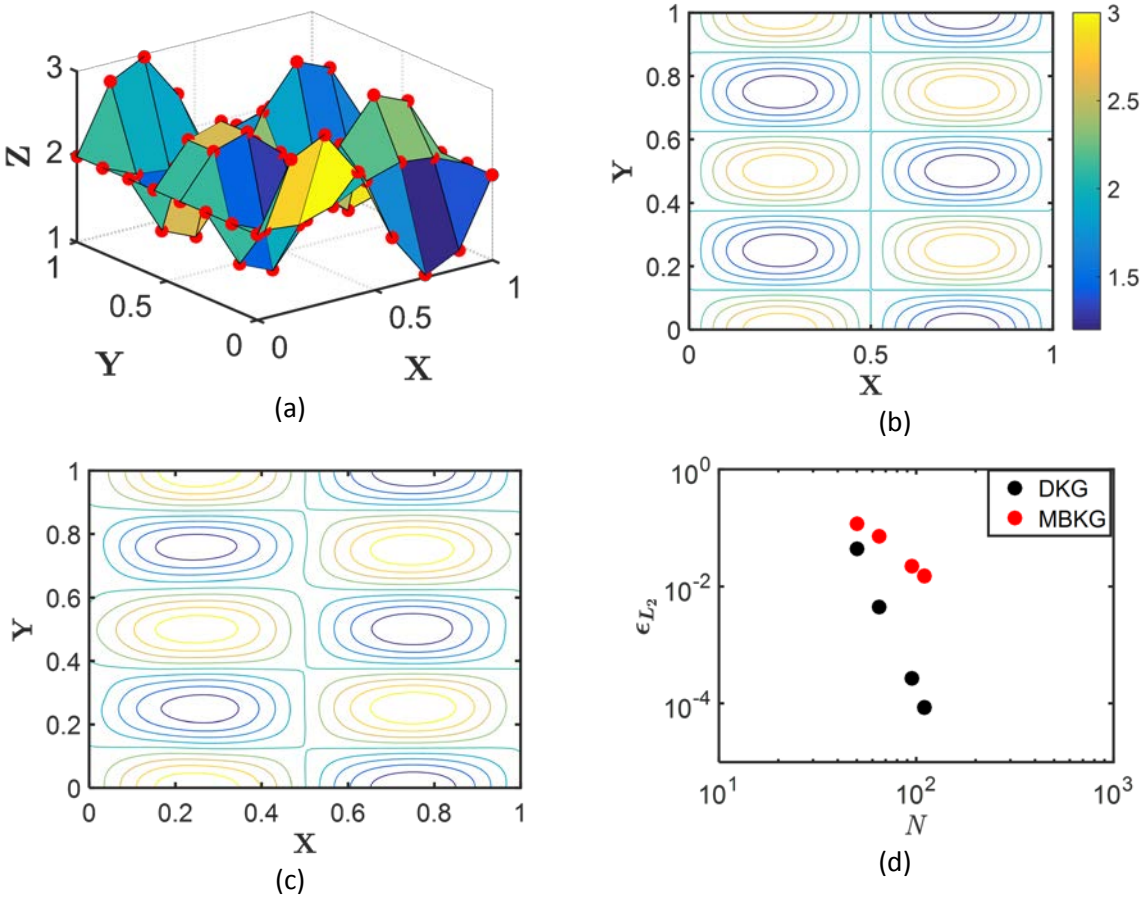


Figure 25 : Training of the DKG and MBKG methods to create surrogate models of the function given by Equation (41). (a) The hypersurface used for training the methods using an 8x8 grid point of training data (N = 64); the precise location of the training points are marked by (o). (b) The contour of the hypersurface predicted by the DKG method and (c) The contour of the hypersurface predicted by the MBKG method and (d) Convergence of the error, $\boldsymbol{\varepsilon_{L_2}}$, defined by Equation (42) with respect to N.

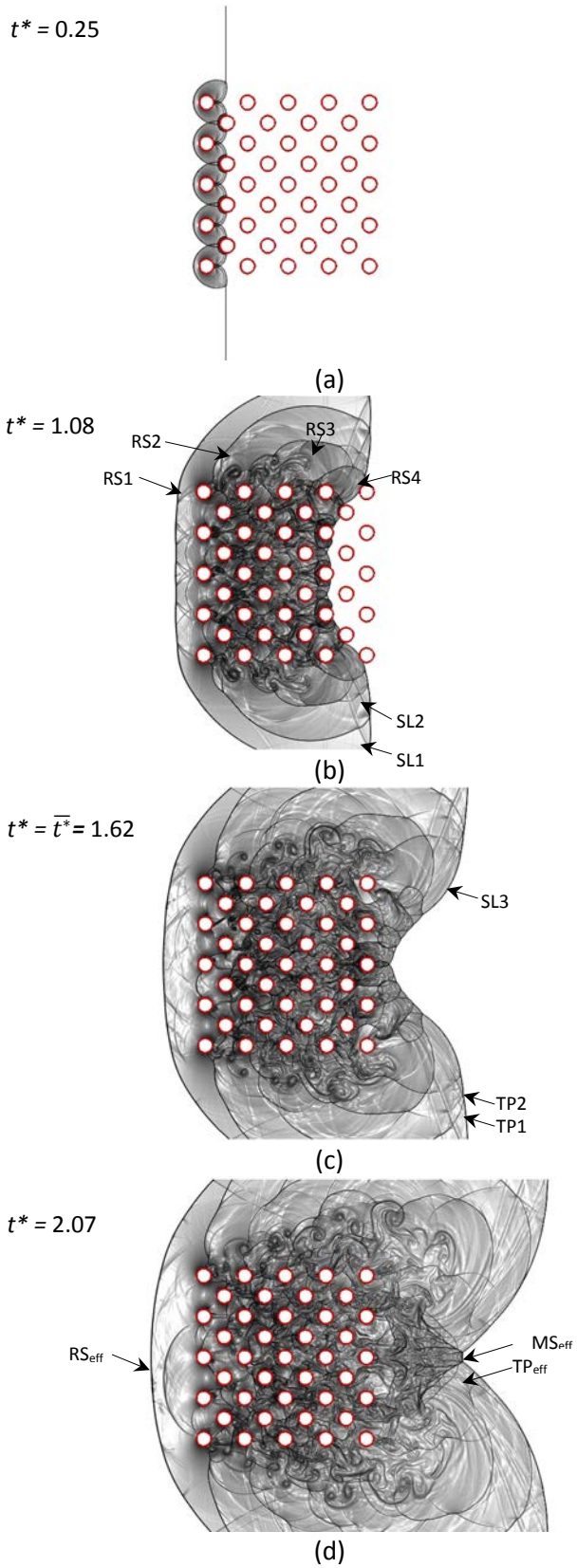
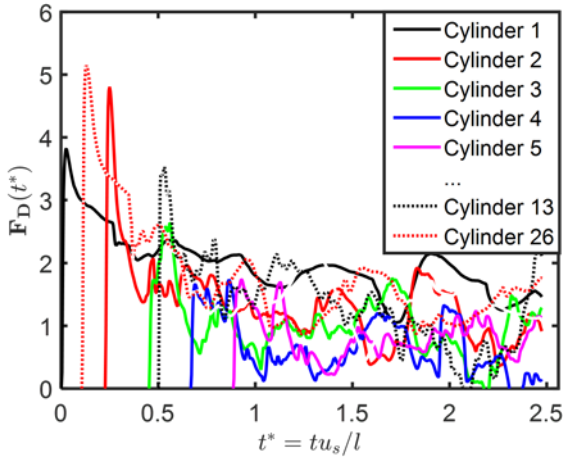
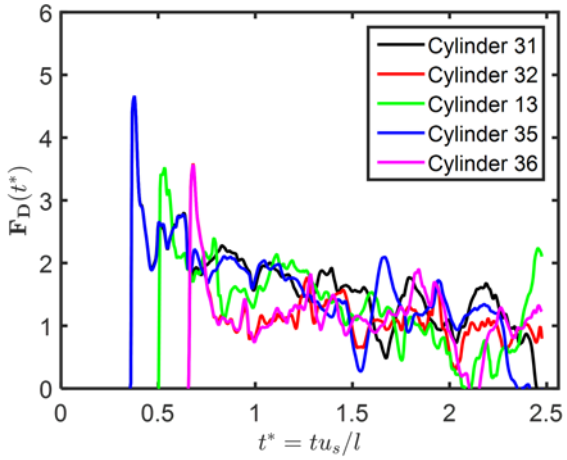


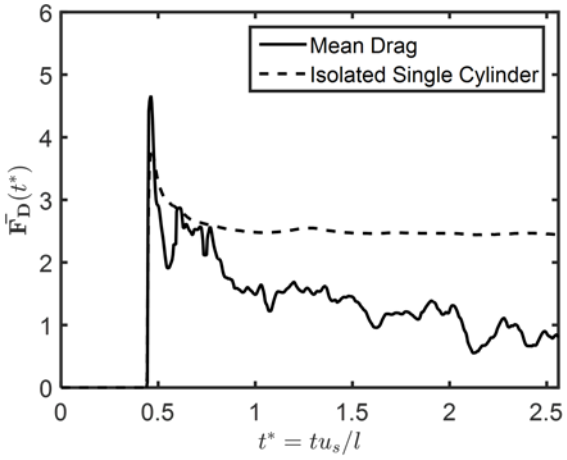
Figure 26 : Time evolution of the flow over an array of particles; Ma = 3.5, $\phi \sim 20\%$, $t^*=tu_s/l$; where u_s is the shock speed and l is the length of the unit square inscribing the cylinder. A time, $\bar{t}^*$ is the time taken by the effective lower triple point of the (resultant) shock system to traverse the array of particles.



(a)



(b)



(c)

Figure 27 : (a) Time series evolution of the drag force F_D with time t^* for Ma =3.5, ϕ = 20% (radius = 0.04) for different particles in the particle cloud (b) Time series evolution of the drag force F_D with time t^* for Ma =3.5, ϕ = 20% (radius = 0.04) for the 5 representative particles (particles 13,31,32,35 and 36) in the particle cloud (c) The drag, $\overline{\boldsymbol{F_D}}(\boldsymbol{t}^*)$ is the mean of F_D (t*) of the 5 representative particles (particles 13,31,32,35 and 36) in the particle cloud and is compared against F_D (t*) corresponding to an isolated cylinder of radius 0.04 impacted by a shock of Ma=3.5.

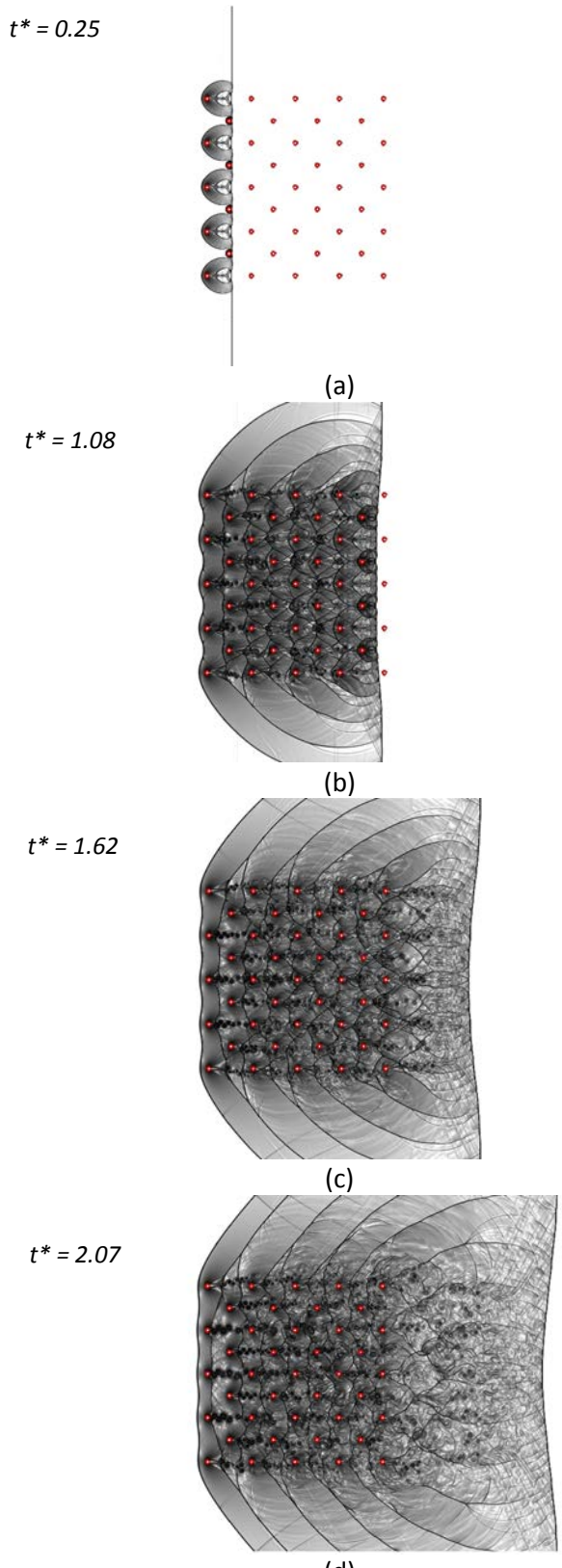


Figure 28 : Time evolution of the flow over an array of particles; Ma = 3.5, $\phi \sim 1.28\%$ $t^*=tu_s/l$; where u_s is the shock speed and l is the length of the unit square inscribing the cylinder.

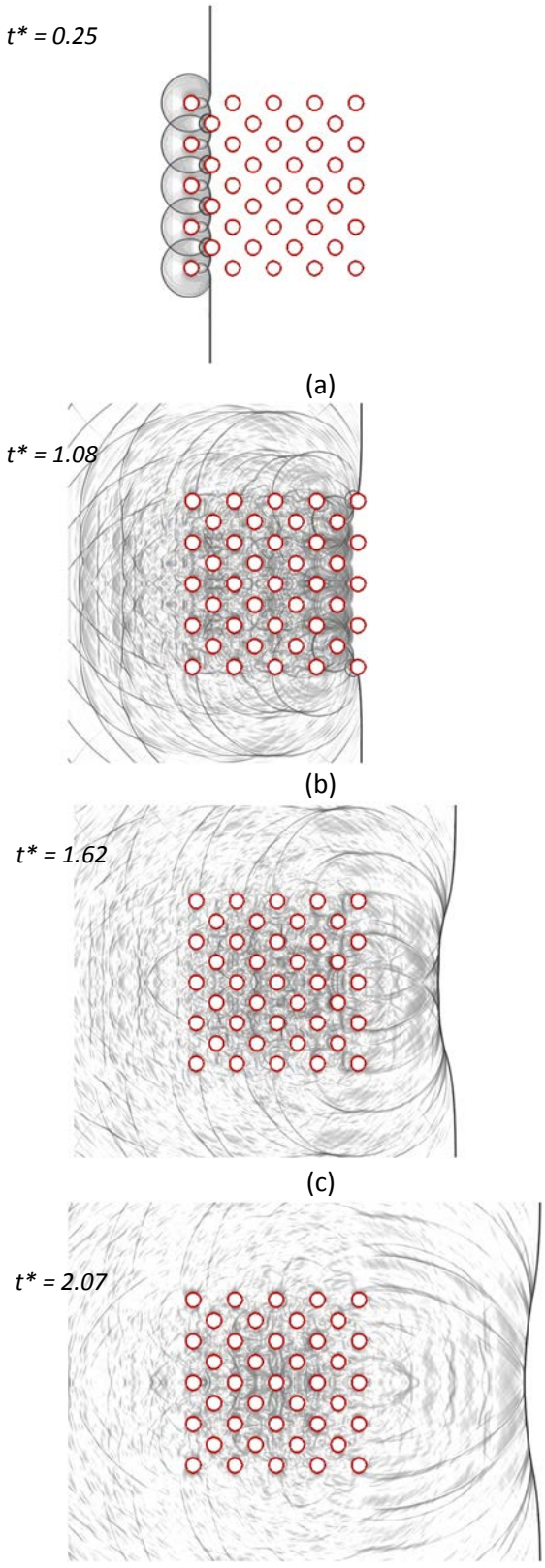


Figure 29 : Time evolution of the flow over an array of particles; Ma = 1.1, $\phi \sim 20\%$; $t^*=tu_s/l$; where u_s is the shock speed and l is the length of the unit square inscribing the cylinder.

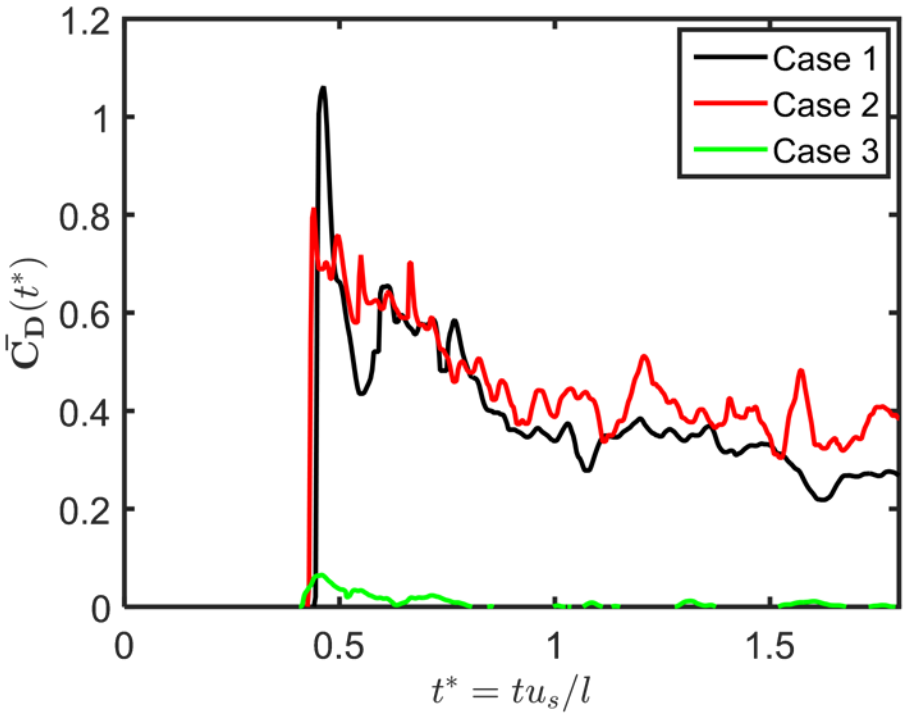


Figure 30 : Time series evolution of the drag coefficient $\bar{C}_D$ with time t^* for Case 1 (Ma =3.5, ϕ = 20%.), Case 2 (Ma =3.5, ϕ = 1%) and Case 3 (Ma =1.1, ϕ = 20%).

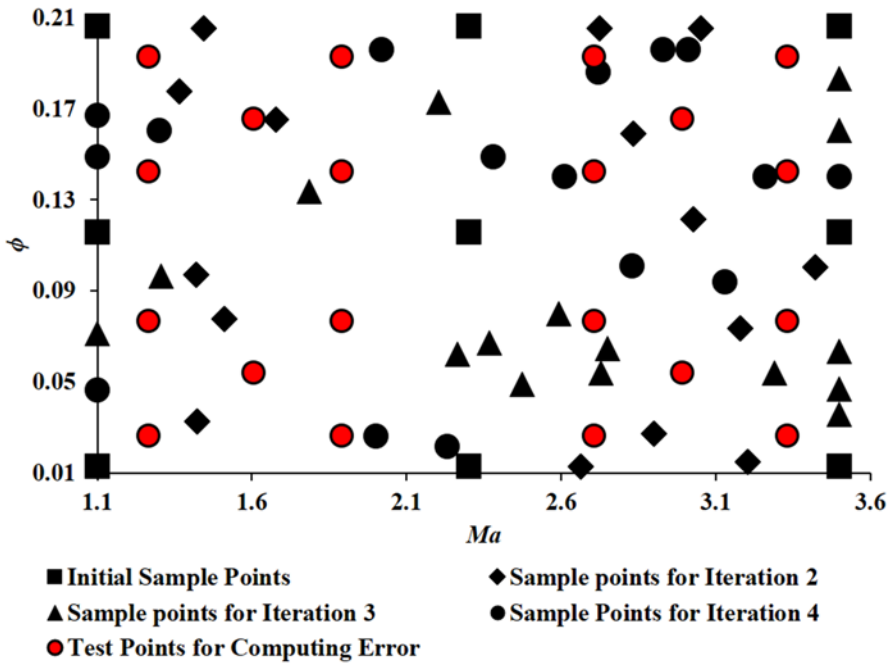


Figure 31 : Location of the points of mesoscale computations in the parameter space; these points serve as input data to create a surrogate model for $\widetilde{\boldsymbol{F}}_{\boldsymbol{D}}$. The drag at points marked as "Initial Sample Points" are first used to create a first approximation of a surrogate model for $\widetilde{\boldsymbol{F}}_{\boldsymbol{D}}$. The points marked as "Sample points for Iteration 2" are as calculated from the adaptive sampling algorithm of DKG; mesoscale computations are performed to obtain the drag at these locations for constructing the second approximation of a surrogate for $\widetilde{\boldsymbol{F}}_{\boldsymbol{D}}$. The process is repeated for other iterations. To cross-validate the quality of approximation a set of mesoscale computations are performed at the points labelled as "Test Points for Computing Error"; the drag from the computations at these points are not used to train the metamodeling techniques but are used to compare against the prediction of the surrogate model after each iteration to compute the error, $\boldsymbol{\varepsilon}_{\boldsymbol{L}_2}$.

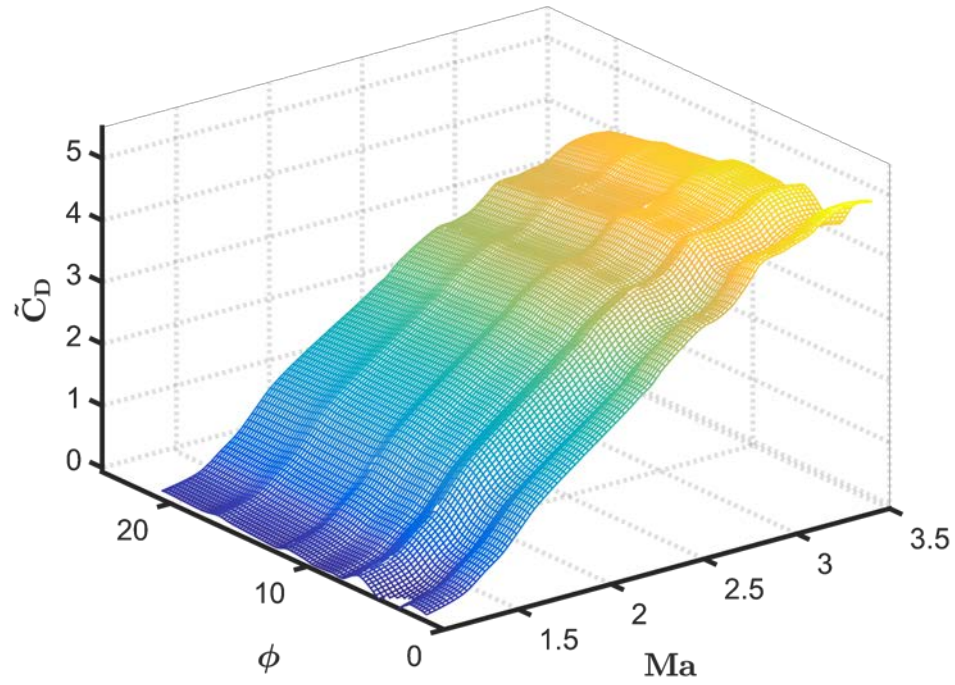
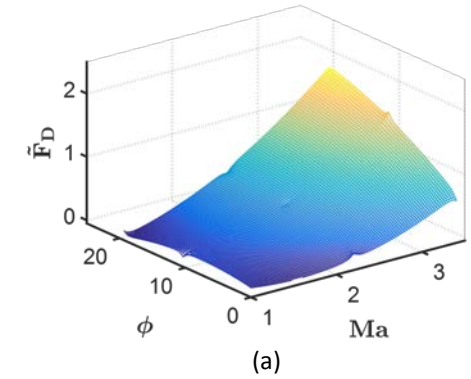
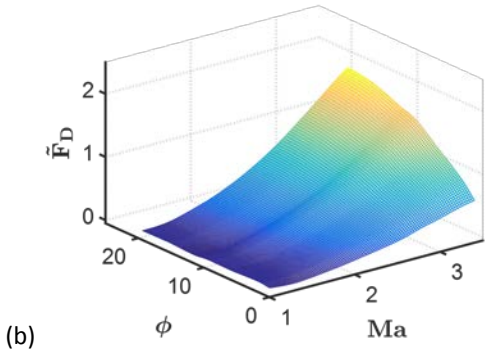


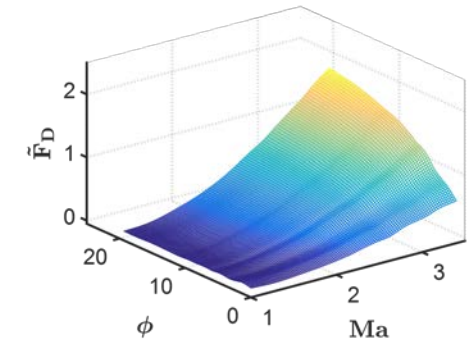
Figure 32 **:** Surrogate Model for $\widetilde{\boldsymbol{C}}_{\boldsymbol{D}}$ constructed using the DKG method using 56 mesoscale computations. The location of the inputs is shown in Figure 31**.**



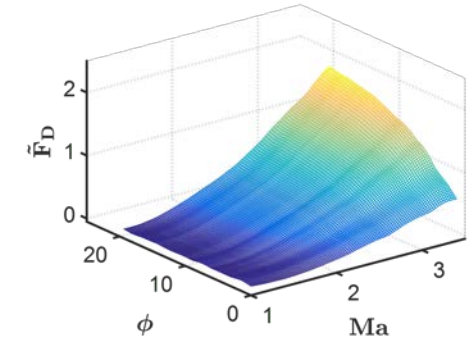
(a)



(b)

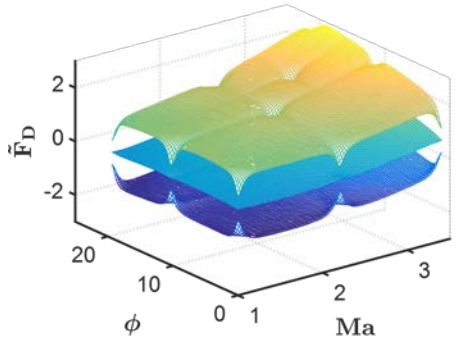


(c)

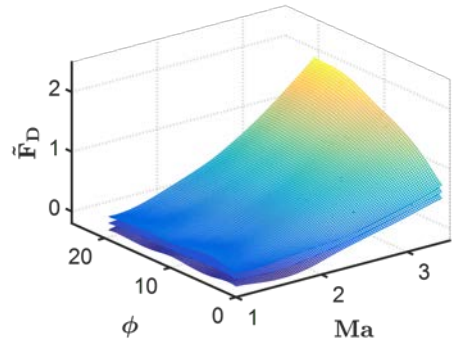


(d)

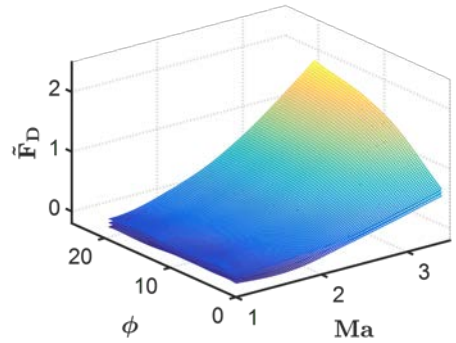
Figure 33 : Surrogate models for $\widetilde{\boldsymbol{F}}_{\boldsymbol{D}}$ constructed using the DKG method using (a) 9 (b) 24 (c) 40 and (d) 56 mesoscale computations.


$\tilde{F}_D$
2
0
-2
20
10
0
ϕ
1
2
3
Ma

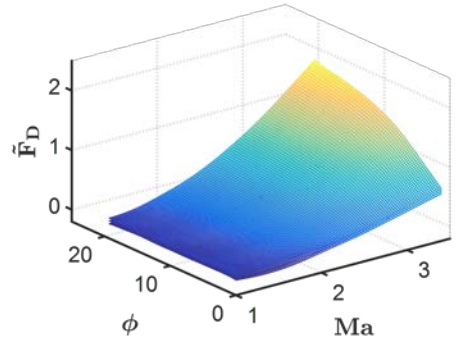
(a)


$\tilde{F}_D$
2
1
0
20
10
0
ϕ
1
2
3
Ma

(b)


$\tilde{F}_D$
2
1
0
20
10
0
ϕ
1
2
3
Ma

(c)


$\tilde{F}_D$
2
1
0
20
10
0
ϕ
1
2
3
Ma

(d)

Figure 34 : Surrogate Models for $\widetilde{F}_D$ constructed using the MBKG method using (a) 9 (b) 24 (c) 40 mesoscale and (d) 56 mesoscale computations.

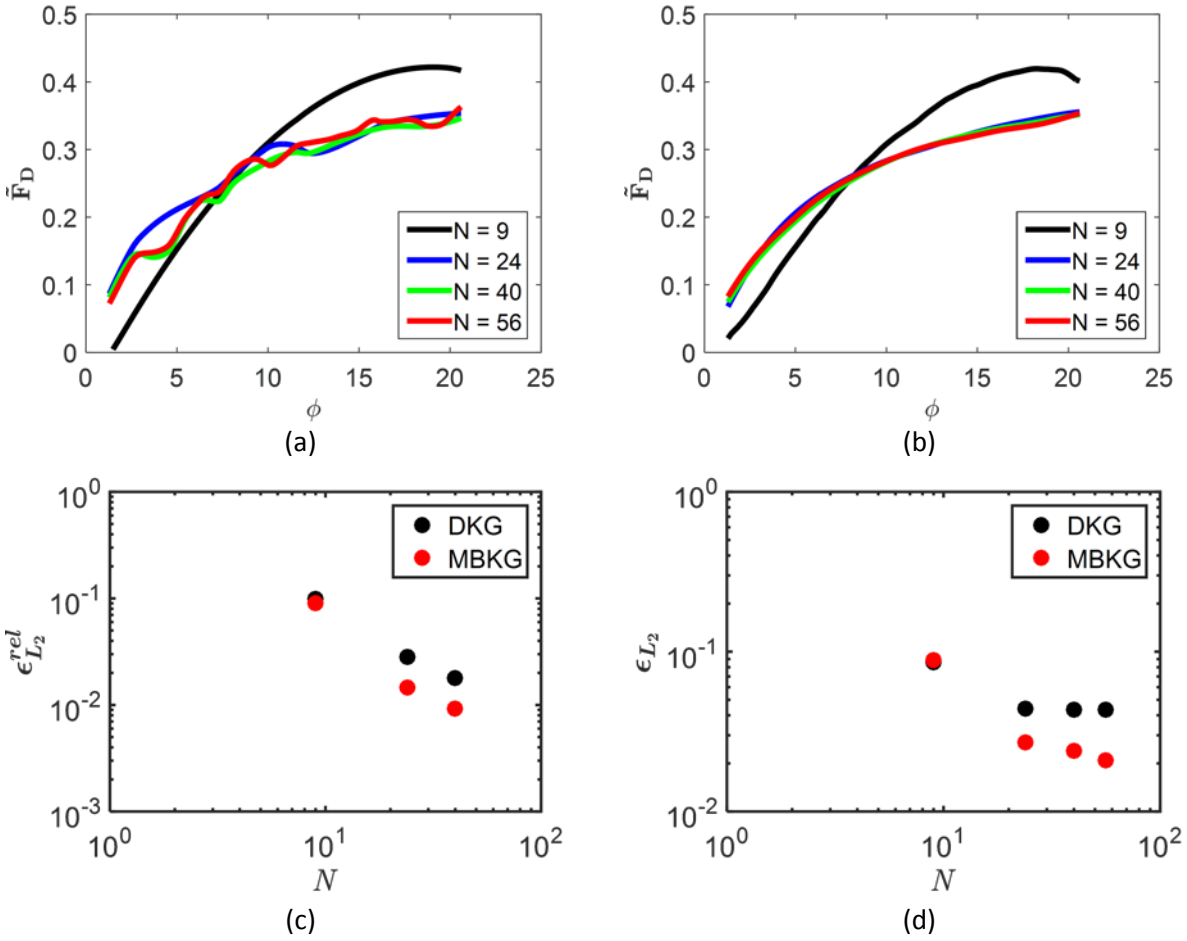


Figure 35 : Variation of $\tilde{\mathbf{F}}_\mathbf{D}$ with ϕ for Ma = 2.1 for N mesoscale computations; the surrogate model for $\tilde{\mathbf{F}}_\mathbf{D}$ is constructed using (a) the DKG method and (b) the MBKG method; (c) convergence of $\varepsilon_{L^2}^{rel}$ defined by Equation (45) with respect to N for the DKG and the MBKG method and (d) convergence of ε_{L_2} defined by Equation (46) with respect to N for the DKG and the MBKG method.

CHAPTER 4. COMPUTATION OF SUBGRID PARTICLE AVERAGED REYNOLDS' STRESS EQUIVALENCE TERMS IN PARTICLE LADEN FLOW

4.1 INTRODUCTION

Multiscale modeling of multiphase flows involving fluid-solid interactions occur in many practical applications like dusty gases [1–5], fluidized beds [6–9], blood flows [10–13], sediment transport [14–16] etc. In such flows, velocity fluctuations arising because of the interaction between the solid and the fluid phases play significant roles in gas-particle heat transfer [17–20], mixing of species [21–23] etc. In macro-scale models of such flows, the velocity-fluctuations appear as subgrid scale phenomenon and are modeled using Reynolds' Stress equivalence terms in the homogenized/phase-averaged macroscale system of equations [24]. To solve the macroscale system of equations, it is necessary to obtain closure laws for the Subgrid Particle Average Reynolds' Stress (SPARSE) terms [24,25]. This paper shows a method of generating closure laws for the SPARSE terms from resolved mesoscale computations of shocks interacting with particles.

The importance of including the effect of velocity fluctuations arising from solid-gas interactions in macroscale models of shocked-particle flows has been shown in a recent work [25]. As an example, consider an Eulerian-Lagrangian (EL) model [26–31] for macroscale computations of shock-particle flows. Traditional EL models such as Cloud in Cell (CIC) methods define a computational particle as a collection of several real physical particles. The computational particles are modeled as singular point sources in the flow field and their average motion is assumed to be under the influence of the drag force exerted by the surrounding gas. Recently [25], the traditional CIC method has been improved model by adding the effects of velocity fluctuations in the particle and fluid phases within a computational particle. In this improved approach, the instantaneous momentum-equation for a computational particle evolves as [25]

$$\frac{dv_i^p}{dt} = f(\bar{a}_i)\bar{a}_i + \frac{df(\bar{a}_j)}{d\bar{a}_j}\overline{a'_i a'_j} \tag{47}$$

Here $\bar{a}_\iota = \bar{u}_i - \bar{v}_i^p$ is the mean slip velocity of the fluid, $\bar{u}_i$ and $\bar{v}_i^p$ are the mean velocities of the fluid and the computational particle and $a_i' = u_i' - v_i^{p\prime}$ where u_i' and $v_i^{p\prime}$ are the fluid-phase and particle-phase velocity fluctuations. The first-term in the right-hand side of Equation (4) represents

the mean momentum-transfer (drag, lift) between the fluid and the particle phases and arises in typical CIC models. The stress tensor in the second term, $\overline{a'_\iota a'_\jmath}$ is an additional term that is not typical of first-order CIC models. This tensor represents the effect of the velocity fluctuations resulting from the interaction of the gas and particle phases in the subgrid/meso-scale and arise in the macroscale because of Reynolds' averaging over computational particles (hence the name SPARSE). It has been shown that [25] rather than including the drag alone, a macroscale formulation that uses both the SPARSE terms and the drag uses lesser number of computational particle than typical CIC models to trace a cloud of particles interacting with shocks.

To close the macroscale system of equations, both the drag/lift forces as well as the SPARSE tensor $\overline{a'_\iota a'_\jmath}$ must be quantified in the parameter range corresponding to shock-particle interactions. However, previous research has been limited to generating closure laws for only the drag forces exerted on the computational particle by the fluid. In a recent work [24], closure laws for the SPARSE terms have been generated using particle resolved direct numerical computations for particle laden turbulent flows; however the model is restricted to particles interactions with incompressible flows. Closure laws for the SPARSE terms for macroscale models of shock-particle interactions has received limited attention in the past and remains to be quantified in the parameter range corresponding to macroscale flow conditions.

How does one construct closure laws for the SPARSE terms as a function of the local flow conditions? Physical experiments can be used for quantifying the velocity fluctuations; however as pointed out in [24], the effect of intrusive instrumentation in experiments alters the flow field and non-intrusive experimental measurements are limited by expensive optical access. With increasing computational power, resolved mesoscale computations of particles interacting with shock are a viable alternative to physical experiments. Velocity fluctuations are explicit to mesoscale simulations and do not require closure models to be incorporated in mesoscale computations. Therefore, as noted in [32] mesoscale computations of shock-particle interactions [4,33,34] offer a promising method to quantify the SPARSE terms as a function of the parameter space.

Mesoscale computations have been used in the past [35–41] to create closure laws in different types of multiscale modeling problems. Recently [42–47], the authors used the concept of metamodeling/surrogate modeling [48–51] to construct numerical closure laws for drag on

particles interacting with shocks. Because the metamodel in [42] was created from expensive numerical computations, particular attention was paid to the selection of a suitable metamodeling technique. Among a variety of methods evaluated [42,52] it was found that a variant of the Kriging method [48,53–55], the Modified Bayesian Kriging Method (MBKG) [56,57] constructed accurate and smooth surrogates from a minimum number of mesoscale simulations. The MBKG method was used to construct surrogates for the drag on particle clusters as a function of Mach Number, *Ma* and the volume-fraction, ϕ.

Similar to the previous work, the current paper utilizes the MBKG method for creating surrogates from mesoscale numerical experiments. However, the present work advances the previous study in several different ways. First, the previous work was limited to computation of a surrogate for drag alone. Second, the particles in [42] were modeled as rigid and static cylinders. Because in most practical applications, particles are not fixed in space, modeling moving particles is a more realistic approximation of the problem of shock-particle interaction. The present work does not assume static particles, the metamodels in this study are constructed from simulations that allow for particle motion. Furthermore, the study is not limited to quantifying the drag alone in the parameter space. A more complete closure law to the macroscale equations – one that includes drag, lift as well as the SPARSE terms – are constructed in the present paper.

In the current work, the aforementioned closure laws are created from ensembles of resolved mesoscale simulations as a function of *Ma* and ϕ. The drag and the SPARSE tensor obtained from these simulations are homogenized in space and time and are used for generating surrogates for spatio-temporally averaged quantities using the MBKG method. In Section 4.2.1 , the governing equations and the numerical procedure for solving the equations for the mesoscale model are described. In Section 4.2.2 , the Modified Bayesian Kriging is briefly outlined. Section 4.3 describes the procedure for the computation of the drag, lift and SPARSE terms from the mesoscale computations. The effect of *Ma* and ϕ on these terms are described in Section 4.4.2 , while the relevant surrogate models are shown and discussed in Section 4.4.3 .Conclusions and recommendations are reserved for the final section.

4.2 GOVERNING EQUATIONS AND NUMERICAL METHODS

4.2.1 Governing Equation and Numerical Method for Mesoscale Computations

4.2.1. 1 The Governing Equations

In the mesoscale computations, the gas flow is modeled by the compressible Euler equations:

$$\begin{gathered} \frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_i)}{\partial x_i} = 0 \\ \frac{\partial (\rho u_i)}{\partial t} + \frac{\partial (\rho u_i u_j + p\delta_{ij})}{\partial x_j} = 0 \\ \frac{\partial (\rho E)}{\partial t} + \frac{\partial (\rho u_i E + p u_i)}{\partial x_i} = 0 \end{gathered} \tag{48}$$

where ρ, u_i, p are the density, velocity components and the pressure of the fluid respectively, while $E = e + 1/2 u_i u_i$ and e are the specific total internal energy and the specific internal energy of the fluid. The equations are closed by the ideal gas equation of state given by

$$p = \rho e(\gamma - 1) \tag{49}$$

where the specific heat ratio $\gamma = 1.4$. A well tested Eulerian code SCIMITAR3D [58–64] is used to solve Equations (27) and (28) and is described in the next section.

4.2.1. 2 Numerical Framework

The governing equations are spatially discretized on a fixed Cartesian mesh using a 3rd order Essentially Non-Oscillatory (ENO) [65] scheme and are marched in time explicitly using a 3rd order Runge-Kutta (RK) scheme. The procedure for tracking the interfaces of the solid particles and the application of the boundary conditions at the interface is summarized as follows.

4. To define embedded objects in the flow a narrow-band level-set [66] method is used; this allows tracking the object interfaces in a sharp manner. The level set field, ϕ_l at any point is the signed normal distance from the l^{th} immersed object with $\phi_l < 0$ inside the immersed object and $\phi_l > 0$ outside. The interface is implicitly determined by the zero level set fields, i.e. $\phi_l = 0$ contour represents the l^{th} immersed boundary.
5. Once the object interfaces are identified by $\phi_l = 0$ contour levels, the no-penetration boundary conditions are applied using a variant of the ghost fluid method (GFM) [67].

GFM relies on the definition of a band of ghost points corresponding to each phase of the interacting materials. The ghost points for the fluid phase are the points lying inside the embedded object (defined by $\phi_l > 0$) and real points are those which are outside.

6. Once these ghost points are identified, the next step is to populate the ghost field for the fluid. The ghost field is obtained by constructing least-squares interpolation of the field variables of real material points and by imposing the appropriate interfacial conditions [63] Once the values of the flow variables (e.g. ρ, u_i, p and E) at these points are populated with the least-squares field, the two-material problem can be converted to two single-material problems consisting of real fields and their corresponding ghost fields.

The computational algorithm described in this section is used for performing mesoscale numerical experiments to study the interaction of a shock of strength, *Ma* with a particle cluster of volume fraction, ϕ. The goal of the simulations is to compute a representative mesoscale drag, $\widetilde{F_D}(Ma, \Phi)$, lift $\widetilde{F_L}(Ma, \Phi)$, and the SPARSE tensor $\langle \tilde{S} \rangle_{ij}$ for different values of *Ma* and ϕ. Once these are computed for several numerical experiments, the next step is to create surrogate models for these terms using the MBKG method, which is described in the next section.

4.2.2 Modified Bayesian Kriging for Construction of Surrogate Models

The MBKG method [56,57] assumes the inputs come from a stationary Gaussian random process, with a mean value of $\mathbf{P\lambda} + \mathbf{Z}$ and variance $\sigma^2\beta$, i.e.

$$\tilde{f}(\mathbf{x_0}) \sim MVN\left(\mathbf{P\lambda} + \mathbf{Z}, \sigma^2 \beta \mathbf{I}\right) \tag{50}$$

where $\mathbf{P\lambda}$ represents the mean structure, and $\mathbf{Z}$ is modeled as a Gaussian random process with zero mean and covariance $E[Z(\mathbf{x}_j)Z(\mathbf{x}_q)] = \sigma^2\mathbf{R}$. The matrix, $\mathbf{R}$ is a spatial correlation of the input points, and is defined as follows.

$$\mathbf{R} = R_{jq} = R\left(\theta, \mathbf{x}_j, \mathbf{x}_q\right) = \prod_{k=1}^{n} \gamma_k\left(\theta_k, d_k\right) \tag{51}$$

where θ is a shape parameter, $\mathbf{d}_k = \left(x_{kj} - x_{kq}\right)$, k = 1,2,…,n, n being the dimension of the vector **x**. Commonly used models of the correlation functions are listed in Table 4 [68]. The unknown parameters in the MBKG model are **λ**, σ^2, **θ** and β.

Correlation Function	$\gamma_k(\theta, d_k)$
Exponential	$\exp(-\theta_k \lvert d_k \rvert)$
General Exponential	$\exp\left(-\theta_k \lvert d_k \rvert^{\theta_{n+1}}\right);\ 0 < \theta_{n+1} \le 2$
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	$1-3{\xi_k}^2 + 2{\xi_k}^3,\ \xi_k = \min\{1, \theta_k \lvert d_k \rvert\}$
Spline	$1-1.5{\xi_k}^2 + 30{\xi_k}^3$ for $0 \le \xi_k \le 0.2$ $1.25(1-\xi_k)^3$ for $0.2 < \xi_k < 1;\ \xi_k = \theta_k \lvert d_k \rvert$ 0 for $\xi_k \ge 1$

Table 4 : Common Correlation Models used in the Kriging Family of Methods

To determine the unknown parameters, the MBKG method considers a probability distribution function over the weight space, representing the relative degrees of belief in different values for the model parameters [69]. The function is initially set to a prior distribution and the observations i.e., the input data are used to convert the prior distribution to a posterior distribution using Bayes' theorem, which is given by Equation (34).

$$f\left(\text{Parameter}\middle|\text{Data}\right) \propto f\left(\text{Parameter}\right) \times f\left(\text{Data}\middle|\text{Parameter}\right) \tag{52}$$

Here, $f\left(\text{Data}\middle|\text{Parameter}\right)$ and $f\left(\text{Parameter}\middle|\text{Data}\right)$ are the prior and posterior probability distributions respectively and $f\left(\text{Parameter}\right)$ is the likelihood. The mean of the posterior is used to obtain a point estimate for the unknown parameter value, while the probability intervals of the

posterior is used to define credible sets, which are believed to contain the true parameter value with a specified probability.

Priors may be chosen such that they are conjugate priors wherever possible, i.e., the prior distribution may be from a parametric family that takes on the same functional form as the likelihood function. In cases where a conjugate prior does not exist for a given problem, any distribution that reflects the prior knowledge about the unknown parameters may be used as priors. In such cases, Bayes' rule is applied using the Markov Chain Monte Carlo (MCMC) to draw samples from high-dimensional and non-standard probability distributions.

For the three unknown model parameters, $\boldsymbol{\lambda}$, σ^2 and β, semi-conjugate prior distributions are used to fit the MBKG surrogate model. For the parameter $\boldsymbol{\lambda}$, the conjugate prior is a multivariate normal distribution and is expressed as

$$\boldsymbol{\lambda} \sim MVN\left(\boldsymbol{\mu}_{\lambda}, \boldsymbol{\Sigma}_{\lambda}\right) \tag{53}$$

where $\boldsymbol{\mu}_{\lambda}$ and $\boldsymbol{\Sigma}_{\lambda}$ are the prior mean vector and the covariance matrix for the distribution. Similarly, the conjugate prior distribution for σ^2 and θ are Inverse-Gamma distributions expressed as

$$\sigma^2 \sim \text{Inverse Gamma}\left(\alpha_{\sigma}, \beta_{\sigma}\right) \tag{54}$$

$$\beta \sim \text{Inverse Gamma}\left(\alpha_{\beta}, \beta_{\beta}\right) \tag{55}$$

where α_{σ}, β_{σ}, α_{β}, β_{β} are the prior parameters of the respective distributions. The parameters $\boldsymbol{\theta}$ are embedded in the correlation matrix, $\mathbf{R}$, and there is no known conjugate distribution type that may be used as prior distributions. The prior distribution for $\boldsymbol{\theta}$ is chosen to be a uniform distribution and is expressed as

$$\theta_j \sim \text{U}\left(a_{\theta_j}, b_{\theta_j}\right) \tag{56}$$

where θ_j is the j^{th} correlation function parameter and $a_{\theta j}$ and $b_{\theta j}$ are the prior parameters for θ_j. Using Equations (35) through (38), the joint posterior distribution for the MBKG formulation given in Equation (34) may be expressed as

$$f\left(\boldsymbol{\lambda},\sigma^2,\boldsymbol{\theta},\beta,\mathbf{R}\middle|\, f(\mathbf{x})\right)\propto \prod_{j=1}^{k}\left[\mathrm{U}\left(a_{\theta_j},b_{\theta_j}\right)\right]\times MVN\left(\boldsymbol{\mu}_{\lambda},\boldsymbol{\Sigma}_{\lambda}\right)\times \\ \text{Inverse Gamma}\left(\alpha_{\sigma},\beta_{\sigma}\right)\times \text{Inverse Gamma}\left(\alpha_{\beta},\beta_{\beta}\right) \\ \times MVN\left(\mathbf{0},\sigma^2\mathbf{R}\right)\times MVN\left(\mathbf{P}\boldsymbol{\lambda}+\mathbf{Z},\sigma^2\beta\mathbf{I}\right) \tag{57}$$

The full conditionals for each of the model parameters, shown in [56], may be used to estimate the values of the model parameter for each MCMC sample. Because the MBKG method is evaluated using posterior distributions, the MBKG surrogate model is not a deterministic surrogate model but rather a surrogate that produces posterior distributions for the MBKG parameters. Therefore, a predicted response value does not have one deterministic value but rather has a distribution that gives the probability of the predicted response value being in any interval. The MCMC samples drawn from the predictive distribution of the response variable may be used to estimate any desired characteristics of the distribution, e.g., the mean, standard deviation and credible sets. The larger the standard deviation and the wider the credible sets, the more uncertainty there is in the predicted value. The credible sets come closer to the mean value with larger number of samples, i.e. the uncertainty in the true parameter value reduce remarkably with increasing values of *N*, the number of samples. Unless otherwise mentioned, in the current work, the term MBKG hypersurface/ MBKG surrogate model is used to denote the *mean response* of the probability distribution of the predictions.

4.3 METHODS FOR CONSTRUCTING SURROGATE MODELS FOR DRAG AND SPARSE TERMS FROM MESOSCALE EXPERIMENTS

To construct surrogate models for the drag, lift and the SPARSE terms, resolved mesoscale computations of particles interacting with shocks are performed. The results from these experiments are used for training the MBKG method to obtain the required metamodels. The details of the procedure for creating surrogates is explained as follows.

4.3.1 Set up for the Mesoscale Numerical Experiments

The mesoscale computational model consists of the gas phase and the particle phase. The particles are modeled as rigid moving cylinders immersed in a flow at various Mach numbers. For sufficiently high fluid velocities and small particles the viscous time scales are larger than the

shock propagation time scales and drag on the particles is dominated by pressure forces. Therefore, the Euler equations are solved in the fluid phase. The mesoscale computational domain is illustrated in Figure 36. The setup consists of an array of n particles of radius r= 0.01 units, immersed in a uniform flow with (u,ρ,p) = (0.0,1.0,1.0). The particles are arranged in a square of 1.0×1.0 units and the local particle volume fraction ϕ is given by

$$\phi = n\pi r^2 \tag{58}$$

To vary the particle volume fraction, the number of particles, *n* is varied to achieve a target volume fraction. The simulation is initiated with an imposed shock of Mach number, *Ma*, located some distance upstream of the particles. The initial thermo-mechanical properties of the shocked gas (air) are computed from the Rankine-Hugoniot jump conditions. Slip boundary conditions are specified at the top and bottom surfaces of the computational domain while inflow and outflow boundary conditions are maintained at the left and right of the domain respectively.

In the mesoscale computations, the dimensions of the square in which the particle cluster is arranged is selected as the reference length scale, l_{ref} and is set to $l_{ref} = 1.0$. Corresponding to l_{ref}, a reference time scale, t_{ref} is defined as $t_{ref} = l_{ref} / u_s$, such that an incident shock of speed u_s traverses the particle cluster in $t_{ref} = 1$ unit of time. Throughout the paper, the drag forces on particles and other transient quantities are presented as a function of a scaled time t^* which is defined as the ratio of the non-dimensional time t (in Equation) to the reference time t_{ref}, i.e. $t^* = t/ t_{ref}$. Since the shock traverses the particle cluster in time of order $t_{ref} = 1$ using the scaled time t^* allows for comparison of the dynamics across the range of parameters (ϕ, *Ma).*

4.3.2 Computation of the Drag and the SPARSE terms in an RVE from mesoscale numerical experiments

Each mesoscale computation at the 56 locations in the parameter space resolves the flow around the *n* cylinders shown in Figure 36. The drag is different for each of the cylinders in the computational domain; but drag correlations that are useful for macroscale computations require a unique value of drag for a given value of *Ma* and ϕ. In the present work, a spatially-averaged

drag, $\overline{F_D}(t^*; Ma, \Phi)$ is defined as the average drag over all the n cylinders in the mesoscale experiment. The spatially-averaged drag, $\overline{F_D}(t^*; Ma, \Phi)$, is still a function of time; $\overline{F_D}(t^*; Ma, \Phi)$ is averaged over a mesoscale time, $\bar{t}^*$ as follows:

$$\left\langle \overline{F}_D \right\rangle (Ma, \phi) = \frac{\int_{t^*=0}^{t^*=\bar{t}^*} \overline{F}_D(t^*) dt^*}{\int_{t^*=0}^{t^*=\bar{t}^*} dt^*} \tag{59}$$

where $\langle \overline{F_D} \rangle (Ma, \phi)$ denotes the drag averaged over both space and time.

One key issue in obtaining the time-averaged drag is the selection of the time sample, i.e. the value of $\bar{t}^*$. Selection of an appropriate $\bar{t}^*$ depends on the flow regime that is of interest to the work. When particles interact with shock, there are two identifiable flow durations. In the initial part, when the shock interacts with particles, the gas in the cluster is gradually accelerated from rest. In this part of the flow, the motion of the particles is negligibly small and the entire dynamics is dominated by that of the incoming and reflected shocks. Once the shock leaves the particle cluster, the particles are accelerated and start following the shock. In the present work, only the initial part of the shock dynamics is modeled. Under this assumption, $\overline{a'_\iota a'_J} = \overline{u'_\iota u'_J}$ where $\mathbf{u} = u_i$ is the velocity of the gas in the particle cluster. Therefore, following the convention of our previous work [42], $\bar{t}^*$ is selected to be the time when the entire incident shock system leaves the particle cluster. Therefore, a spatio-temporally averaged drag on a representative particle $\langle \overline{F_D} \rangle (Ma, \phi)$is obtained for different values of Ma and ϕ. The drag-coefficient, $\langle \overline{C_D} \rangle (Ma, \phi)$, is computed by normalizing the drag force, $\langle \overline{F_D} \rangle (Ma, \phi)$ with the product of the density of the unshocked fluid,ρ, the square of the shock-velocity, u_s^2, and a diameter, D_{eq} as follows

$$\langle \overline{C_D} \rangle (Ma, \phi) = \frac{\langle \overline{F_D} \rangle (Ma, \phi)}{\frac{1}{2} \rho u_s^2 D_{eq}} \tag{60}$$

where $D^{eq} = \sqrt{\frac{4}{\pi} \times A \times \frac{\phi}{100}}$ is the equivalent diameter of the particle cluster and *A* is the total area of the unit-square inscribing the particle-cluster.

The SPARSE term is defined by the stress tensor, S_{ij}, computed as follows.

$$S_{ij}(\mathbf{x}, t) = \frac{u_i'(\mathbf{x}, t) u_j'(\mathbf{x}, t)}{\rho u_s^2} \tag{61}$$

Here, $u_i'(t)$ is the fluctuating component of the velocity field u_i of the fluid in the RVE and is defined as follows

$$u_i'(\mathbf{x}, t) = u_i(\mathbf{x}, t) - \widetilde{u}_i(t) \tag{62}$$

where $\widetilde{u}_i(t)$ is the Favre-average of the velocity u_i of the fluid in the RVE. The SPARSE terms in Equation (61) are a function of space,**x** and time, *t*. Similar to the drag, because macroscale closure laws require the SPARSE tensor to be a scalar-valued function of the parameter space, it is necessary to perform a spatio-temporal average of the terms in Equation (61). A spatial homogenization is performed by performing a Favre-average of the terms in Equation (61) over the entire RVE. The Favre-averaged SPARSE tensor, $\tilde{S}_{ij}(t)$, is still a function of time; a temporal average of the tensor is performed over time to obtain the spatio-temporally averaged SPARSE tensor, $\langle \tilde{S} \rangle_{ij}$ as follows

$$\left\langle \widetilde{S} \right\rangle_{ij} (Ma, \phi) = \frac{\int_{t^*=0}^{t^*=\bar{t}^*} \widetilde{S}_{ij}(t^*) dt^*}{\int_{t^*=0}^{t^*=\bar{t}^*} dt^*} \tag{63}$$

The spatio-temporally averaged drag and the terms of SPARSE tensor, computed using Equations (60) and (4), are scalar valued functions depending only the values of (*Ma*,ϕ). Therefore, closure laws for these terms can be used in macroscale systems of equations. To construct surrogate models for these, it is necessary to compute these terms for different values of (*Ma*,ϕ).

4.3.3 Selection of Inputs Points in the Parameter Space

Because mesoscale computations are expensive, numerical experiments can only be performed at selected values (Ma, ϕ) in the parameter space. In general in Kriging methods, to construct an initial surrogate model, a set of initial inputs are required. The subsequent input points for further computations are typically chosen by the user based on the regions of interest in the parameter space. Examples of sampling strategies include choosing samples at regular intervals, Latin Hypercube techniques [70], Latin Centroidal Voronoi Tessellation (LCVT) methods [71], etc. A majority of the sampling techniques are unsupervised in nature, i.e. the locations of the subsequent input points in the parameter space are independent of the regions of complexity of the hypersurface to be approximated. Because the MBKG method gives a measure of the uncertainty associated with the fit, it is possible to use the credible sets to identify regions in the parameter space where the error in approximation is higher. This can be used to select locations for subsequent inputs in constructing surrogate models. The locations for performing mesoscale simulations in the parameter space are systematically selected in the following way:

1. First, the limits for the parameter space are specified. Numerical experiments are designed to be conducted for $1.1<Ma<3.5$ and $1.28\%<\phi<10\%$. To start with N = N_1 numerical experiments are conducted at uniformly spaced locations in the parameter space to obtain $\langle\overline{C_D}\rangle(Ma, \phi)$values. An initial surrogate model for $\langle\overline{C_D}\rangle$ is constructed using the MBKG methods.
2. The value of credible sets obtained from fitting the model is computed at 100x100 locations, uniformly distributed in the parameter space. A new set of N_2 candidate locations with the highest values of the credible sets in the parameter space are selected for performing the next set of numerical experiments. An improved surrogate model is then constructed based on the new values of $\langle\overline{C_D}\rangle$ at all the $N = N_1+N_2$ locations.
3. This process is continued to a desired number of times *k*, each time adding N_k new points until a surrogate model of acceptable accuracy is constructed. Note that the number of points N_k is not fixed *a priori* and the user is free to choose the number of new mesoscale simulations based on the computational resources available.

Surrogate models are constructed for N_k = 13, 29, 45 and 61 mesoscale computations for *k = 1,2,3 and 4* respectively. The location of the mesoscale computations at each step *k* are shown in Figure 37, while the surrogate models are shown in the next section.

4.4 RESULTS AND DISCUSSIONS

The construction of surrogate models from numerical experiments relies of the accuracy of the mesoscale solver and results from the mesoscale code is compared with the results with the work of previous research [86, 88]. Following this, numerical experiments are performed to compute the drag and the SPARSE terms for different values of *Ma* and ϕ. The effect of *Ma* and ϕ on $\langle\overline{C_D}\rangle$ and $\langle\tilde{S}\rangle_{ij}$ is shown in Section **Error! Reference source not found.**. Finally, surrogate models for $\langle\overline{C_D}\rangle$ and $\langle\tilde{S}\rangle_{ij}$ are presented in Section 4.4.3 .

4.4.1 Validation of the Computational Code: Interaction of a Shock with a Single Cylinder

To validate the calculations of SCIMITAR3D [58–64] in the present context the interaction of a shock with a single cylinder is studied. The diameter of the cylinder is selected to be 0.1 non-dimensional units and the center of the cylinder is placed at (1.1,1.0) inside a domain of size 3.0 x 1.0. A planar shock of *Ma* = 2.6 is initially placed at $x_1 = 1.0$. Slip boundary conditions are enforced at the top and bottom faces, while inflow and outflow conditions are maintained at the west and east faces of the domain respectively. The study is conducted for different mesh sizes of $\Delta x = \Delta y = 0.004, 0.0025, 0.001, 0.0009$ and 0.00075.

The drag on a cylinder is compared with the calculations of [72]. Figure 23 shows that the peak drag coefficient agrees well with the benchmark [72]. The initial part of the drag corresponds to the pressure forces on the particle resulting from the passage of the shock and is in good agreement. The disagreement in the decreasing part of the drag curve is because the present calculations are inviscid while [72] used the Navier-Stokes equations. To further validate the computational model, the case of a shock of Ma=2.81 interacting with a cylinder is studied. The locus of the upper and lower triple-points of the shock-system created after the onset of Mach reflections are compared with experimental observations of [73] and calculations of [72]. Figure 24 shows that the current calculations are in excellent agreement with the experimental observations as well as the calculations of [72]. While more extensive validation of the code SCIMITAR3D is shown in [58–

64], the two problems chosen in this section validate the computations in the context of shock interactions with particles.

To construct surrogate models for the drag, lift and the SPARSE terms, numerical experiments of particles interacting with a shock is performed for different values of ϕ and *Ma* following the procedure described in Section 4.3 . Before presenting and analyzing the surrogates, it is instructive to observe how the local shock features in the flow field and the drag, lift and the SPARSE terms differ for different values of ϕ or *Ma.*

4.4.2 The Effect of ϕ and Ma on Local Flow Features and the SPARSE Terms for Shocks Interacting with Particles

4.4.2. 1 Effect of ϕ on Local Flow Features and the Closure Terms

To study the influence of ϕ on the flow field and the closure terms, three different cases corresponding to particle volume fractions of ϕ = 1%, ϕ = 5% and ϕ = 10% are studied. In all of these cases, the shock strength is set to a constant value of *Ma* = 3.5. While the local shock dynamics differ considerably from each other on changing the volume fraction (Figure 40), in the present work, features that are relevant to understanding the parametric dependence of the closure terms on ϕ are highlighted.

Effect of ϕ on local shock dynamics

For the same Mach Number, the shocks reflected from individual cylinders interact differently with each other for different volume fractions (Figure 40). For ϕ = 1%, the shocks reflected from individual particles do not interact with each other in the initial stages of flow (Figure 40a). At t^* = 0.74, the shock that is reflected from the overall particle cluster comprises individual bow shocks of each particle. It is only at later times (t^* = 1.82), when the two-dimensional effects of the particle cluster appear – the individual bow shocks from particles merge with those located north/south of the particles and the overall reflected shock front becomes normal to the direction of shock propagation (Figure 40a). For higher volume fractions (ϕ = 5% and ϕ = 10%), because of closer proximity of the particles in the cluster, the shocks reflected from individual cylinders merge with each other at an earlier time (Figure 40b-c). At t^* = 0.74, the overall reflected shock front is normal to the direction of the flow for y = 0.5 to y = 1.5 and becomes circular around the edges of the particle cluster. As the flow evolves with time (e.g. at t^* = 1.82), the reflected shock recedes further

for higher volume fraction (i.e. ϕ = 5% and ϕ = 10% in Figure 40b-c) compared to ϕ = 1% and curvature of the reflected shock front is larger than that of ϕ = 1%. In fact, between ϕ = 5% and ϕ = 10%, the curvature of the reflected shock front is higher for ϕ = 10%. Curvature of a shock front indicates bending of the streamlines of flow; by comparing the reflected shock fronts for the three cases in Figure 40, the two-dimensional nature of the flow is seen to increase with increasing volume fraction.

The two-dimensional nature of the flow of higher volume fractions can also be seen by observing the curvature of the incident shock front for the three cases in Figure 40. For lower volume fractions (ϕ = 1%), the incident shock travels nearly unobstructed through the flow and the incident shock front is nearly normal to the direction of shock propagation (Figure 40a). The incident shock front is more obstructed for ϕ = 10%, compared to ϕ = 5% (Figure 40b and 2c); similar to the reflected front, the incident shock front is observed to be have higher curvature for higher volume fractions (Figure 40).

A larger obstruction to the mean flow results in higher transverse flow for increasing volume fractions. This can be seen by observing the barotropic vortices that are generated in the wake of the particles for different volume fractions (Figure 40). The size of the vortices depend on the diameter of the particles, which is same for all the three cases shown in Figure 40. However, for low volume fractions (ϕ = 1%), particles are located farther away from each other than for higher volume fractions (ϕ = 10%). Because of this, the vortices in the wake of particles for ϕ = 1% are shed in the direction of propagation of the incident shock (Figure 40a). For higher volume fractions, the vortices from individual particles interact with each other – the interaction is highest for ϕ = 10% followed by ϕ = 5% (Figure 40b-c). The vortices are diverted transverse to the direction of propagation of the incident shock; the transversely directed vortices are marked as TV in Figure 40b-c.

An observation that is of interest in the present study is that in Figure 40, after the incident shock leaves the particle cluster completely, shocks from adjacent particles continue to reflect internally for higher volume fractions (ϕ = 5% and 10%). This sustains a fluctuating transient flow pattern in the particle cluster in Figure 40b and c for higher volume fractions. Because the particles are

located further away from each other in ϕ = *1*%, internal reflections of the shocks from adjoining particles are lesser compared to higher volume fractions.

To summarize the observations, it is seen in Figure 40 that the two-dimensional effect of particle clusters is higher for higher volume fractions – this results in directing the mean flow transversely. Because of higher internal reflections of shocks from particles within a cluster, a fluctuating transient flow is maintained for higher volume fractions after the incident shock leaves the cluster completely. In light of these two observations, the variation of the drag, lift and SPARSE terms with volume fraction can be analyzed for the three cases of a shock interacting with ϕ = 1%, 5% and 10%.

Effect of ϕ on the lift, drag and the SPARSE terms

In order to quantify the effect of volume fractions on the drag, lift and SPARSE terms, the evolution of $\overline{C_D}$, $\overline{C_L}$, $\tilde{S}_{ij}$ and $\overline{TKE}$ with time is studied and is shown in Figure 41. The drag, $\overline{C_D}$ is observed to be decreasing with increasing volume fraction (Figure 41a). This is because for increasing volume fractions, shocks reflected from neighboring particles interfere constructive and destructively with each other; the overall incident shock is weakened as it faces more obstruction to the flow and results in lower drag for higher volume fraction. The overall lift, $\overline{C_L}$ is negligibly small (~ 1-2% of the incident shock momentum) for the particle clusters (Figure 41b); this is attributed to the overall symmetry of the particle clusters with respect to the mean flow.

The SPARSE terms are seen to be influenced by the volume fraction of the particle cluster. The $\tilde{S}_{11}$ terms are observed to decrease and the $\tilde{S}_{22}$ terms increase with increasing volume fractions (Figure 41c-d). This is because of the bending of the streamlines and the increasing transverse flow for higher volume fractions. The cross-correlation term, $\tilde{S}_{12}$ component of the SPARSE tensor is negligibly small and is less than 0.2% of the kinetic energy of the incident shock (Figure 41e). This implies that the directions parallel and normal to the direction of shock propagation are the principal directions of the SPARSE tensor.

It is interesting to note that the $\overline{TKE}$ term varies within a few percentages of each other for differing volume fractions – especially for early regimes of the flow (Figure 41f). In this time frame, $\tilde{S}_{11}$ is inversely related to volume fraction while $\tilde{S}_{22}$ increases with increasing volume fraction. The

decrease of $\tilde{S}_{11}$ with volume fraction is almost compensated by the increase of $\tilde{S}_{22}$ keeping the $\overline{TKE}$ nearly constant with volume fraction. In the later potions of the flow, the $\overline{TKE}$ decreases rapidly for low volume fractions compared to higher volume fractions (Figure 41f). This is because of higher internal reflections of shocks within the particle cluster after the incident shock has passed by, which sustains the local shock dynamics within the particle cluster.

In conclusion, it is observed that the $\overline{TKE}$ terms are of the order of 15% of the kinetic energy of the shock (Figure 41f) - at least for high Mach Number flows, such as *Ma* = 3.5 as presented in this section. The variation of the closure terms with *Ma* is analyzed in the following section.

4.4.2. 2 Effect of Ma on Local Flow Features and the Closure Terms

To study the effect of *Ma* on the closure terms, the cases of shocks of *Ma* = 1.1, 2.3 and 3.5 interacting with particle clusters are studied. The volume fraction, ϕ of the clusters are kept constant and is set to 10%. Similar to the previous section, to understand the variation of the closure terms with *Ma*, first the flow field in the particle clusters are qualitatively studied. This is then followed by quantifying the variation of the spatially homogenized closure terms -$\overline{C_D}$, $\overline{C_L}$, $\tilde{S}_{ij}$ and $\overline{TKE}$ with time for different Mach Numbers.

Effect of Ma on local shock dynamics

The effect of varying *Ma* of the shock interacting a particle cluster of ϕ = 10% on the mesoscale flow features can be observed in Figure 42. For weaker shocks, the flow behind the shock is slower. In particular, for *Ma* = 1.1, the flow behind the shock is subsonic -unlike for *Ma* = 2.3 and 3.5, where the flow behind the shock is supersonic. The shocks reflected from the particle cluster recedes faster for *Ma* = 1.1 than for *Ma* = 2.3 and 3.5 (Figure 42a, b and c). Similar to the cases shown in the previous section, the front of the reflected shock is observed to be normal to the direction of the propagation of the incident shock from *y* = 0.5 to *y* = 1.5, while the front bends around the edges of the particle cluster for *Ma* = 2.3 and 3.5 (Figure 42b and c). The curvature of the reflected shock is seen to be highest for *Ma* = 3.5 and lowest for *Ma* = 1.1 (Figure 42a,b and c). Similar to the reflected shock front, the incident shock front also bends as it traverses through the particle cluster. The curvature of the incident shock front is seen to highest for *Ma* = 3.5 and

negligibly small for *Ma* = 1.1 (Figure 42a,b and c). The cases of *Ma* = 2.3 and *Ma* = 3.5 are also characterized by baroclinic vortices in the wakes of individual particles; in both these cases the vortices are directed transversely by the particle cluster (Figure 42b and c). These two cases are also characterized by unsteady vortical structures within the particle cluster (Figure 42b and c). In particular, the case of a stronger shock (*Ma* =3.5) is characterized by more complex unsteady vortical patterns within the particle cluster compared to *Ma* = 2.3 (Figure 42b and c). In conclusion, a higher *Ma* results in bending the incident and the reflected shock front and the flow field is characterized by more unsteady shock interactions for a given volume fraction.

Effect of Ma on the lift, drag and the SPARSE terms

The variation of the closure terms $\overline{C_D}$, $\overline{C_L}$, $\tilde{S}_{ij}$ and $\overline{TKE}$ with time for different *Ma* is shown in *Figure 43*. Because for higher *Ma*, there is a larger jump in the pressure, density and other thermomechanical properties of the gas, the drag coefficient $\overline{C_D}$ increases with increasing *Ma* (Figure 43a). Similar to the cases presented in the previous section, the lift, $\overline{C_L}$ is negligibly small (Figure 43b); this is because of the overall symmetry of the particle cluster.

The two principal terms of the SPARSE tensor - $\tilde{S}_{11}$ and $\tilde{S}_{22}$ as well as the kinetic energy ratio, $\overline{TKE}$ (Figure 43c,d and f) increases with increasing *Ma*. This is because a stronger shock results in more complex unsteady shock interactions within the particle cluster, which leads to higher values of velocity fluctuations in an RVE. Similar to the results shown in the previous section, the $\tilde{S}_{12}$ term in the SPARSE tensor is negligibly small and is less than 0.5 % of the kinetic energy of the incoming flow.

It is noteworthy that in Figure 41f and Figure 43f, the kinetic energy ratio, $\overline{TKE}$ is negligibly small for low *Ma* flows. However, as the shock strength increases, the kinetic energy of the fluctuations is seen to be significant and is of the order of 12-15% for high *Ma* flows. This implies that at least for certain classes of problems of shock interaction with particles, it is necessary to model the SPARSE terms as a function of the parameter space. The surrogate models for the SPARSE tensor is shown in the next section.

4.4.3 Surrogate Models for SPARSE terms

In the previous section, local flow features of representative mesoscale computations were analyzed to study the variation of the homogenized lift and drag coefficients as well as the SPARSE terms with *Ma* and ϕ. While in the previous section, the analysis was limited to the time evolution of the spatially homogenized closure terms for selected value of *Ma* and ϕ, in this section the variation of the spatio-temporally homogenized quantities in the parameter space is analyzed. Surrogate models for the closure terms - $\langle\overline{C_D}\rangle$, $\langle\overline{C_L}\rangle$, $\langle\bar{S}\rangle_{ij}$, and $\langle\overline{TKE}\rangle$ - are constructed from 61 mesoscale computations, following the procedure described in Section 4.4.3 . The hypersurfaces of the surrogates are shown in Figure 44.

Similar to $\overline{C_D}$, the spatio-temporally homogenized drag coefficient $\langle\overline{C_D}\rangle$ increases with increasing *Ma* (Figure 44a); this is because of stronger discontinuities across shocks of higher *Ma*. Furthermore, $\langle\overline{C_D}\rangle$ is also observed to be lower for higher volume fractions (Figure 44a). This is because, as shown in Section 4.4.2. 1 , a denser particle cluster attenuates the incoming shock [74]. Similar to the results shown in Section **Error! Reference source not found.**, the spatio-temporally averaged lift coefficient, $\langle\overline{C_L}\rangle$ is observed to be negligibly small throughout the parameter space (Figure 44b).

The $\langle\bar{S}\rangle_{11}$, $\langle\bar{S}\rangle_{22}$ components of the SPARSE tensor as well as the $\langle\overline{TKE}\rangle$ are small for lower values of *Ma* (Figure 44c-d and f) and increase linearly from *Ma* = 1.1 to *Ma* ~ 2.0. As the shock strength increases beyond *Ma* ~ 2.0, the terms nearly saturate for a given volume fraction and do not increase sharply with increasing shock velocity. In other words, for a given value of ϕ, the slopes of $\langle\bar{S}\rangle_{11}$and $\langle\bar{S}\rangle_{22}$ as well as $\langle\overline{TKE}\rangle$ with respect to *Ma* is higher when *Ma* < 2.0 than when *Ma* >2.0. A possible explanation for this is the flow behind the shock transitions from subsonic to supersonic at *Ma* = 2.067 (for γ = 1.4). As is typical for pressure, density and other thermodynamic fluid properties, the velocity fluctuations in an RVE is also observed to be more sensitive to the shock speed when the flow behind the shock is subsonic as opposed to supersonic.

For a given *Ma*, $\langle\bar{S}\rangle_{11}$ is higher for lower volume fractions compared to higher volume fractions (Figure 44c). The variation of $\langle\bar{S}\rangle_{22}$ is exactly opposite; $\langle\bar{S}\rangle_{11}$ is higher for higher volume fractions compared to lower volume fractions (Figure 44d). This is because a higher particle fraction turns the flow transversely (as shown in Section 4.4.2. 1) thereby the velocity fluctuations from the

direction of shock propagation to the transverse direction. It is interesting to note that on increasing ϕ, the decrease in $\langle \bar{S} \rangle_{11}$ is almost compensated by the increase in $\langle \bar{S} \rangle_{22}$. This is observed by noting that the $\langle \overline{TKE} \rangle$ varies negligibly with ϕ (Figure 44f). The total kinetic energy of the fluctuations is conserved along the ϕ and is predominantly a function of *Ma*.

From Figure 44f, it is also noted that the kinetic energy of the fluctuations is about 5-8% of the kinetic energy of the incoming shock for *Ma* > 2.0. Finally, it is noted that $\langle \bar{S} \rangle_{12}$ is negligibly small (Figure 44e) – the direction of shock propagation and the one transverse to it comprise the principal directions of the SPARSE tensor.

4.5 CONCLUSIONS

This paper quantifies the Subgrid Particle Average Reynolds' Stress (SPARSE) terms which arise as unclosed terms in homogenized Eulerian Lagrangian macroscale models of shock-particle interactions. The closure laws or surrogate (meta-) models are derived from resolved mesoscale computations of particles (modeled as a cluster of cylinders) of volume fraction ϕ interacting with a shock of strength, *Ma*. The SCIMITAR3D code is used to perform numerical experiments for various values of *Ma* and volume fraction, ϕ. The drag, lift and SPARSE terms computed from mesoscale simulations are averaged over space (using a suitable RVE) and time (using a characteristic mesoscale time scale) and are calculated for each combination of *Ma* and ϕ. These are then used to construct surrogate model for the drag and lift coefficients, SPARSE terms and the kinetic energy of the velocity fluctuations using a Modified Bayesian Kriging Method.

It is found that the kinetic energy of the velocity fluctuations is approximately 8% for higher Mach number flows (*Ma* > 2.0). The kinetic energy, SPARSE terms and the drag coefficients are also seen to asymptotically increase for *Ma* > 2.0. In particular, the kinetic energy is predominantly a function of *Ma* and is conserved along the ϕ axis of the parameter space. As the volume fraction increases, the mean flow is transversely directed in such a way that the kinetic energy of the fluctuations is kept constant.

In ongoing and future work several extensions of the work presented here are being pursued. First, the effect of viscous dissipation on the SPARSE terms and kinetic energy of the velocity fluctuations is being pursued. Second, the present work initiates the quantification of the closures at the initial flow regimes of shock particle interaction. Once the shock has passed, the particles

are accelerated in the flow; the effect of the interaction of particle-velocity fluctuations with the fluid velocity fluctuations is being pursued. Finally, three-dimensional particle calculations at the meso-scale are being pursued to obtain more comprehensive closure models.

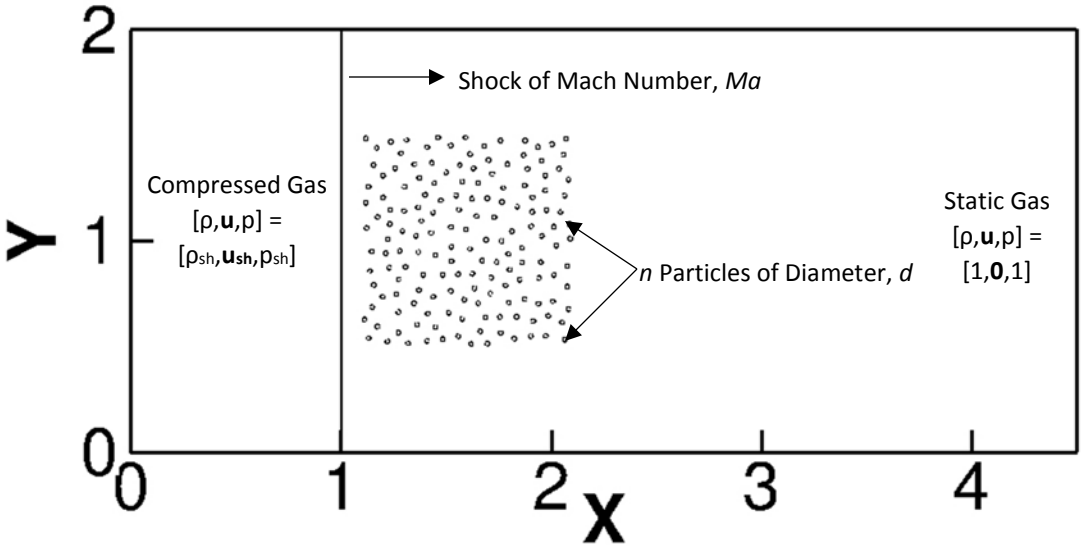


Figure 36 : Computational geometry for mesoscale calculations of interaction of a shock with a cluster of particles. The Mach Number of the shock is Ma. To the right of the shock is a static fluid ($\gamma = 1.4$), with unit density and pressure. The properties of the fluid to the left of the shock is governed by the Rankine-Hugonoit jump conditions for an ideal gas. The cluster of particles is modelled by an array of n cylinders, each of diameter d. The volume fraction of the particle cluster is given by $\phi = n\pi d^2/4$.

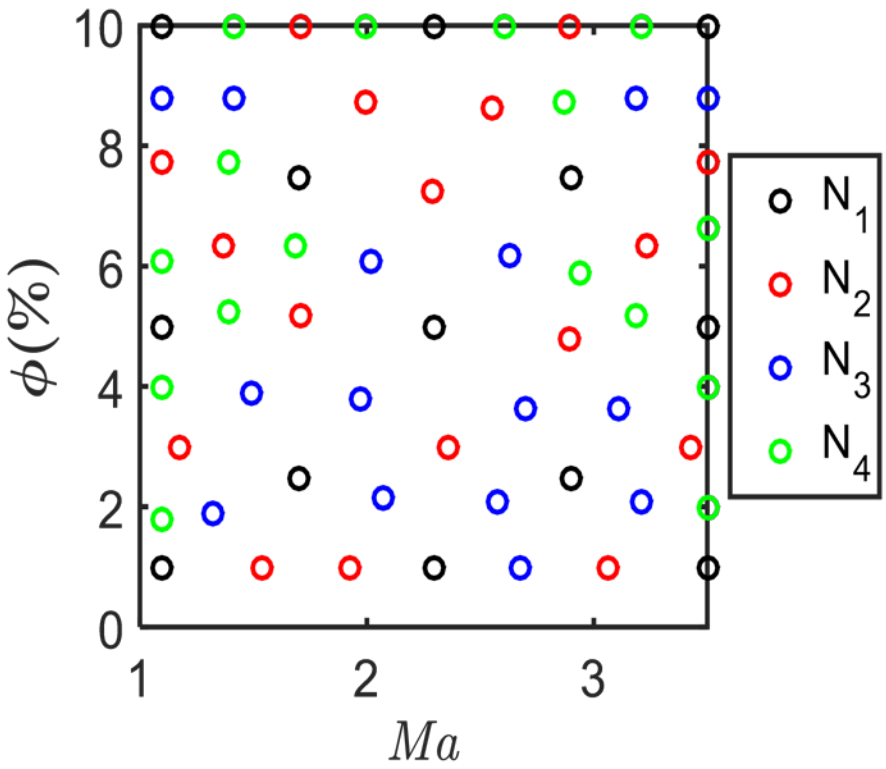


Figure 37 : Location of the points of numerical experiments in the parameter space. The input points are selected sequentially using the MBKG sampling technique described in Section 4.3.3 . The N_1 points are the initial locations of simulations, the points marked by N_2 are the locations of the next set of numerical experiments, followed by N_3 and so on.

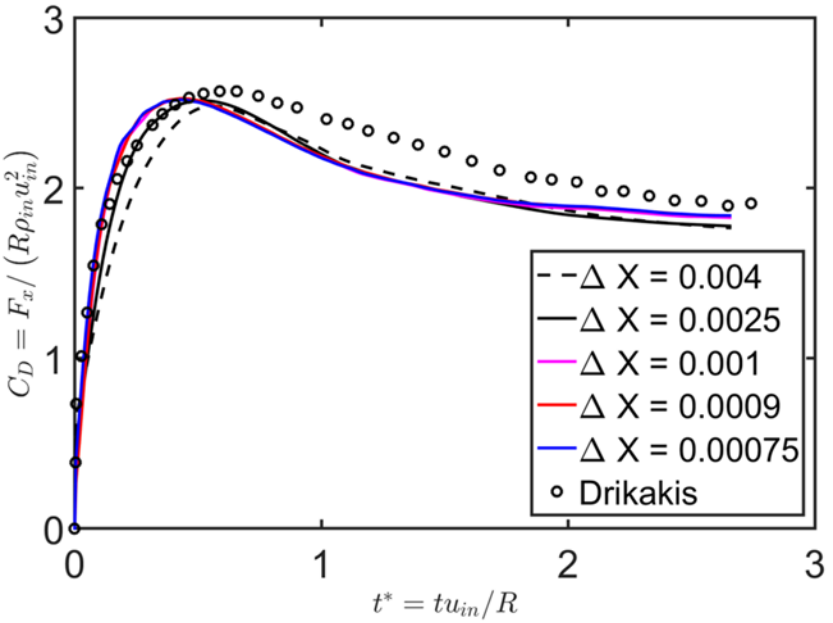


Figure 38 : Comparison of the coefficient of drag force, C_D , on a cylinder as obtained from SCIMITAR3D with the computations of [72]; the shock Mach Number, Ma is 2.6.

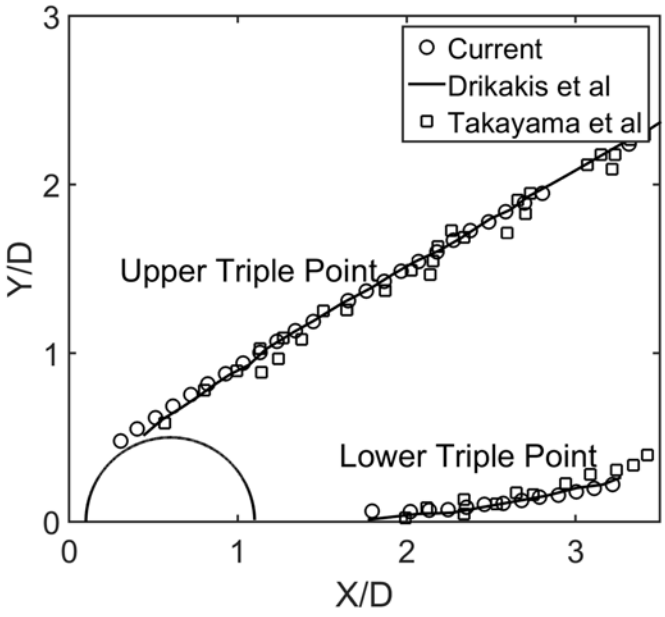


Figure 39 : Comparison of the trajectory of the lower and upper triple points as calculated from SCIMITAR3D with the computations of [72] and the experiments of [73]; the shock Mach number, Ma = 2.81; the mesh size used is Δx=Δy= 0.0025.

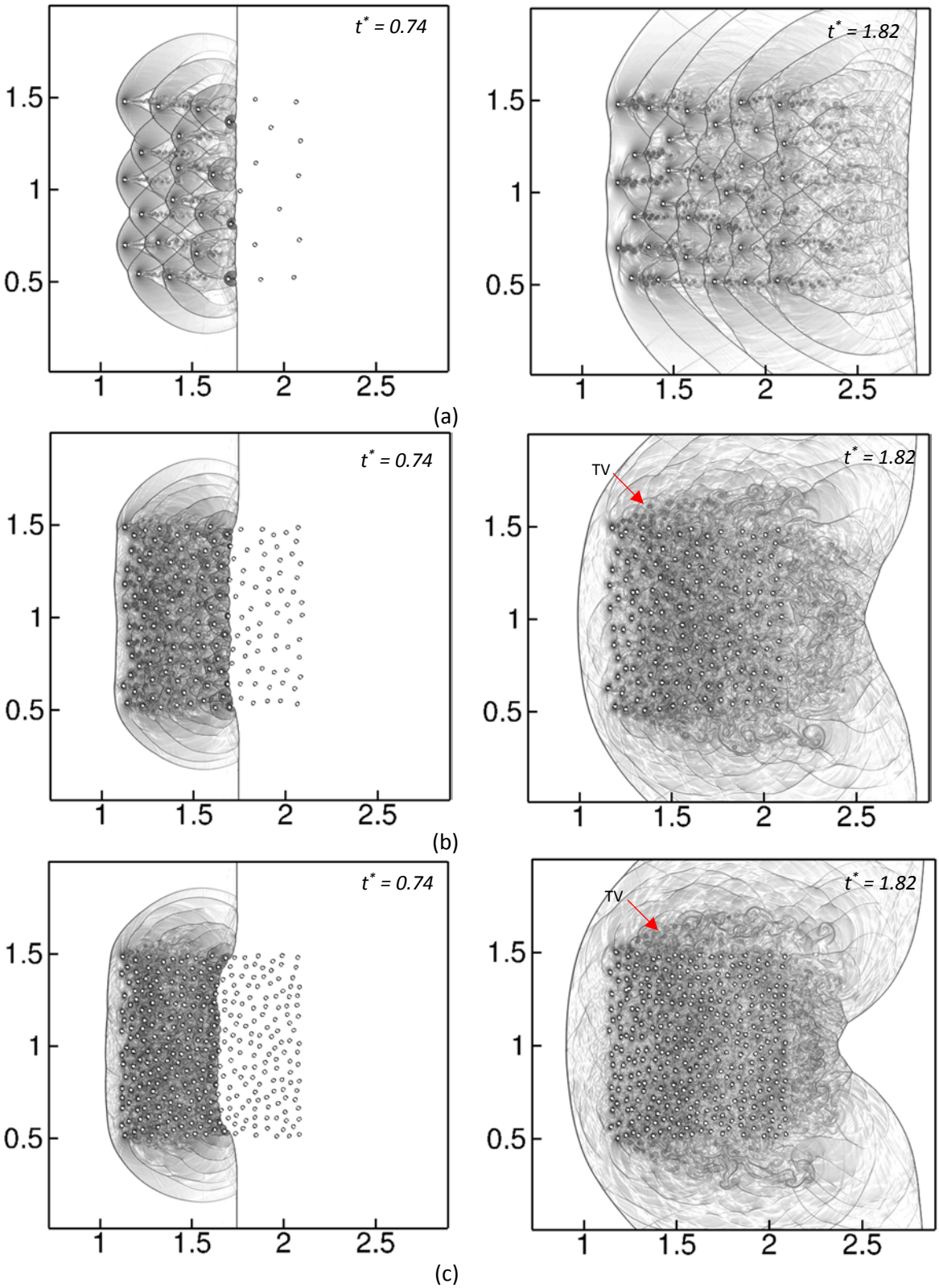


Figure 40 : Numerical Schlieren images of the interaction of a shock of Mach Number Ma with a cluster of particles of volume fraction ϕ at $t^* = 0.74$ and $t^* = 1.82$.(a) [Ma, ϕ] = [3.5,0.01], (b) [Ma, ϕ] = [3.5,0.05], and (c) [Ma, ϕ] = [3.5,0.1].

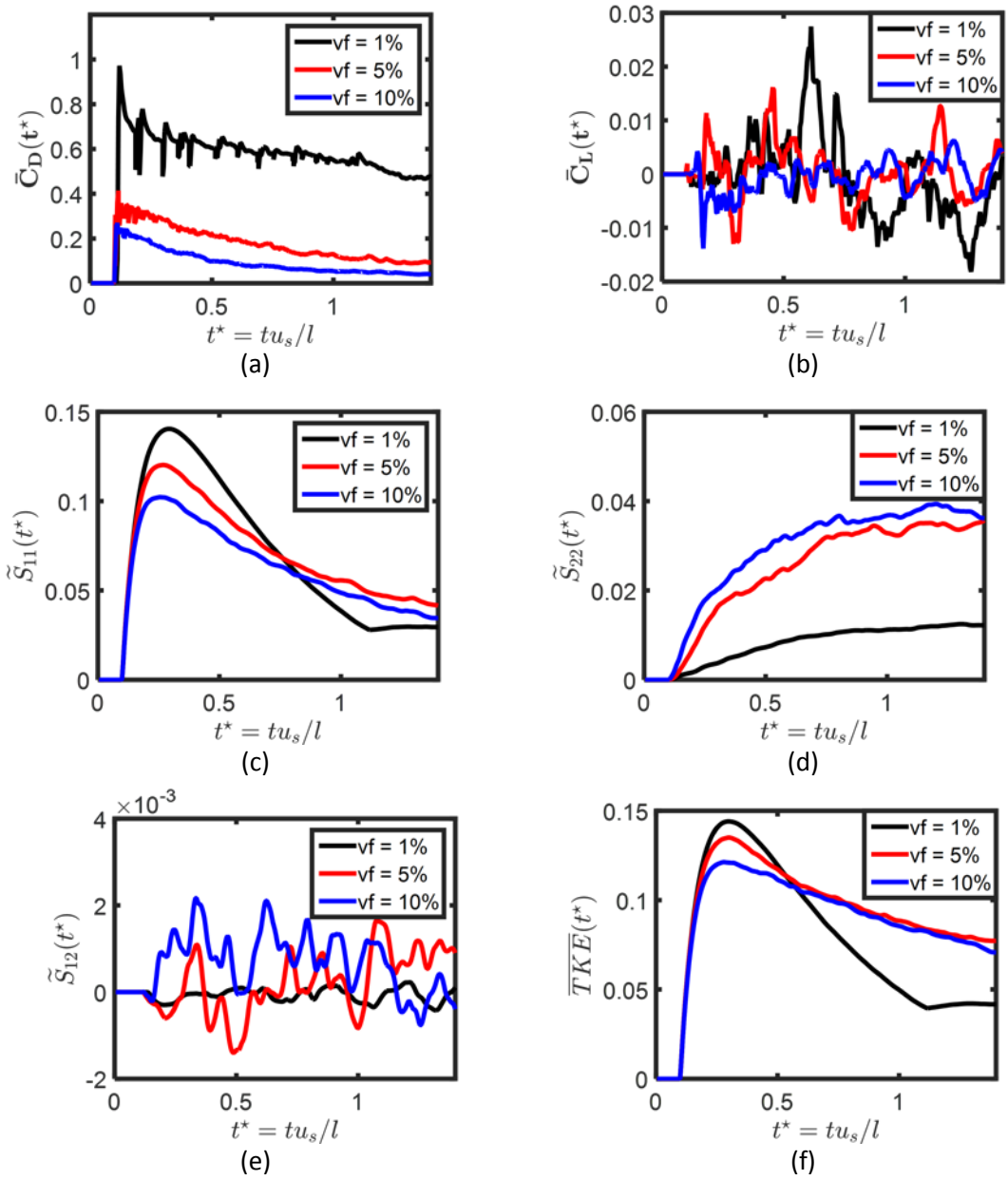


Figure 41 : The variation of the closure terms, viz (a) $C_D(t^*)$ (b) $C_L(t^*)$ (c) $\widetilde{u'u'}(t^*)$(d $\widetilde{v'v'}(t^*)$, (e) $\widetilde{u'v'}(t^*)$ and (f) $TKE(t^*)$ with time. for a shock of Mach Number, Ma = 3.5 interacting with particles clusters of ϕ = 1%, 5% and 10%.

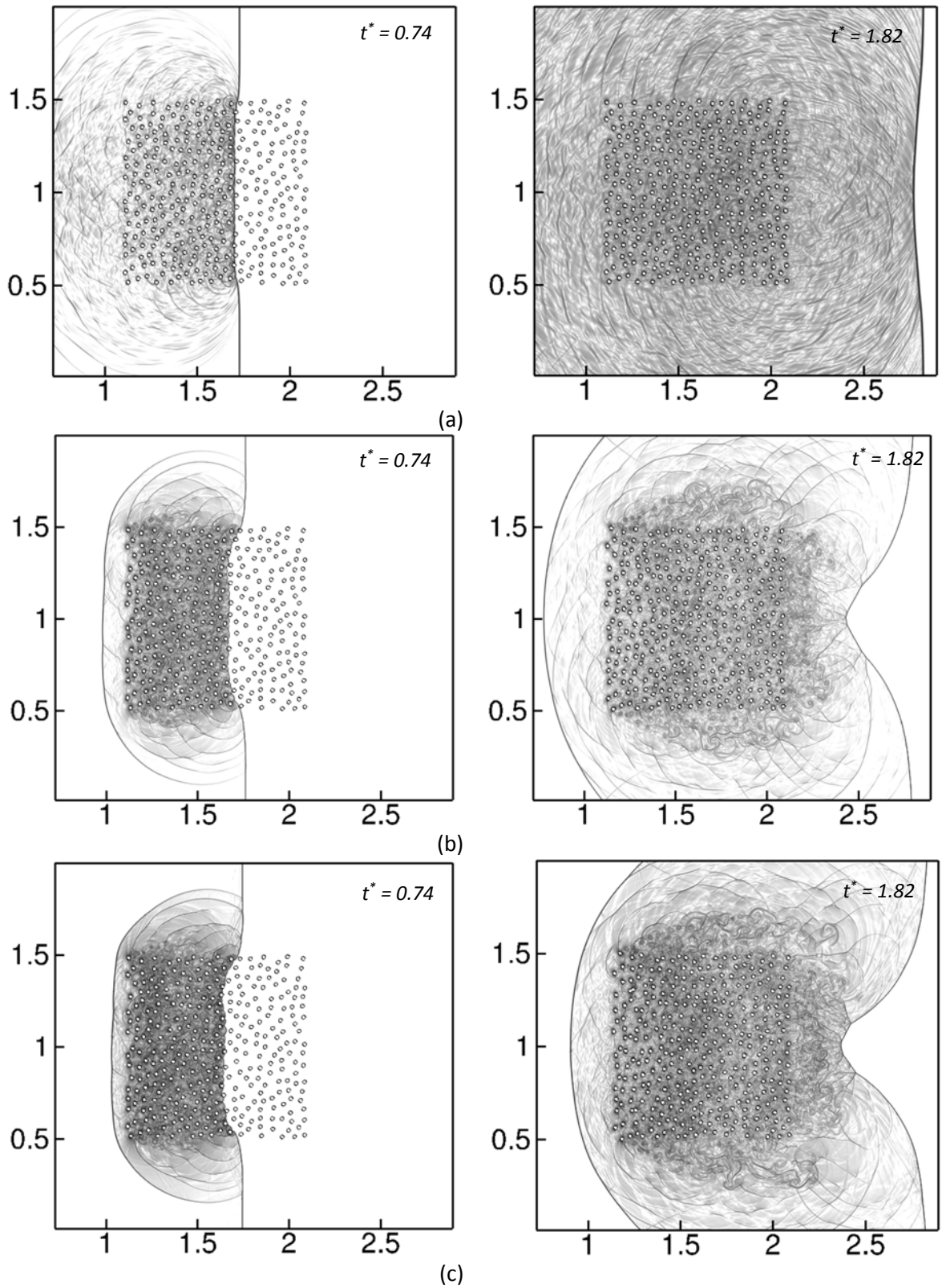


Figure 42 : Numerical Schlieren images of the interaction of a shock of Mach Number Ma with a cluster of particles of volume fraction ϕ at $t^* = 0.74$ and $t^* = 1.82$.(a) [Ma, ϕ] = [1.1,0.1], (b) [Ma, ϕ] = [2.3,0.1] and (c) [Ma, ϕ] = [3.5,0.1].

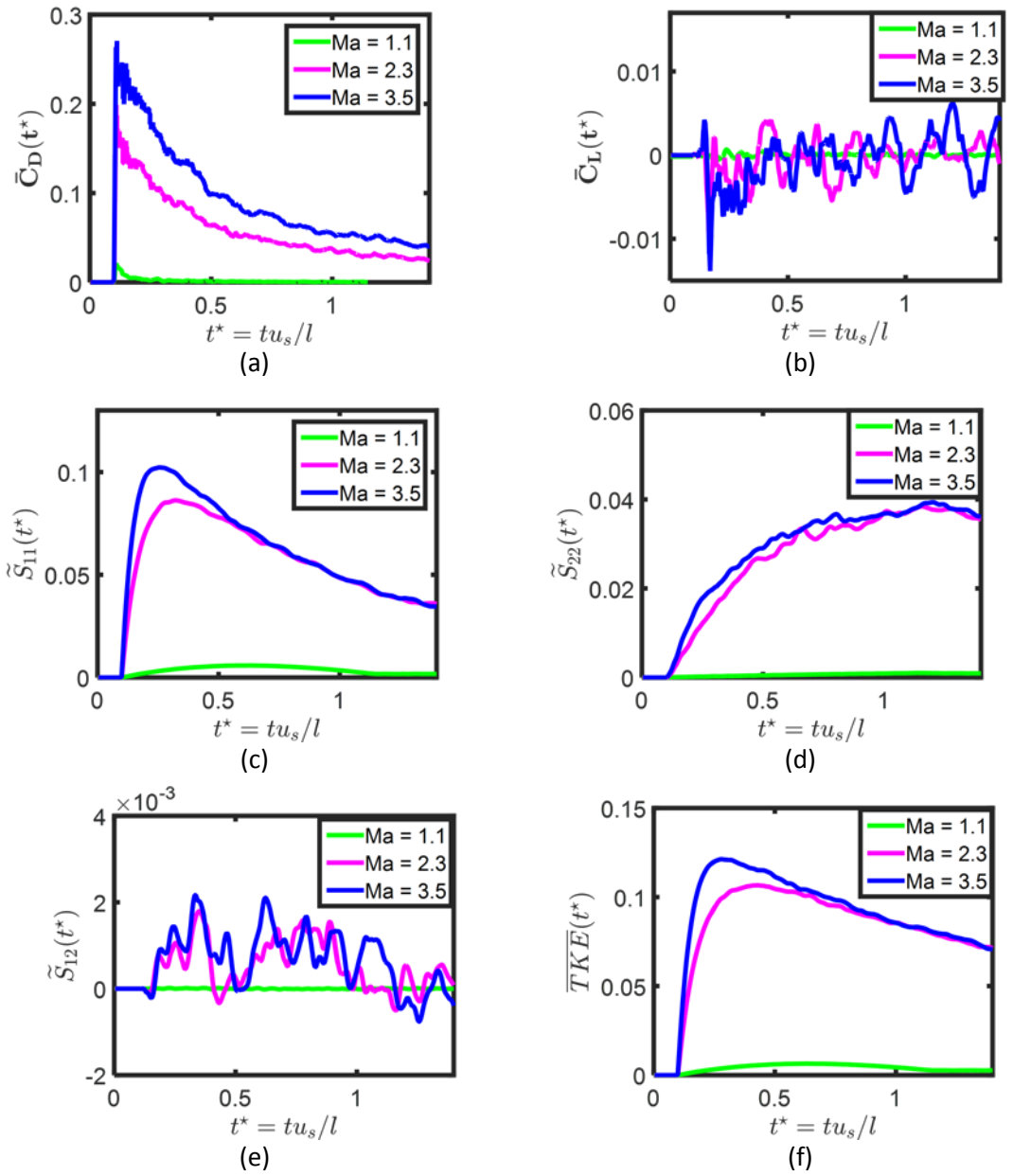


Figure 43 : The variation of the closure terms, viz (a) $C_D(t^*)$ (b) $C_L(t^*)$ (c) $\widetilde{u'u'}(t^*)$(d $\widetilde{v'v'}(t^*)$, (e) $\widetilde{u'v'}(t^*)$ and (f) $TKE(t^*)$ with time. for a shock of Mach Number, *Ma* = 3.5 interacting with particles clusters of ϕ = 1%, 5% and 10%.

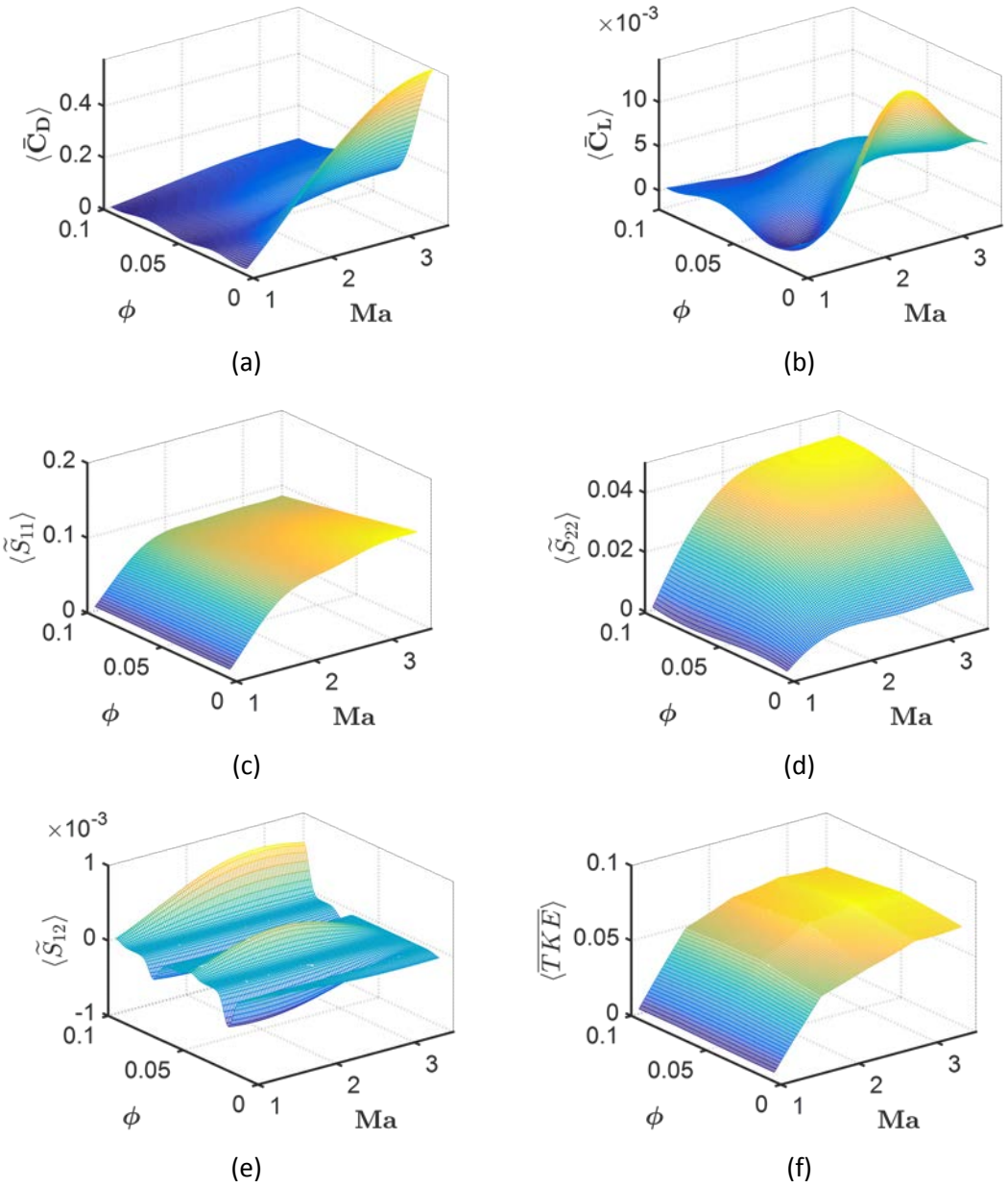


Figure 44 : Surface plots of the variation of the closure terms, viz (a) $\widetilde{C_D}(Ma, \phi)$ (b) $\widetilde{C_L}(Ma, \phi)$ (c) $\langle \widetilde{u'u'} \rangle$ (d) $\langle \widetilde{v'v'} \rangle$, (e) $\langle \widetilde{u'v'} \rangle$ and (f) $\langle \widetilde{TKE} \rangle$ with *Ma* and ϕ . The surface plots are computed using the MBKG method at the input locations shown in Figure (Fig No). Because of the overall symmetry of the computational set up, the magnitude of $\widetilde{C_L}(Ma, \phi)$ and $\langle \widetilde{u'v'} \rangle$ is negligibly small.is negligibly small.

CHAPTER 5. EVALUATION OF VARIABLE FIDELITY BASED SURROAGATE MODELS FOR MULTISCALE MODELING OF SHOCK PARTICLE INTERACTIONS

5.1 INTRODUCTION

Multiscale problems with distinct meso- and macroscales appear in several important engineering applications. Examples of typical applications include the dynamics of particle-laden gases [1,2], deformation of heterogeneous materials such as bones [3–5], concrete [6,7], heterogeneous explosives[8–11], sediment transport in river beds [12], and meso-scale models of blood flow [13]. In such problems, macroscale computations typically resolve a limited range of length and time scales. The unresolved scales are analyzed separately and coupled to the macroscale using closure laws in homogenized macroscale systems of equations [10,14–16]. With increasing computational power and improved physical models and algorithms, there has been an increasing interest [3,17–21] in constructing closure laws using metamodeling [22–25] techniques. In this approach, ensembles of resolved mesoscale simulations are used to create surrogate models for the closure terms in the macroscale system of equations. While the method allows realistic modeling of closures from high-fidelity mesoscale simulations, a bottleneck of this method is the cost of performing large number of mesoscale simulations for constructing the metamodels [26,27]. This limitation becomes severe as the dimension of the parameter space increases [39], making the computational cost of performing the required number of high-fidelity simulations prohibitively high. In such scenarios, it is useful to investigate techniques which allow the construction of metamodels from limited sets of high-fidelity simulations. This work compares different methods of constructing robust and accurate surrogates using ensembles of relatively inexpensive low resolution mesoscale computations in the context of shocked-particle laden flows. The techniques examined in this paper can be applied generally to multi-scale modeling problems where there is a distinct separation of scales and the macro-scale and meso-scale can be computed using continuum thermo-mechanical descriptions.

5.1.1 Surrogate Based Multiscale Modeling for Shock-Particle Interaction

The specific problem of shocks interacting with particles is an example of the use of expensive mesoscale simulations for constructing metamodels for bridging scales in a multiscale modeling framework. Macro-scale models of shock-particle interactions, such as the Cloud-in-Cell models

[14,28,20,29,30] track the mean trajectory of "computational" particles when impacted by a shock. The computational particles, which are collections of physical particles, are modeled as singular point sources in the flow field and their motion is forced by the drag exerted on them by the surrounding fluid. Resolved mesoscale computations of shock-particle interactions have been used in the past by the authors [19,31–35] to generate surrogate models [22–25] for the drag on particles as functions of shock strength and particle volume fractions [20,19]. Although the mesoscale simulations comprised inviscid two-dimensional computational models, each mesoscale simulation was worth several hours of CPU time even in high-performance computing systems. In general, the computational cost of performing numerical experiments increase exponentially with the number of simulations (Figure 45). Therefore, in the interest of practical computational affordability, it is necessary to keep the number of high-fidelity simulations as low as possible.

The first step in ensuring that surrogates are obtained from sparse sets of high-fidelity simulations is to choose a robust metamodeling technique. The performance of metamodeling techniques under sparse data sets is an active field of research [22,31,36,37]. The authors, in a previous work [31], studied the convergence behavior of several metamodeling techniques for analytical functions and semi-empirical drag laws and observed that the Kriging family of methods required the least number of inputs to construct reliable and accurate surrogates. In particular, for inputs obtained from real-time simulations of shock-particle interactions, it was found that the Modified Bayesian Kriging Method [37] created smooth and monotonically convergent surrogates [38] for drag from the least number of inputs, even when the input data was noisy. However, even with the MBKG method, 56 mesoscale computations were required to construct a reasonably accurate surrogate model of the drag [38]. This was found to be the true when the parameter space in [38] was only two-dimensional. It is well known that the number of experiments increase exponentially with the dimension of the parameter space [39]. Therefore, creating closures based on higher dimensional parameter spaces (e.g. drag as a function of Mach Number, Particle Volume Fraction, Reynolds Number) following the approach in [38] becomes practically impossible [39]. The problem is even more severe if mesoscale computations are based on more expensive computational models (such as viscous, three-dimensional computations). In such scenarios, a robust surrogate modeling technique operating on high-fidelity numerical simulations alone is insufficient to create metamodels from mesoscale computations.

5.1.2 Variable Fidelity Based Surrogate Methods for Multiscale Modeling

Variable-fidelity based surrogate modeling [26,27,40–45] provides an affordable route for constructing metamodels in such scenarios. In this approach, the burden of creating a first approximation of the surrogate model is shifted to low-resolution/coarse grid computations. Coarse-grid computations are relatively inexpensive; several such simulations are therefore affordable to construct an initial surrogate. However, because low resolution computations suffer from high discretization errors, the initial surrogate has low fidelity. The low-fidelity surrogate is subsequently corrected using only a few high fidelity computations. Because high-fidelity simulations are limited only for correcting a low-fidelity surrogate model, variable-fidelity based surrogate methods requires fewer resolved simulations compared to high-fidelity surrogate modeling techniques. This makes variable-fidelity based surrogates modeling methods an attractive choice for metamodeling in cases where the cost of constructing high fidelity surrogates becomes prohibitively large.

In the past, variable-fidelity based methods have been limited to surrogate-based design optimization problems [26,27,40–45]. Several variable-fidelity techniques, such as Space Mapping [26,27], SPRM [41], co-kriging based methods [46–48] have been used in the past to create surrogates for objective functions in optimization studies. However, the use of these methods for creating surrogates in the specific context of bridging scales in multiscale modeling problems remains to be studied. The requirements of the two classes of problems differ from each other. In case of design optimization problems, an optimum design is based on the gradients and the locations of the extrema of the surrogate model in the parameter space – rather than the actual values of the (surrogate-based) objective function [26,45–47,49]. In contrast, in surrogate-based multiscale methods, the surrogate model serves as numerical closure laws in macro-scale computations. Therefore, it is not the gradient (shape) or the location of extrema of the surrogate in the parameter space, but the actual values of the surrogate model at all points in the parameter space that are significant in multiscale modeling problems. In other words, the criteria for a variable-fidelity method to be appropriate for creating surrogates is more stringent in multiscale modeling problems. Evaluating the suitability of the framework of variable-fidelity based surrogate modelling in multiscale problems remains to be explored.

In the above context, it is necessary to quantify the rates of convergence of different variable-fidelity methods for surrogates constructed from numerical experiments. In the past, there has been limited research on comparing the rates of convergence of different variable-fidelity techniques even for design optimization problems. However because high-fidelity simulations are expensive, to ensure that no high-fidelity simulation is wasted, it is important to study the rates of convergence of different variable-fidelity techniques. A method with a higher rate of convergence ensures better accuracy at the expense of lesser high-fidelity simulations and is more suitable for multiscale modeling problems than methods with lower rates of convergence.

A related issue which has received limited attention in the past is the definition of low-fidelity computations itself. In other words, can a variable fidelity method correct any low-fidelity surrogate model or is there a limit on the grid resolution below which obtaining a corrected surrogate is impossible? This issue is important because the coarser the mesh size is, the lesser is the compute time required for generating an initial surrogate. As an example, for particle laden flows, an ensemble of 56 mesoscale simulations take approximately half the time if each cylinder is discretized by two mesh points across the diameter as opposed to four mesh points (Figure 45). This motivates the comparison of surrogates obtained from different levels of fidelity, even among low-fidelity simulations.

5.1.3 Evaluation of Different Correction Methods to Create a Robust Variable-fidelity Based Surrogate Model for Drag on Particles interacting with Shocks

The present paper evaluates the suitability of different variable fidelity techniques for use in multiscale modeling problems. In the present paper, three different correction techniques - Space Mapping [26,27] , Radial Basis Function Networks (RBFs) [50–54] and the Modified Bayesian Kriging method (MBKG) [37] – are evaluated based on their ability to construct reliable and accurate surrogates for drag in particles interacting with shock. Each of these methods represent state-of the-art correction techniques/fitting methods which can be suitably employed to create variable-fidelity surrogates from mesoscale computations. Two definitions of low-fidelity computations of shock-particle interactions are considered. In the first case, cylinders are discretized by 2 mesh points across the diameter, while in the second case, they are discretized by 4 mesh points across the diameter. The three aforementioned methods are used to correct both the

low-fidelity surrogates. The error of these methods are evaluated by comparing the values of the variable-fidelity surrogates with those of high-fidelity surrogates. The rates of convergence of these methods with respect to the number of high-fidelity computations needed are studied and the compute time for constructing a surrogate with a given error value is evaluated. The methods are then compared with one other to ascertain their relative advantages and disadvantages for creating variable-fidelity surrogates for drag in particle laden flows.

The rest of the paper is organized as follows. In Section 5.2.1 , the governing equations and the numerical algorithms for performing the mesoscale computations are described. In Section 5.2.2 , the MBKG method used for constructing a surrogate model is summarized. The different techniques for obtaining a variable-fidelity surrogate model by correcting low-fidelity surrogates are described in Section 5.2.3 . The step-by-step procedure for constructing variable surrogates are discussed in Section 5.3 . Variable-fidelity surrogates are compared with high-fidelity surrogates and the rates of convergence and computational times of different variable-fidelity methods are shown in Section 5.4 . Relevant inferences and possible directions of work for future are discussed in Section 5.5 .

5.2 GOVERNING EQUATIONS AND NUMERICAL METHODS

5.2.1 Governing Equation and Numerical Method for Mesoscale Computations

5.2.1. 1 The Governing Equations

In the mesoscale computations, the gas flow is modeled by the compressible Euler equations:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_i)}{\partial x_i} &= 0 \\ \frac{\partial (\rho u_i)}{\partial t} + \frac{\partial (\rho u_i u_j + p\delta_{ij})}{\partial x_j} &= 0 \\ \frac{\partial (\rho E)}{\partial t} + \frac{\partial (\rho u_i E + p u_i)}{\partial x_i} &= 0 \end{aligned} \tag{64}$$

where ρ, u_i, p are the density, velocity components and the pressure of the fluid respectively, while $E = e + 1/2 u_i u_i$ and e are the specific total internal energy and the specific internal energy of the fluid. The equations are closed by the ideal gas equation of state given by

$$p = \rho e(\gamma - 1) \tag{65}$$

where the specific heat ratio γ = 1.4. A well tested Eulerian code SCIMITAR3D [55–61] is used to solve Equation (64) and is described in the next section.

5.2.1. 2 Numerical Framework

The governing equations are spatially discretized on a fixed Cartesian mesh using a 3rd order Essentially Non-Oscillatory (ENO) [62] scheme and are marched in time explicitly using a 3rd order Runge-Kutta (RK) scheme. The procedure for tracking the interfaces of the solid particles and the application of the boundary conditions at the interface is summarized as follows.

1. To define embedded objects in the flow a narrow-band level-set [63] method is used; this allows tracking the object interfaces in a sharp manner. The level set field, ϕ_l at any point is the signed normal distance from the l^{th} immersed object with $\phi_l < 0$ inside the immersed object and $\phi_l > 0$ outside. The interface is implicitly determined by the zero level set fields, i.e. $\phi_l = 0$ contour represents the l^{th} immersed boundary.
2. Once the object interfaces are identified by $\phi_l = 0$ contour levels, the no-penetration boundary conditions are applied using a variant of the ghost fluid method (GFM) [64]. GFM relies on the definition of a band of ghost points corresponding to each phase of the interacting materials. The ghost points for the fluid phase are the points lying inside the embedded object (defined by $\phi_l > 0$) and real points are those which are outside.
3. Once these ghost points are identified, the next step is to populate the ghost field for the fluid. The ghost field is obtained by constructing least-squares interpolation of the field variables of real material points and by imposing the appropriate interfacial conditions [60] Once the values of the flow variables (e.g. ρ, u_i, p and E) at the ghost points are populated with the least-squares field, the two-material problem can be converted to two single-material problems consisting of real fields and their corresponding ghost fields. The field variables can be updated in space and time using the ENO and the RK schemes.

The computational algorithm described in this section is used for performing mesoscale numerical experiments to study the interaction of a shock of strength, *Ma* with a particle cluster of volume fraction, $^{\phi}$. The goal of the simulations is to compute a representative mesoscale drag, $\widetilde{F_D}(Ma, ^{\Phi})$

for different values of *Ma* and ϕ. Once $\widetilde{F_D}(Ma, \Phi)$ is computed for several numerical experiments, the next step is to create a surrogate model for $\widetilde{F_D}(Ma, \Phi)$ using a metamodeling technique - the MBKG method - which is described in the next section.

5.2.2 Metamodeling Technique for Construction of Surrogate Models

Metamodeling involves the estimation of an unknown function $f(\mathbf{x})$ which is known only at certain discrete and distinct points $\mathbf{x}_j$ (j=1,N. The set of the known values of the function and their locations, i.e. the set ($\mathbf{x}_j$, $f(\mathbf{x}_j)$) are known as the *inputs* of the surrogate model and are used to construct the unknown function $f(\mathbf{x})$ in the parameter space. The point $\mathbf{x}_0$, where the value of $f(\mathbf{x})$ is of interest, is called the "*probe point*" of the surrogate model, while the value of the function at the probe point, $f(\mathbf{x}_0)$ is called the *output* of the surrogate model.

5.2.2. 1 The Modified Bayesian Kriging Method

The MBKG method [37,65] assumes the inputs come from a stationary Gaussian random process, with a mean value of $\mathbf{P\lambda} + \mathbf{Z}$ and variance $\sigma^2\beta$, i.e.

$$\tilde{f}(\mathbf{x_0}) \sim MVN\left(\mathbf{P\lambda} + \mathbf{Z}, \sigma^2\beta\mathbf{I}\right) \tag{66}$$

where $\mathbf{P\lambda}$ represents the mean structure, and $\mathbf{Z}$ is modeled as a Gaussian random process with zero mean and covariance $E[Z(\mathbf{x}_j)Z(\mathbf{x}_q)] = \sigma^2\mathbf{R}$. The matrix, $\mathbf{R}$ is a spatial correlation of the input points, and is defined as follows.

$$\mathbf{R} = R_{jq} = R(\theta, \mathbf{x}_j, \mathbf{x}_q) = \prod_{k=1}^{n} \gamma_k(\theta_k, d_k) \tag{67}$$

where θ is a shape parameter, $\mathbf{d}_k = (x_{kj} - x_{kq})$, k = 1,2,…,n, n being the dimension of the vector $\mathbf{x}$. Commonly used models of the correlation functions are listed in Table 4 [66]. The unknown parameters in the MBKG model are $\boldsymbol{\lambda}$, σ^2, $\boldsymbol{\theta}$ and β.

Correlation Function	$\gamma_k(\theta, d_k)$
Exponential	$\exp(-\theta_k \lvert d_k \rvert)$
General Exponential	$\exp\left(-\theta_k \lvert d_k \rvert^{\theta_{n+1}}\right);\ 0 < \theta_{n+1} \le 2$
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	$1-3\xi_k^{\,2} + 2\xi_k^{\,3},\ \xi_k = \min\{1, \theta_k \lvert d_k \rvert\}$
Spline	$1-1.5\xi_k^{\,2} + 30\xi_k^{\,3}$ for $0 \le \xi_k \le 0.2$ $1.25(1-\xi_k)^3$ for $0.2 < \xi_k < 1;\ \xi_k = \theta_k \lvert d_k \rvert$ 0 for $\xi_k \ge 1$

Table 5 : Common Correlation Models used in the Kriging Family of Methods

To determine the unknown parameters, the MBKG method considers a probability distribution function over the weight space, representing the relative degrees of belief in different values for the model parameters [39]. The function is initially set to a prior distribution and the observations i.e., the input data are used to convert the prior distribution to a posterior distribution using Bayes' theorem, which is given by Equation (34).

$$f(\text{Parameter}|\text{Data}) \propto f(\text{Parameter}) \times f(\text{Data}|\text{Parameter}) \tag{68}$$

Here, $f(\text{Data}|\text{Parameter})$ and $f(\text{Parameter}|\text{Data})$ are the prior and posterior probability distributions respectively and $f(\text{Parameter})$ is the likelihood. The mean of the posterior is used to obtain a point estimate for the unknown parameter value, while the probability intervals of the posterior is used to define credible sets, which are believed to contain the true parameter value with a specified probability.

Priors may be chosen such that they are conjugate priors wherever possible, i.e., the prior distribution may be from a parametric family that takes on the same functional form as the likelihood function. In cases where a conjugate prior does not exist for a given problem, any distribution that reflects the prior knowledge about the unknown parameters may be used as priors.

In such cases, Bayes' rule is applied using the Markov Chain Monte Carlo (MCMC) to draw samples from high-dimensional and non-standard probability distributions.

For the three unknown model parameters, $\boldsymbol{\lambda}$, σ^2 and β, semi-conjugate prior distributions are used to fit the MBKG surrogate model. For the parameter $\boldsymbol{\lambda}$, the conjugate prior is a multivariate normal distribution and is expressed as

$$\boldsymbol{\lambda} \sim MVN\left(\boldsymbol{\mu}_{\lambda}, \boldsymbol{\Sigma}_{\lambda}\right) \tag{69}$$

where $\boldsymbol{\mu}_{\lambda}$ and $\boldsymbol{\Sigma}_{\lambda}$ are the prior mean vector and the covariance matrix for the distribution. Similarly, the conjugate prior distribution for σ^2 and θ are Inverse-Gamma distributions expressed as

$$\sigma^2 \sim \text{Inverse Gamma}\left(\alpha_{\sigma}, \beta_{\sigma}\right) \tag{70}$$

$$\beta \sim \text{Inverse Gamma}\left(\alpha_{\beta}, \beta_{\beta}\right) \tag{71}$$

where α_{σ}, β_{σ}, α_{β}, β_{β} are the prior parameters of the respective distributions. The parameters $\boldsymbol{\theta}$ are embedded in the correlation matrix, $\mathbf{R}$, and there is no known conjugate distribution type that may be used as prior distributions. The prior distribution for $\boldsymbol{\theta}$ is chosen to be a uniform distribution and is expressed as

$$\theta_j \sim \text{U}\left(a_{\theta_j}, b_{\theta_j}\right) \tag{72}$$

where θ_j is the j^{th} correlation function parameter and $a_{\theta j}$ and $b_{\theta j}$ are the prior parameters for θ_j. Using Equations (35) through (38), the joint posterior distribution for the MBKG formulation given in Equation (34) may be expressed as

$$\begin{aligned} f\left(\boldsymbol{\lambda}, \sigma^2, \boldsymbol{\theta}, \beta, \mathbf{R} \middle| f(\mathbf{x})\right) \propto & \prod_{j=1}^{k}\left[\text{U}\left(a_{\theta_j}, b_{\theta_j}\right)\right] \times MVN\left(\boldsymbol{\mu}_{\lambda}, \boldsymbol{\Sigma}_{\lambda}\right) \times \\ & \text{Inverse Gamma}\left(\alpha_{\sigma}, \beta_{\sigma}\right) \times \text{Inverse Gamma}\left(\alpha_{\beta}, \beta_{\beta}\right) \\ & \times MVN\left(\mathbf{0}, \sigma^2 \mathbf{R}\right) \times MVN\left(\mathbf{P}\boldsymbol{\lambda} + \mathbf{Z}, \sigma^2 \beta \mathbf{I}\right) \end{aligned} \tag{73}$$

The full conditionals for each of the model parameters, shown in [37], may be used to estimate the values of the model parameter for each MCMC sample. Because the MBKG method is evaluated using posterior distributions, the MBKG surrogate model is not a deterministic surrogate model

but rather a surrogate that produces posterior distributions for the MBKG parameters. Therefore, a predicted response value does not have one deterministic value but rather has a distribution that gives the probability of the predicted response value being in any interval. The MCMC samples drawn from the predictive distribution of the response variable may be used to estimate any desired characteristics of the distribution, e.g., the mean, standard deviation and credible sets. The larger the standard deviation and the wider the credible sets, the more uncertainty there is in the predicted value. The credible sets come closer to the mean value with larger number of samples, i.e. the uncertainty in the true parameter value reduce remarkably with increasing values of *N*, the number of samples. Unless otherwise mentioned, in the current work, the term MBKG hypersurface/ MBKG surrogate model is used to denote the *mean response* of the probability distribution of the predictions. In the present work, the MBKG method is used to create surrogates for drag from numerical experiments of a given fidelity.

5.2.3 Methods for Correction of Low Fidelity Surrogate Models

A variable-fidelity surrogate modeling technique starts out with a low-fidelity surrogate model constructed from N_c coarse mesh computations using the MBKG method. The initial surrogate is then corrected by performing a few (say N_f) high-fidelity computations. There are several methods of correcting the initial hypersurface by comparing the outputs of the low-fidelity and the high-fidelity computations at points in the parameter space [26,27,40–45]. In the present paper, the following three methods are used to correct the low-fidelity hypersurface using high-fidelity corrections/simulations.

5.2.3. 1 Space Mapping

The space mapping (output space mapping) is a computationally inexpensive method of correcting a low-fidelity surrogate model used commonly is surrogate-based design optimization problems [26,27,40,41]. The surrogate correction is based on an affine transformation on the low-fidelity surrogate model to match the high-fidelity simulation outputs. If $f^c(\mathbf{x})$ and $f^f(\mathbf{x})$ are the outputs of the low-fidelity and high-fidelity surrogate models respectively at a location $\mathbf{x}$ in the parameter space, then the space mapping method corrects the low-fidelity responses using the following model.

$$f^f(\mathbf{x}) = af^c(\mathbf{x}) + b \tag{74}$$

where *a* and *b* are scalar parameters. The parameters *a* and *b* are estimated by fitting the values of the low-fidelity surrogate model $f_i^c(\mathbf{x})$ to the high-fidelity surrogate model $f_i^f(\mathbf{x})$ at *i = 1,2,…*, N_f locations in the parameter space in a least squares sense. Tthe parameters are determined by minimizing the functional Ψ with respect to *a* and *b*, where Ψ is defined as follows.

$$\Psi = (f^f(\mathbf{x}) - af^c(\mathbf{x}) - b)^2 \tag{75}$$

Once *a* and *b* are estimated, a low-fidelity response can be corrected using Equation (74) to obtain an estimate of the high-fidelity response at that location. Because in this method, the parameters *a* and *b* depend only on the responses $f_i^c(\mathbf{x})$ and $f_i^f(\mathbf{x})$, the estimation of the parameters does not depend on the dimensionality of the parameter space, making the correction method robust for surrogates constructed on higher dimensional parameter spaces.

5.2.3. 2 Radial Basis Function Networks

A Radial Basis Function Network is a fitting method, which uses a (finite) set of identical basis Radial Basis Functions (RBF) centered on several distinct points in the input space to estimate an unknown function, *f*(**x**). The estimated value of the function at a point $\mathbf{x_0}$ is given by [50,52,53,67,68]

$$\tilde{f}(\mathbf{x_0}) = \sum_{k=1}^{M} \lambda_k \phi(\mathbf{x_0}, \mathbf{x}_{c_k}; \theta_{c_k}) \tag{76}$$

where λ_k is the weight associated with each of the basis functions $\phi(\mathbf{x_0}, \mathbf{x}_{c_k}; \theta_{c_k})$, 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 $\phi(\mathbf{x_0}, \mathbf{x}_{c_k}; \theta_{c_k}) = \exp\left(-\frac{d^2_{ck}}{\theta^2_{ck}}\right)$ where $d_{ck} = \left\|\mathbf{x_0} - \mathbf{x}_{c_k}\right\|$, is the Euclidean Norm. The weights λ_k are given by solving the least-squares problem

$$\boldsymbol{\lambda} = \mathbf{H}^{\perp}\mathbf{f} \tag{77}$$

where $\mathbf{H}=H_{jk} = \phi\left(\left\|\mathbf{x}_i - \mathbf{x}_{c_k}\right\|;\theta_{c_k}\right)$ j =1, 2, ..., N, k =1, 2, ..., M, with *N* being the number of inputs and *M* the number of Gaussians used. The determination of optimal parameters, *M*, $\mathbf{x}_{ck}$ and θ_{ck} of an RBFANN is a subject of active research [7,50,52,67,69]. In the current approach the parameters are determined as follows.

1. Determination of the number of basis functions, *M*: To avoid exact interpolation or over-fitting, the number of RBFs is chosen to be less than the number of inputs [65]. The number of Gaussians are chosen to be approximately 0.8 times the number of inputs.
2. Determination of the center of the basis function, $\mathbf{x}_{ck}$, *k =1, 2, ..., M*. The RBFs are initially uniformly distributed in the domain; the centers are then updated by a *K*-means clustering algorithm to avoid the possibility of an empty cluster in case of non-uniform inputs.
3. Determination of the shape parameter, θ_{ck}, of the RBFs: The shape parameter is chosen to be equal to the mean distance of an RBF to its three nearest neighboring RBF such that they span the entire domain of the input space.

The RBF method is used to estimate an error, $e(\mathbf{x}) = f^f(\mathbf{x}) - f^c(\mathbf{x})$, where $f^f(\mathbf{x})$ and $f^c(\mathbf{x})$ are the responses from the high-fidelity and low-fidelity computations respectively. In the RBF method, it is assumed that the shape of the low-fidelity hypersurface is different from the high-fidelity hypersurface and the error $e(\mathbf{x})$ is a function of the parameter space, **x –** as opposed to the space mapping method (Section 5.2.3. 1), where the error is not explicitly a function of **x.**

5.2.3. 3 The Modified Bayesian Kriging Method

Similar to Radial Basis Function Networks, the MBKG method (Section 5.2.2. 1) is a fitting method, which is used in the present work to fit the error, $e_i(\mathbf{x}) = f_i^c(\mathbf{x}) - f_i^f(\mathbf{x})$, *i = 1,2,…,*$N_f$ in order to estimate $e(\mathbf{x})$ in the parameter space.

5.3 METHODS FOR CONSTRUCTING VARIABLE FIDELITY SURROGATE MODELS FOR DRAG IN PARTICLES INTERACTING WITH SHOCK

In constructing a variable fidelity surrogate model, several coarse grid computations are performed to create an initial low-fidelity surrogate model. The initial surrogate model is then corrected using very few highly resolved mesoscale computations. In the context of shocked particle laden flows, the construction of a variable fidelity-based surrogate model relies on the following components.

a) A mesoscale computational model for performing coarse grid numerical experiments
b) A metamodeling technique for constructing a low-fidelity surrogate model of the drag from numerical experiments
c) A mesoscale computational model for performing a few high resolution mesoscale experiments
d) A method for correcting the low-fidelity surrogate model using the results of resolved (high-fidelity) mesoscale computations

In the present work, mesoscale computations - both low and high fidelity - are performed using the code SCIMIATR3D, while low/high fidelity surrogates are constructed using the MBKG method. The low fidelity surrogates are corrected using the methods described in Section 5.2.3 . The details of obtaining variable fidelity surrogates is described as follows.

5.3.1 Description of the Mesoscale Computational Experiments

The mesoscale computational model consists of the gas phase and the particle phase. The particles are modeled as rigid cylinders immersed in a flow at various Mach numbers. For sufficiently high fluid velocities and small particles the viscous time scales are larger than the shock propagation time scales and drag on the particles is dominated by pressure forces. Therefore, the Euler equations are solved in the fluid phase. The mesoscale computational domain is illustrated in Figure 22. The setup consists of an array of n particles of radius r= 0.01 units, immersed in a uniform flow with $(u,\rho,p) = (0.0,1.0,1.0)$. The particles are arranged in a square of 1.0×1.0 units and the local particle volume fraction ϕ is given by

$$\phi = 41\pi r^2 \tag{78}$$

To vary the particle volume fraction, the number of particles, the particle radius, *r* is varied to achieve a target volume fraction. The simulation is initiated with an imposed shock of Mach number, *Ma*, located some distance upstream of the particles. The initial thermo-mechanical properties of the shocked gas (air) are computed from the Rankine-Hugoniot jump conditions. Slip boundary conditions are specified at the top and bottom surfaces of the computational domain

while inflow and outflow boundary conditions are maintained at the left and right of the domain respectively.

In the mesoscale computations, the dimensions of the square in which the particle cluster is arranged is selected as the reference length scale, l_{ref} and is set to $l_{ref} = 1.0$. Corresponding to l_{ref}, a reference time scale, t_{ref} is defined as $t_{ref} = l_{ref} / u_s$, such that an incident shock of speed u_s traverses the particle cluster in $t_{ref} = 1$ unit of time. Throughout the paper, the drag forces on particles and other transient quantities are presented as a function of a scaled time t^* which is defined as the ratio of the non-dimensional time t to the reference time t_{ref}, i.e. $t^* = t/ t_{ref}$. Since the shock traverses the particle cluster in time of order $t_{ref} = 1$, the use of the scaled time t^* allows for comparison of the dynamics across the range of parameters (ϕ, *Ma).*

5.3.2 The Parameter Space for Construction of Variable Fidelity Surrogates

In the present work, the drag is assumed to depend on *Ma* and ϕ alone; while the effects of Reynolds' Number, Particle Size Distribution etc are ignored. It is noteworthy that the motivation of the present work is a higher dimensional parameter space. However, for the purpose of developing and testing the efficacy of a variable fidelity-based surrogate modeling framework, a two-dimensional parameter space is considered in the present work. This is because, in a two-dimensional parameter space, it is possible to generate a high-fidelity surrogate model as well as a variable-fidelity surrogate model relatively easily. This allows a direct comparison of the savings in computational cost and the error entailed in a variable-fidelity based surrogate model with surrogates constructed from resolved high-fidelity computations. For this reason, it is assumed that the parameter space in the present work is two-dimensional and the drag in mesoscale computations is a function of *Ma* and ϕ alone – the selection of values of *Ma* and ϕ for performing numerical experiments is discussed in the next section.

5.3.3 Sequential Sampling Method for Selection of Locations of Mesoscale Computations in the Parameter Space

Because mesoscale computations are expensive, numerical experiments can only be performed at selected values (*Ma,* ϕ*)* in the parameter space. To construct an initial Kriging surrogate, a set of initial inputs are selected by the user. The subsequent input points for further computations are typically chosen by the user based on the regions of interest in the parameter space. Examples of sampling strategies include choosing samples at regular intervals, Latin Hypercube techniques

[70], Latin Centroidal Voronoi Tessellation (LCVT) methods [71], etc. A majority of the sampling techniques are unsupervised in nature, i.e. the locations of the subsequent input points in the parameter space are independent of the regions of complexity of the hypersurface to be approximated. Because the MBKG method gives a measure of the uncertainty associated with the fit, it is possible to use the credible sets to identify regions in the parameter space where the error in approximation is higher. This can be used to select locations for subsequent inputs in constructing surrogate models. The locations for performing mesoscale simulations in the parameter space are systematically selected in the following way:

1. First, the limits for the parameter space are specified. Numerical experiments are designed to be conducted for $1.1<Ma<3.5$ and $1.28\%<\phi<10\%$. To start with N = N_1 numerical experiments are conducted at uniformly spaced locations in the parameter space to obtain $\tilde{F}_D(Ma, \phi)$ values. An initial surrogate model for $\tilde{F}_D$ is constructed using the MBKG methods.
2. The value of credible sets obtained from fitting the model is computed at 100x100 locations, uniformly distributed in the parameter space. A new set of N_2 candidate locations with the highest values of the credible sets in the parameter space are selected for performing the next set of numerical experiments. An improved surrogate model is then constructed based on the new values of $\tilde{F}_D$ at all the $N = N_1+N_2$ locations.
3. This process is continued to a desired number of times *k*, each time adding N_k new points until a surrogate model of acceptable accuracy is constructed. Note that the number of points N_k is not fixed *a priori* and the user is free to choose the number of new mesoscale simulations based on the computational resources available.

Surrogate models are constructed for N_k = 13, 29, 45 and 61 mesoscale computations for $k = 1,2,3$ *and 4* respectively. The location of the mesoscale computations at each step *k* are shown in Figure 31.

5.3.4 Homogenization of the Mesoscale Drag for Generating Inputs to the Surrogate Model

Each mesoscale computation at the 56 locations in the parameter space resolves the flow around the 41 cylinders shown in Figure 22. The drag is different for each of the cylinders in the computational domain; but drag correlations that are useful for macroscale computations require a unique value of drag for a given value of *Ma* and ϕ. Following the procedure of our previous

work [38], a spatially-averaged drag, $\overline{F_D}(t^*; Ma, \phi)$ is defined as the average drag over the five central cylinders – viz. cylinders 13, 31, 32, 35 and 36 in Figure 22. The spatially-averaged drag, $\overline{F_D}(t^*; Ma, \Phi)$, is still a function of time; $\overline{F_D}(t^*; Ma, \Phi)$ is averaged over a mesoscale time, $\bar{t}^*$ as follows:

$$\widetilde{F}_D(Ma, \phi) = \frac{\int_{t^*=0}^{t^*=\bar{t}^*} \overline{F}_D(t^*) dt^*}{\int_{t^*=0}^{t^*=\bar{t}^*} dt^*} \tag{79}$$

where $\tilde{F}_D(Ma, \phi)$ denotes the drag averaged over both space and time. One key issue in obtaining the time-averaged drag is the selection of the time sample, i.e. the value of $\bar{t}^*$. Following the convention of our previous work [38], $\bar{t}^*$ is selected to be the time when the incident shock system completely leaves the particle cluster. A spatio-temporally averaged drag on a representative particle $\tilde{F}_D(Ma, \phi)$ is obtained for different values of *Ma* and ϕ and is used for constructing surrogates for the drag using the MBKG method.

5.3.5 Low Fidelity Mesh Resolutions for Construction of Surrogate Models

To construct an initial low-fidelity surrogate model, coarse grid computations are performed at N_c = 56 aforementioned locations in the parameter space – i.e. N_c = 56 sets of $\tilde{F}_D(Ma, \phi)$ are used as inputs for the MBKG method. In the present work, two sets of coarse grid computations are adopted. The coarsest grid size, named as Mesh A, corresponds to $r/\Delta x = 2$ and is the first low-fidelity computational model used in the present work. A slightly more refined grid resolution – that corresponding to $r/\Delta x = 4$ - is used for constructing another low-fidelity surrogates and is hereafter referred to as Mesh B. Each Mesh A and Mesh B computations independently result in two sets of low fidelity surrogate models. The next step is to correct the low-fidelity surrogates so constructed using high fidelity simulations.

5.3.6 Procedure for Selecting Locations for High-Fidelity Computations to Construct Variable Fidelity Surrogates

High-fidelity computations are expensive; the N_h high fidelity computations must be judiciously selected to avoid wastage of computational effort. Typically, a subset of the input locations of the low-fidelity experiments are chosen for performing high-fidelity computations [26,41]. Because $N_c > N_f$, there are different ways of selecting the locations for performing the N_f high-fidelity computations – the Kaufmann algorithm is a method that allows the selection of a subset comprising N_f points from a set of N_c points by ensuring that the selected subset is widely distributed in the parameter space. The pseudocode of the algorithm, as implemented in the current work, is as follows.

1. To ensure that the high fidelity input points span the entire parameter space, the first input locations are selected at the corners of the parameter space. Thus, for a parameter space of dimension n, select the first 2^n+1 high-fidelity sample locations at the 2^n corners of the parameter space and the remaining one located closest to the centroid of the parameter space.
2. For every non-selected input location, $\mathbf{x}_i$ ($i = 1,2,\ldots, n_c$) do
    a. For every non-selected input location, $\mathbf{x}_j$ ($i = 1,2,\ldots, n_c$) do
    Calculate $C_{ij} = \max(D_j - d_{ji}, 0)$ where $d_{ji} = \| \mathbf{x}_i - \mathbf{x}_j \|$ and $D_j = \min_s d_{sj}$, s being one of the selected inputs
    b. Calculate the sum of the distances, $\sum_j C_{ji}$
3. Select the unselected input location $\mathbf{x}_i$ which maximizes $\sum_j C_{ji}$
4. If the number of selected inputs reaches the target value n_h, then stop, else go to step 2.

The sequence of the N_f high fidelity computational locations, as determined using the Kaufmann Algorithm, is shown in Figure 50. Thus, if $N_f = 15$, the points marked as 1,2,3,…,15 would be the locations of performing the 15 Mesh C computations. Once the locations of performing the high-fidelity computations are selected, the next step is to use these results to correct the low fidelity hypersurface to obtain a variable-fidelity hypersurface.

5.3.7 Construction of Variable Fidelity Surrogate Models form Low Fidelity Surrogates

The low-fidelity surrogates are corrected using N_f high-fidelity computations using the techniques described in Section 5.2.3 .

The Space Mapping technique is used to perform an affine transformation of the low-fidelity surrogate to obtain a least-squares estimate of the high fidelity surrogate using Equation (74).

The RBF method is used to estimate an error, $e_i(\mathbf{x}) = f_i^c(\mathbf{x}) - f_i^f(\mathbf{x})$, where $f_i^c(\mathbf{x}), f_i^f(\mathbf{x})$ are the responses of the high and low-fidelity surrogates. To estimate $e(\mathbf{x})$, RBFs are fitted to the error $e_i(\mathbf{x}) = f_i^c(\mathbf{x}) - f_i^f(\mathbf{x})$ at $i = 1,2,\ldots,N_f$ points in the parameter space.

Similar to RBFs, the MBKG method (Section 5.2.2. 1) is a fitting method, which is used in the present work to fit the error, $e_i(\mathbf{x}) = f_i^c(\mathbf{x}) - f_i^f(\mathbf{x})$, $i = 1,2,\ldots,N_f$ in order to estimate $e(\mathbf{x})$ in the parameter space. Each of the three methods are used to create variable fidelity surrogates from the Mesh A and Mesh B low fidelity surrogates.

5.3.8 Evaluation of the Errors and Compute Times in Variable Fidelity Surrogate Modeling

To compare the surrogates created using Mesh A and Mesh B computations with a high-fidelity surrogate, a third hypersurface is created using high fidelity computations alone. High fidelity computations are performed using a mesh size corresponding to $r/\Delta x > 8$ and is hereafter referred to as Mesh C computations. In general, obtaining a high fidelity surrogate model from Mesh C computations would be almost impossible; in the present work, because the parameter space is 2D and the mesoscale computations are also two-dimensional and inviscid, a direct comparison of high and low-fidelity surrogates is affordable and is used to ascertain the effectiveness of a variable-fidelity surrogate model.

For each of Mesh A and Mesh B low fidelity surrogates, the three methods described in Section 5.2.3 result in three different sets of variable fidelity hypersurfaces for a given value of N_f. Which method results in the "best" hypersurface for low values of N_f ? This requires measures for quantifying the error in the variable fidelity surrogates so created as a function of N_f.

To compare the error involved in creating variable-fidelity surrogates, the low-fidelity surrogate model after correction by each of the three methods described in Section 5.2.3 is probed at M = 100x100 points in the parameter space. These values are then compared with the value obtained

by probing the high-fidelity surrogate at the same points in the parameter space. A normalized L^2 error, ε_{L^2}, is defined as follows.

$$\varepsilon_{L^2} = \frac{\sum_{i=1}^{M}\left(\tilde{F}_D^{VF}(Ma_i, \phi_i) - \tilde{F}_D^{HF}(Ma_i, \phi_i)\right)^2}{\sum_{i=1}^{M}\left(\tilde{F}_D^{HF}(Ma_i, \phi_i)\right)^2} \tag{80}$$

where $\tilde{F}_D^{VF}(Ma_i, \phi_i), \tilde{F}_D^{HF}(Ma_i, \phi_i)$ denotes the values of $\tilde{\mathrm{F}}_{\mathrm{D}}(\mathrm{Ma}, \phi)$ at the point (Ma_i, ϕ_i) in the parameter space as obtained from the variable fidelity hypersurface and the high fidelity hypersurface respectively.

If the correction methods are monotonically convergent, the error ε_{L^2} is expected to decrease for increasing N_f. However, adopting larger value of N_f is impractical because of the increasing computational cost. Typically, one is interested in a balance between the computational cost and the error entailed in using a value of N_f. For this reason, a cost function, Ψ, is defined as follows.

$$\Psi(N_f) = \frac{1}{2}\left(\frac{\varepsilon_{L^2}}{\widehat{\varepsilon_{L^2}}} + \frac{T}{\hat{T}}\right) \tag{81}$$

Here, ε_{L^2} is the normalized L^2 error defined in Equation (80), T is the compute time, and $\widehat{\varepsilon_{L^2}}$ and $\hat{T}$ are normalization quantities chosen so that each of the terms in the right hand side of Equation (4) has an order of magnitude of 1. In the present work, because Mesh A is the coarsest approximation, the Mesh A low-fidelity surrogate is expected to have the highest value of L^2 error. Therefore, $\widehat{\varepsilon_{L^2}}$ is chosen to be the value of ε_{L^2} for N_f = 0 corresponding to Mesh A computations so that the supremum of the first term in the right hand side of Equation (4) is of the order of magnitude 1. Similarly, because Mesh C computations are most expensive (in comparison to Mesh A and Mesh B), the high-fidelity Mesh C surrogate is expected to have the highest value of T. Therefore $\hat{T}$ is set to the compute time of creating Mesh C hypersurface using 56 high fidelity simulations. The parameter, Ψ, thus helps in quantifying the relative contributions of error in approximation and the cost of constructing surrogates from N_f high fidelity simulations.

The above procedure can be applied to generate variable fidelity surrogates for Mesh A and Mesh B; in the next section, the surrogates are presented and compared for each of the three correction methods.

5.4 RESULTS AND DISCUSSIONS

The first step in construction of variable-fidelity surrogate models involve performing mesoscale numerical experiments to compute the drag on particle-clusters interacting with a shock. The mesoscale numerical algorithm is first validated by comparing the results with the case of a shock interacting with a single cylinder in Section 5.4.1 , while the results of the variable-fidelity surrogate modeling techniques are described in Section 5.4.2 .

5.4.1 Validation of the Computational Code: Interaction of a Shock with a Single Cylinder

To validate the calculations of SCIMITAR3D [55–61] in the present context the interaction of a shock with a single cylinder is studied. The diameter of the cylinder is selected to be 0.1 non-dimensional units and the center of the cylinder is placed at (1.1,1.0) inside a domain of size 3.0 x 1.0. A planar shock of Ma = 2.6 is initially placed at $x_1 = 1.0$. Slip boundary conditions are enforced at the top and bottom faces, while inflow and outflow conditions are maintained at the west and east faces of the domain respectively. The study is conducted for different mesh sizes of $\Delta x = \Delta y =$ 0.004, 0.0025, 0.001, 0.0009 and 0.00075.

The drag on a cylinder is compared with the calculations of [72]. Figure 23 shows that the peak drag coefficient agrees well with the benchmark [72]. The initial part of the drag corresponds to the pressure forces on the particle resulting from the passage of the shock and is in good agreement. The disagreement in the decreasing part of the drag curve is because the present calculations are inviscid while [72] used the Navier-Stokes equations. To further validate the computational model, the case of a shock of Ma=2.81 interacting with a cylinder is studied. The locus of the upper and lower triple-points of the shock-system created after the onset of Mach reflections are compared with experimental observations of [73] and calculations of [72]. Figure 24 shows that the current calculations are in excellent agreement with the experimental observations as well as the calculations of [72]. While more extensive validation of the code SCIMITAR3D is shown in [55–61], the two problems chosen in this section validate the computations in the context of shock interactions with particles. Mesoscale numerical experiments of shock particle interactions are performed following the procedure outlined in Section 5.3 and is described next.

5.4.2 Variable Fidelity-Based Surrogate Models for Constructing Numerical Drag Law as a Function of Ma and ϕ

Variable fidelity models use a low resolution mesh in combination with a high resolution mesh to create surrogates. Before comparing the surrogates obtained from variable-fidelity methods with that from high-fidelity methods, it is useful to briefly see the effect of different mesh sizes on the local flow features in mesoscale computations.

5.4.2. 1 Comparison of the Flow Fields and the Drag Values for Different Mesh Sizes for Ma = 3.5 and ϕ = 1%.

Figure 51 shows representative computations for a shock of Ma = 3.5 interacting with a particle-cluster of ϕ = 1%. Because of the coarseness of the mesh, in Mesh A computations, the shape of the cylinders are inadequately represented (Figure 51a); some cylinders do not confirm to a circular shape in the flow domain. This causes sharp corners in the immersed geometry and slightly alters the flow field in Mesh A computations (Figure 51a) compared to the high fidelity computations Figure 51c).

In addition to altering the flow profile resulting from spurious sharp corners in low resolution computations, the shock features - the incident shocks, reflected shocks, slip lines and barotropic vortices are more resolved in Mesh C compared to Mesh A and Mesh B computations. Among Mesh A and Mesh B, the vortices and the shocks are more "diffused" for Mesh A compared to Mesh B.

The coarseness of the mesh not only influences the flow characteristics, but also the temporal drag on particles in mesoscale computations. A comparison of $\overline{F_D}(t^*; Ma, \Phi)$ for Meshes A, B and C (Figure 52) reveals three important features that are of interest in the present work. First, because of higher numerical diffusion, the high-fidelity (Mesh C) drag is underestimated by Mesh B and even more by Mesh A. In other words, the low fidelity computations are negatively biased and Mesh C drag values are a supremum to Mesh A and Mesh B drags. Second, the effect of numerical diffusion is not the same for all values of t^* . Because of this, the aforementioned bias is not constant with t^* - there are values of t^* (e.g at $t^* \sim 0.75$) when the bias is higher compared to other values (e.g at $t^* \sim 1.2$) in Figure 52. Therefore, the mean shape of the high-fidelity drag is different from the mean shape of the low-fidelity drag. Third, in addition to the bias, the high frequency

components of the shape of the low-fidelity drag values ($\overline{F_D}(t^*; Ma, \Phi)$) is also different from the high-fidelity drag values in Figure 52. The drag from Mesh A has least high frequency oscillations, followed by that of Mesh B and finally followed by Mesh C. This is because a finer mesh captures the fine-scale shock interactions in the particle cluster more accurately compared to coarser grids. Therefore in addition to the bias, which is always one-sided and under-approximates the high fidelity drag, the high frequency oscillations produce both a positive and a negative error field which causes the smoothness of low fidelity drags to differ from Mesh C drag (Figure 52).

In summary, there are two major components to the overall error in low fidelity drag – a bias which is a function of time and alters the mean shape of the high fidelity drag and an error which alters the high-frequency components of the shape of high-fidelity drag. The overall discrepancy of the temporal drag, $\overline{F_D}(t^*; Ma, \Phi)$, results in different values of the spatio-temporally homogenized drag $\tilde{F}_D(Ma, \phi)$ for the different mesh sizes - $\tilde{F}_D(Ma, \phi)$ is the lowest for Mesh A, followed by Mesh B and Mesh C (Table 3). To quantify the differences as a function of the parameter space, in the next section, the low-fidelity as well as the variable-fidelity surrogates obtained using Mesh A and Mesh B are compared with the high fidelity surrogate constructed using Mesh C.

5.4.2. 2 Correction of the Low Fidelity Hypersurfaces to obtain a Variable-Fidelity Surrogate Model for $\tilde{F}_D(Ma, \phi)$

Low Fidelity surrogates for drag are constructed individually for both Mesh A and Mesh B computations using the MBKG method (5.2.3. 3). The low-fidelity surrogates are then corrected using N_f high-fidelity computations using the techniques described in Section 5.2.3 . To visualize the surrogates of $\tilde{F}_D(Ma, \phi)$, a set of M = 100x100 probe points, distributed regularly inside the domain, are selected. To compare the low-fidelity/variable-fidelity hypersurfaces with a high fidelity hypersurface, the MBKG method is also trained using Mesh C computations – this surrogate is interrogated at the aforementioned M = 100x100 probe points to create a hypersurface of the high fidelity drag. The variable fidelity surrogates constructed for $N_f = 15$ are used for comparison with the high-fidelity surrogates.

Correction of Mesh A hypersurface using Nf high fidelity computations

Mesh A hypersurface represents a low-fidelity surrogate model, constructed using 56 mesoscale simulations in each of which every cylinder in the domain is discretized using approximately 2

mesh points across the radius. Because of the coarseness of the mesh, the mesoscale computations are computationally inexpensive – each numerical experiment is worth less than an hour's of compute time in a multiprocessor system. However, the low-fidelity hypersurface created using Mesh A computations differ from the high-fidelity (Mesh C) hypersurface in several aspects.

The differences – an overall non-uniform bias and high-frequency oscillations – between drags of different fidelities described in Section 5.4.2. 1 can be seen to translated to the respective surrogate models in the following way in Figure 53a. First, the low-fidelity hypersurface has more high frequency noise compared to the Mesh C hypersurface (Figure 53a). This is true especially for higher values of ϕ, for example for ϕ > 10% (Figure 53a), where the contour lines of the Mesh A hypersurface can be observed to be more oscillatory compared to the Mesh C hypersurface. Second, the shape of the low-fidelity and the high-fidelity hypersurfaces differ from one another. This can be seen by observing that the contour lines of the two hypersurfaces are not parallel to each other (Figure 53a), especially for Ma > 2.5.Third, the Mesh A hypersurface underestimates the drag value in the parameter space. For instance, unlike the high-fidelity hypersurface, where the maximum value of $\tilde{\mathrm{F}}_{\mathrm{D}}(\mathrm{Ma}, \phi)$ is approximately 1.6, the Mesh A hypersurface estimates the maximum value of $\tilde{\mathrm{F}}_{\mathrm{D}}(\mathrm{Ma}, \phi)$ to be approximately 0.9 (Figure 53a). This is in agreement with the trends shown in Figure 52, where $\overline{F_D}(t^*; Ma, \Phi)$ was shown to be under-approximated using $r/\Delta x = 2$ (Mesh A) compared to $r/\Delta x = 8$ (Mesh C) computations for (Ma, ϕ) = (3.5,1.0%) (Section 5.4.2. 1).

The Mesh A hypersurface is corrected using N_f high-fidelity computations using the methods described in Section 5.2.3 . Representative contour plots of the variable-fidelity hypersurface obtained after correcting the low-fidelity (Mesh A) hypersurface using N_f = 15 are shown in Figure 53(b)-(d); the results are compared with the high-fidelity hypersurface.

The space-mapping correction technique is based on an affine transformation of the low-fidelity hypersurface. Because of this, the bias in the low-fidelity hypersurface is corrected efficiently and the multifidelity hypersurface no longer underestimates the value of $\tilde{\mathrm{F}}_{\mathrm{D}}(\mathrm{Ma}, \phi)$ in the parameter space. In contrast to the Mesh A hypersurface, where the peak value of $\tilde{\mathrm{F}}_{\mathrm{D}}(\mathrm{Ma}, \phi)$ was found to be 0.9 (Figure 53a), both the (corrected) variable fidelity hypersurface as well the high fidelity hypersurface estimates the peak value of $\tilde{\mathrm{F}}_{\mathrm{D}}(\mathrm{Ma}, \phi)$ to be approximately 1.6 (Figure 8b). However, the bias in space mapping method is not a function of the parameter space. Therefore, the overall

shape of the variable fidelity hypersurface is similar to that of the low-fidelity hypersurface (Figure 8a and 8b) and differs from the high-fidelity hypersurface. The contour lines of the corrected hypersurface are not parallel to those of the high-fidelity hypersurface (Figure 8b), especially for Ma > 2.5. In addition to this, because the constants of the space mapping correction is constant in the parameter space, the variable fidelity surrogate retains the noise from the low-fidelity hypersurface in the variable fidelity hypersurface (Figure 53b). This can be seen as oscillations in the contours of variable fidelity hypersurface, especially for ϕ > 10%. In conclusion, while the space-mapping technique is effective in correcting the discrepancy in the values of the high and low fidelity surrogate models, both the overall shape as well as high-frequency oscillations of the low-fidelity hypersurface is not corrected effectively to resemble that of the high-fidelity hypersurface by the space-mapping technique.

Both the RBF correction technique as well as the MBKG technique are based on adding an error as a function of the parameter space to the low-fidelity hypersurface. The variable fidelity hypersurfaces obtained by fitting the error with an RBF network (Figure 8c) or with the MBKG method (Figure 8d) do not retain the high-frequency oscillations present in the low-fidelity hypersurface. This is in contrast to the space-mapping method which retains the noise in the low-fidelity hypersurface (Figure 8b). This is because unlike the space-maping method, where the multifidelity hypersurface is a global affine map of the low-fidelity hypersurface, the error function in the RBF or MBKG techqniues are non-linear functions of the parameter space. Hence, the RBF network or the MBKG method can locally correct the bias as well as the high frequency oscillations in the low-fidelity hypersurface to obtain a smooth multifidelity surrogate model. For the same reason, the shape of the low-fidelity hypersurface (Figure 8a) can also be effectively corrected to resemble that of the high-fidelity hypersurface (Figure 8c-d). This can be seen in Figure 8c-d where the contour lines of the variable fidelity hypersurfaces almost coincide with those of the high-fidelity hypersurface. The only regions in the parameter space where the contour lines of the two hypersurfaces differ are near the boundaries (i.e. ϕ > 12% and Ma >3) for the RBF correction method (Figure 8c). This is because the RBFs, which are not compactly supported, are truncated at the boundaries and do not satisfy the consistency condition. Non-compactly supported RBFs are erroneous near the boundaries of the parameter space and the contours of the variable fidelity hypersurface do not follow those of the high fidelity hypersurface. The MBKG method, on the other hand, does not suffer from this limitation are the contours of the variable fidelity hypersurface

closely follows those of the high-fidelity hypersurface almost everywhere in the parameter space (Figure 8d). Finally, similar to the space-mapping techqniue, the variable fidelity hypersurfaces using the RBF method as well as the MBKG do not underestimate the value of $\tilde{F}_D(Ma, \phi)$ in the parameter space (Figure 8c-d).

In summary, a low-fidelity surrogate created using Mesh A using 56 computations is computationally inexpensive and affordable, but it suffers from high-frequency oscillations, and differs in shape and underestimates the values of a high-fidelity hypersurface created using 56 Mesh C computations. However, a high-fidelity hypersurface is computationally expensive and can become prohibitive for high-dimensional parameter space. On the other hand, a multifidelity hypersurface created using Mesh A computations and N_f (which is set to 15 in this section) high-fidelity simulations is both computationally more affordable than a purely high-fidelity hypersurface (Figure so-and-so for compute time) as well as less erroneous compared to a low-fidelity hypersurface. A variable fideilty hypersurface obtained using any of the three techqniues discussd previously does not under-estimate the value of a high-fidelity hypersurface, is noise-free if obtained using RBF or the MBKG method and resembles the shape of the high-fidelity hypersurface very closely almost everywhere in the parameter space if created using the MBKG method. Therefore a variable fidelity hypersurface serves as a good trade-off between affordability (in terms of compute time) and accuracy (in terms of error entalied) for surrogate-based multiscale modeling.

Do the same conclusions hold if the low-fidelity surrogate is better approximated than the Mesh A hypersurface shown in Figure 8a? In the next section, the error entailed and the computation cost of variable-fidelity surrogate models created from slightly better (Mesh B) low-fidelity surrogate is analyzed using the three correction methods descried previously.

Correction of Mesh B hypersurface using N_f high fidelity computations

Similar to the Mesh A hypersurface, a Mesh B hypersurface is also a visual representation of low-fidelity surrogate model. Each of these computations is such that every cylinder in the domain is discretized by approximately 4 mesh points across the radius; these simulations are slightly more resolved than Mesh A computations (Figure 51), but are still less resolved than high-fidelity (Mesh C) computations. The simulations are computationally more expensive than Mesh A computations

– a typical numerical experiment is worth slightly more an hour's compute time in a multiprocessor system; however because the mesoscale experiments are two-dimensional, large number of Mesh B numerical experiments are still inexpensive compared to high-fidelity computations.

Unlike the Mesh A hypersurface, the low-fidelity hypersurface created using Mesh B computations does not contain high-frequency oscillations (Figure 54a). Furthermore, although the shape of the low-fidelity hypersurface differs from that of the high-fidelity hypersurface – the contours lines of the two hypersurfaces are misaligned at higher values of ϕ and *Ma* (Figure 54a) - the misalignment is not as severe as the Mesh A hypersurface (Figure 8a and Figure 54a). Similarly, the low-fidelity hypersurface underestimates the values of $\tilde{F}_D(\mathrm{Ma}, \phi)$ in the parameter space. But compared to the Mesh A hypersurface, the values of $\tilde{F}_D(\mathrm{Ma}, \phi)$ obtained using Mesh B are closer to those of Mesh C computations (Figure 8a and Figure 54a). For instance, the maximum value of $\tilde{F}_D(\mathrm{Ma}, \phi)$ is approximately 0.9 for the Mesh A hypersurface, while it is 1.5 and 1.6 for Mesh B and Mesh C hypersurfaces respectively. In short, for the purpose of variable-fidelity surrogate modeling, the Mesh B surrogate model serves as a better first approximation compared to a Mesh A hypersurface. The low-fidelity surrogate model is corrected by each of the three methods described in Section 5.2.3 using N_f = 15 high fidelity computations.

Similar to the variable-fidelity model obtained from the Mesh A hypersurface, the variable-fidelity surrogate model created using the space-mapping technique retains the shape of the low-fidelity (Mesh B) surrogate model (Figure 54b). However, unlike the Mesh A hypersurface, the shapes of the low-fidelity (Mesh B) and the high-fidelity hypersurfaces differ only slightly to begin with; therefore the variable-fidelity surrogate created using space-mapping technique agree reasonably well with the high-fidelity hypersurface (Figure 54b). The variable-fidelity hypersurfaces created from Mesh B surrogates by using the RBF method or the MBKG methods almost coincide with the high-fidelity hypersurface (Figure 54c-d). In particular when using the RBF method, the variable fidelity hypersurface created from Mesh B agrees better with the high-fidelity hypersurface compared to the variable-fidelity hypersurface created from Mesh A (Figure 8c and Figure 54c). However, when the MBKG method is used for corrections, the variable-fidelity hypersurfaces approximate the high-fidelity surrogate model equally well (Figure 8d and Figure 54d) – irrespective of whether the starting hypersurface is from Mesh A or Mesh B.

In conclusion, Mesh B computations are slightly more expensive than Mesh A simulations, but are still affordable compared to high-fidelity simulations. Because Mesh B computations are more resolved than Mesh A computations, the low-fidelity hypersurface provides a better first approximation to the high-fidelity hypersurface. Because of this, all the three correction techniques result in reasonably well variable fidelity surrogate models, i.e. the variable fidelity surrogate models corrected from Mesh B computations using any of the three methods approximately the high-fidelity hypersurface closely. However, of all the three correction methods, the MBKG method is particularly attractive.

5.4.2. 3 Comparison of the Methods of Correcting Low-Fidelity Surrogates created from Mesh A/Mesh B computations

Figure 55(a) shows the variation of ε_{L^2} with the number, N_f, of high-fidelity computations used for correcting the low-fidelity hypersurface. For a given low-fidelity hypersurface, irrespective of the value of N_f, among all the methods for corrections, the error is highest for the space-mapping technique and lowest for the MBKG technique. This is because as explained in Sections 0 and 0, the space-mapping technique only corrects the bias of the low-fidelity hypersurfaces and cannot correct the shape or smoothness of the low-fidelity hypersurface to match that of the high-fidelity hypersurface. The RBF network, when used for obtaining a variable-fidelity model, corrects both the shape, smoothness as well as the bias of the low-fidelity surrogate. The variable-fidelity model obtained using RBFs are therefore less erroneous than the space-mapping technique. But RBFs are erroneous near the boundaries of the parameter space. This was seen to be particularly true for variable-fidelity models created from Mesh A surrogates in Section 0. The MBKG method on the other hand does not suffer from boundary issues and corrects both the shape as well as the values of the low-fidelity surrogate, while simultaneously eliminating noise. Therefore, the variable-fidelity model obtained using the MBKG method has the lowest values of ε_{L^2} compared to the other two methods.

Figure 55(a) also shows that for a given correction method, the value of ε_{L^2} is lower when a variable-fidelity surrogate model is created starting from Mesh B computations than for Mesh A computations. This is because, as shown in Sections 0 and 0, the low-fidelity compared to that obtained from Mesh A computations. However, for the MBKG method, the value of ε_{L^2} is comparable irrespective of the choice of the low-fidelity surrogate model. Mesh A computations

are cheaper than Mesh B computations and is particularly attractive for applications to surrogates based on higher dimensional parameter spaces and/or 3D mesoscale computations.

In summary, for a given starting low-fidelity surrogate, the error in the MBKG variable-fidelity surrogate is lowest. Furthermore, for the MBKG method, irrespective of the starting low-fidelity surrogate - Mesh A or Mesh B – the residual errors are comparable. Mesh A computations are cheaper than Mesh B computations and is particularly attractive for applications to surrogates based on higher dimensional parameter spaces and/or 3D mesoscale computations. This makes the MBKG method particularly attractive over the other two methods.

While Figure 55(a) shows that the error decreases for higher values of N_f, the use of larger number of N_f leads to larger ensembles of resolved mesoscale simulations which makes the computational cost prohibitively high. A balance between the compute time and error entailed in using smaller values of N_f is more useful for practical purposes; the variation of the cost function Ψ, given by Equation (4) with N_f is instructive for this purpose.

For the Space Mapping method, Ψ is seen to be the lowest for $N_f = 5$ (Figure 55(b)). The value of Ψ is seen to be lower for Mesh A than for Mesh B. This is because although the error is higher for Mesh A, the overall computational cost of creating Mesh A surrogates outweighs the decrease in error when moving from Mesh A to Mesh B. For $N_f > 5$, the computational cost of high fidelity simulations increases faster than the rate at which the error decreases and the value of Ψ increases for $N_f > 5$. Therefore for the Space Mapping method, the minimum value of Ψ is obtained for $N_f = 5$ for Mesh A computations.

The minimum value of Ψ obtained using the RBF methods is the same as the minimum value of Ψ obtained using the Space Mapping technique (Figure 55(b)). Furthermore, like the Space Mapping technique, because of lesser cost of creating an initial surrogate, Mesh A yields a lower value of Ψ compared to Mesh B. However, RBFs require $N_f = 15$ to reach this minima, as opposed to $N_f = 5$ for the Space Mapping method. This is because the error ε_{L^2} decreases appreciably for RBFs when N_f is changed from 5 to 15 compared to the Space Mapping method (Figure 55(a)) , while the increase in compute time is the same for both the methods. Beyond $N_f = 15$, the increase in compute time outweighs the decrease in error for the RBF method and Ψ increases. In conclusion, for RBFS, the minimum value of Ψ is obtained for $N_f = 15$ for Mesh A computations; this minima is lower than that of Space Mapping methods.

Similar to the RBF method, for the MBKG method, the lowest value of Ψ is obtained to be at N_f = 15, while beyond N_f = 15 the value of Ψ increases. Furthermore, because the errors in Mesh A and Mesh B are comparable to each other but Mesh A is computationally less expensive, Ψ is lowest for Mesh A. Because MBKG methods yields the lowest values of ε_{L^2} compared to RBFs and Space Mapping methods, the value of Ψ is minimum for the MBKG method compared to the other two techniques. In conclusion, the minimum value of Ψ is obtained for Mesh A computations, using N_f = 15 for MBKG methods; the value of Ψ is minimum for MBKG method compared to RBFs and the Space Mapping Techniques. Therefore, a Mesh A low fidelity hypersurface, when corrected using 15 high fidelity samples with the MBKG method, yields the most optimum balance between compute time and error in variable fidelity surrogate modeling.

5.5 CONCLUSIONS

This paper analyzes several variable fidelity methods for surrogate modeling in the context of multiscale modeling of shock-particle interactions. Mesoscale simulations of particles interacting with shocks are performed using the code SCIMITAR3D; the drag on the particle cluster is computed as function of the shock strength and volume fraction of the particle. The drag so obtained is used to create surrogate models using the MBKG method. In the present work, three different sets of surrogate models are constructed. Mesh A and Mesh B surrogates are low fidelity metamodels and are created from ensembles of coarse-grid mesoscale computations, the grid being coarser for Mesh A. For the purpose of determining the bias and errors of the low fidelity surrogates, a third set of surrogate - Mesh C surrogate is created from resolved mesoscale computations. In addition to this, a few extra high fidelity computations are performed and are used to correct the Mesh A and Mesh B surrogates. Three different variable-fidelity techniques, SM, RBFs and MBKG are then used to correct the Mesh A and Mesh B surrogates; to ascertain the effectiveness of these methods, in correcting the low fidelity surrogates, the corrected variable fidelity surrogate is compared to the Mesh C surrogate. The computational cost of performing high-fidelity simulations to correct the low-fidelity surrogates and the error entailed in using variable fidelity models is compared for the three aforementioned techniques.

It is found that among the three methods, the SM techniques is found to correct the bias of low fidelity surrogates alone and preserves the shape of the low-fidelity surrogates. The RBF method, being a non-linear correction technique corrects both the bias as well as the error in the low-fidelity

surrogate. However, compared to these two methods, the MBKG method is found to entail the least error in obtaining a variable fidelity surrogate. In fact not only does the MBKG method yield the best approximation, but it also uses the minimum number of high-fidelity computations to achieve this correction, thereby ensuring minimum computational cost.

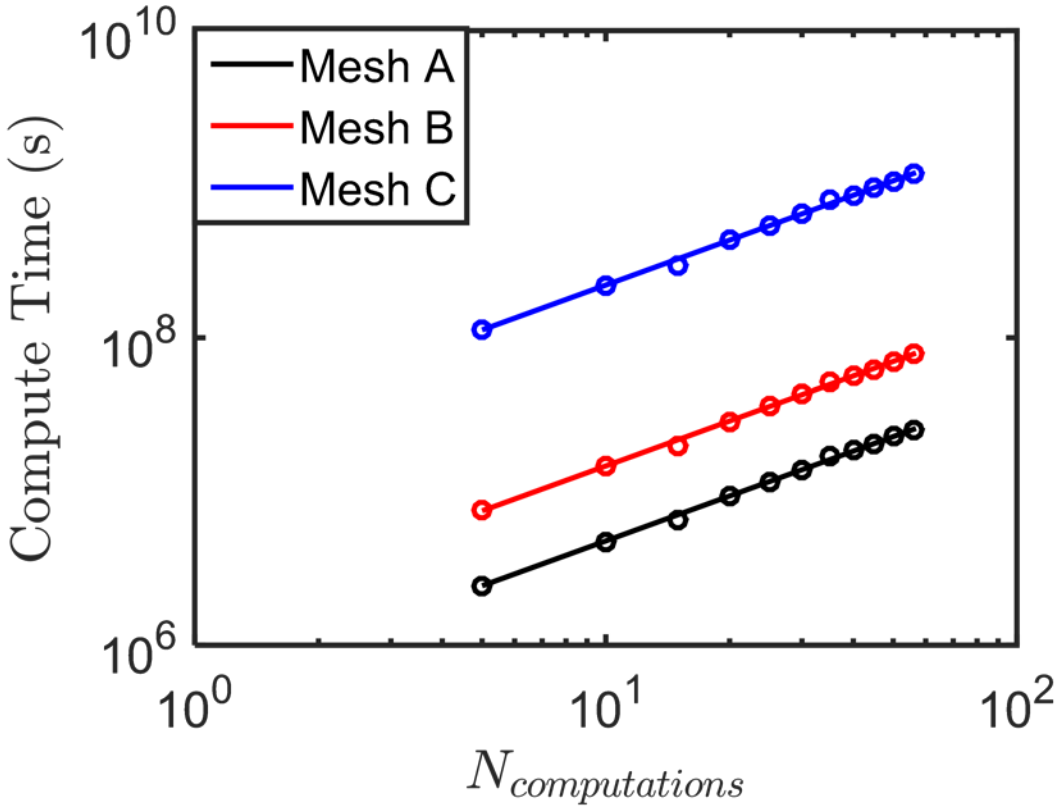


Figure 45 : Indicative compute time for performing mesoscale simulations of shocks interacting with particles. The computational set up and the mesh resolutions are described in [33]; simulations are performed on the high-performance computing system, Helium, at the University of Iowa (https://its.uiowa.edu/hpc).

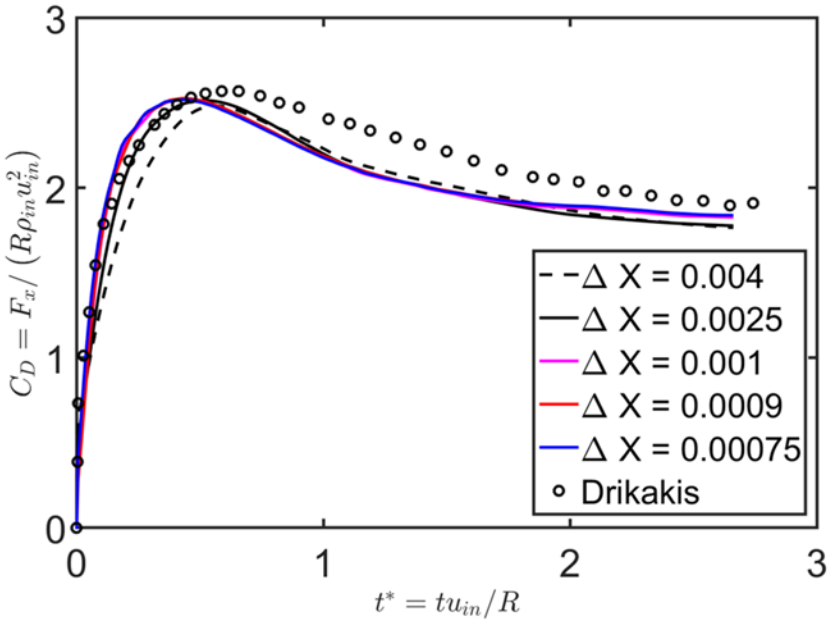


Figure 46 : Comparison of the coefficient of drag force, C_D , on a cylinder as obtained from SCIMITAR3D with the computations of [72]; the shock Mach Number, Ma is 2.6.

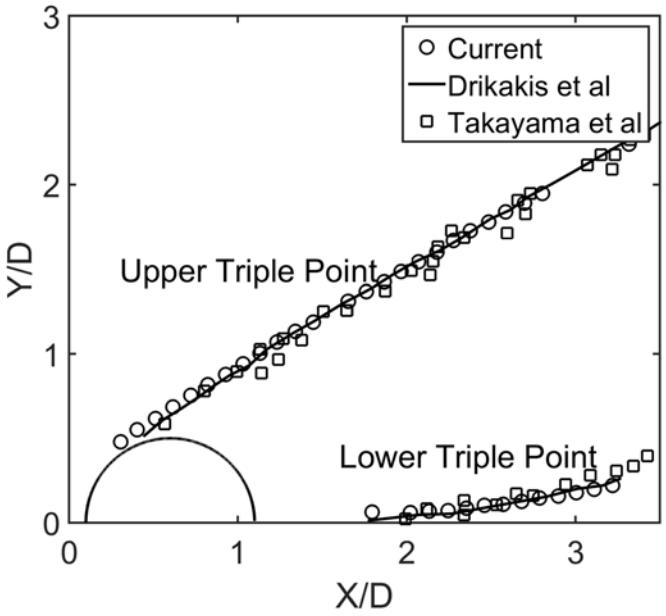


Figure 47 : Comparison of the trajectory of the lower and upper triple points as calculated from SCIMITAR3D with the computations of [72] and the experiments of [73]; the shock Mach number, Ma = 2.81; the mesh size used is $\Delta x=\Delta y= 0.0025$.

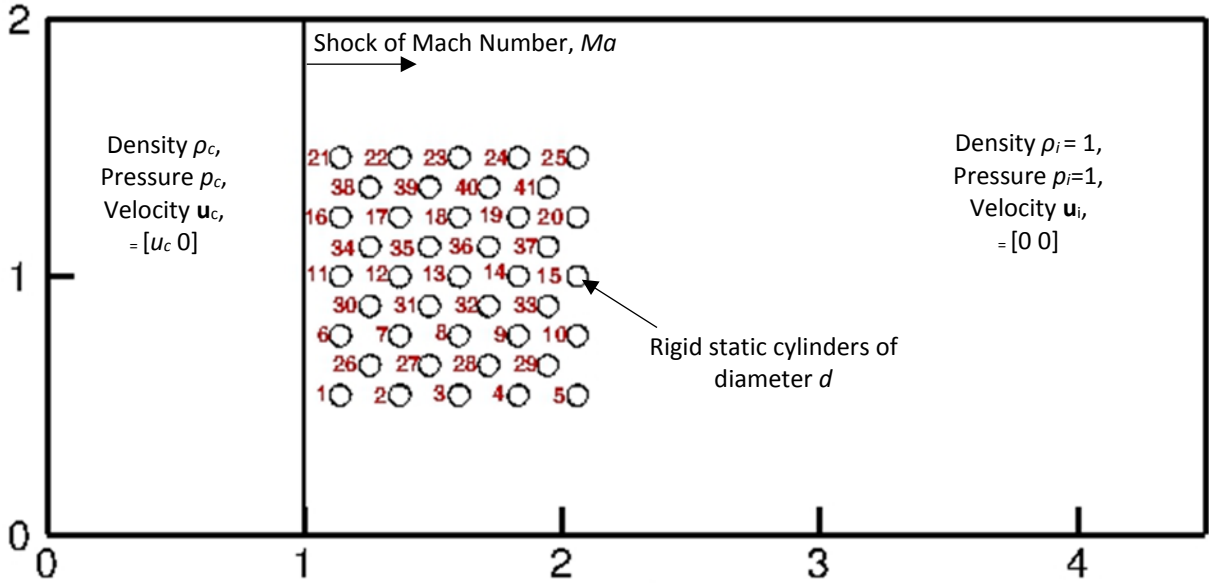


Figure 48 : Description of the set up for numerical experiments; the domain of interest comprises a right moving shock of Mach Number, Ma. To the right of the shock is an array of 41 particles of equal diameter d inscribed in an imaginary unit square (of area A = 1). The volume fraction, ϕ, of the array of particles is given by $\phi = 41\pi d^2/4$. The particles are numbered 1 through 41 as indicated in the figure.

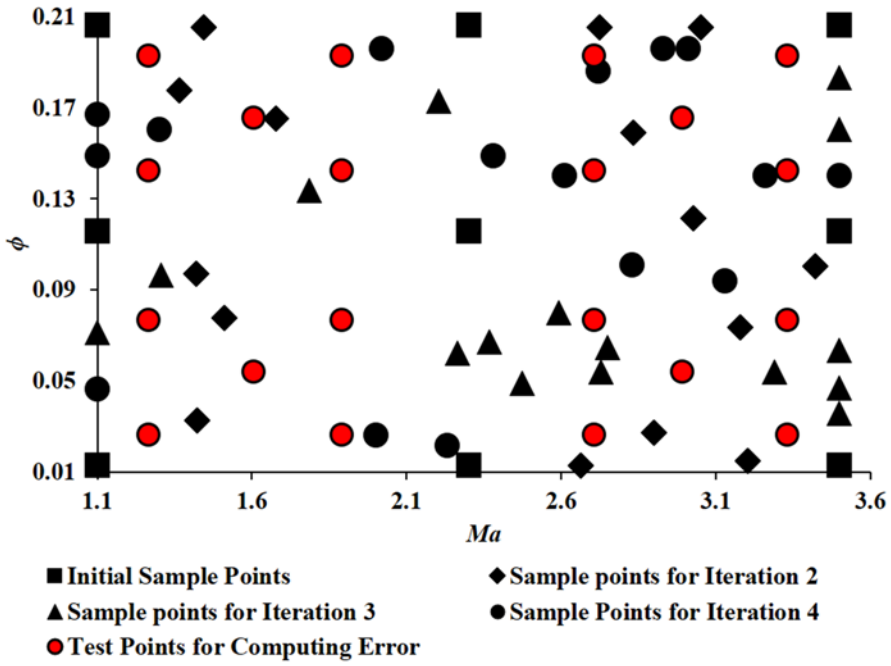


Figure 49 : Location of the points of mesoscale computations in the parameter space; these points serve as input data to create a surrogate model for $\widetilde{\boldsymbol{F}}_{\boldsymbol{D}}$. The drag at points marked as "Initial Sample Points" are first used to create a first approximation of a surrogate model for $\widetilde{\boldsymbol{F}}_{\boldsymbol{D}}$. The points marked as "Sample points for Iteration 2" are as calculated from the adaptive sampling algorithm of DKG; mesoscale computations are performed to obtain the drag at these locations for constructing the second approximation of a surrogate for $\widetilde{\boldsymbol{F}}_{\boldsymbol{D}}$. The process is repeated for other iterations. To cross-validate the quality of approximation a set of mesoscale computations are performed at the points labelled as "Test Points for Computing Error"; the drag from the computations at these points are not used to train the metamodeling techniques but are used to compare against the prediction of the surrogate model after each iteration to compute the error, $\boldsymbol{\varepsilon}_{\boldsymbol{L}_2}$.

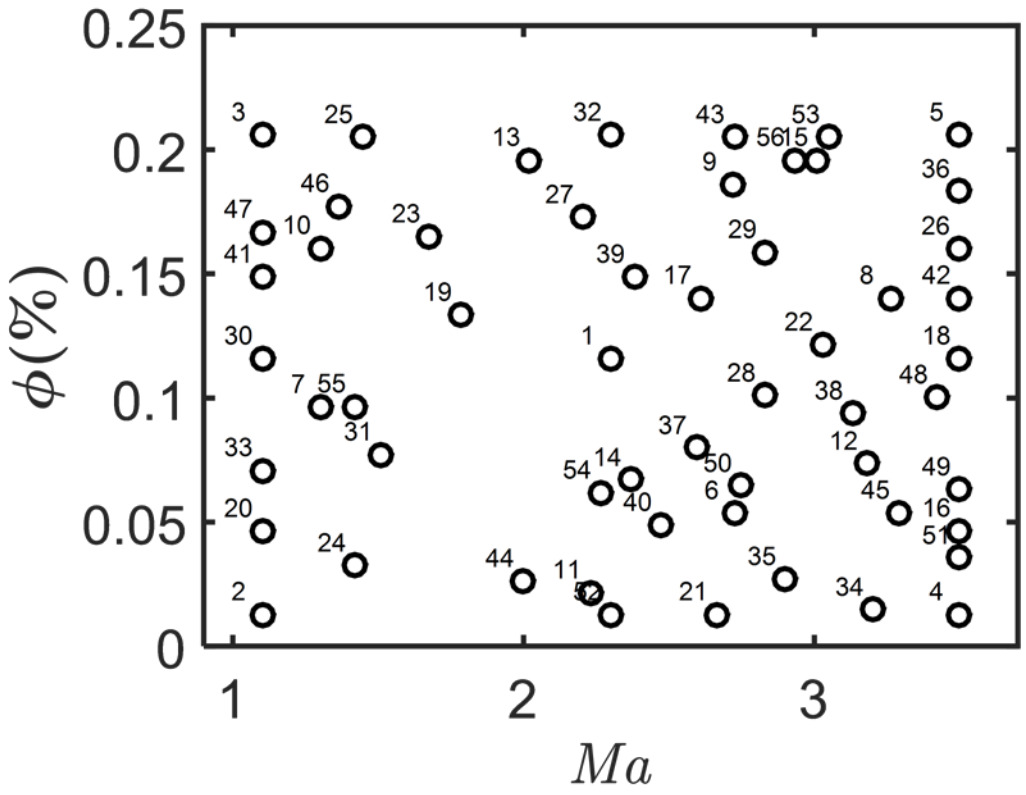


Figure 50 : Sequential selection of locations of N_f locations for high –fidelity simulations for correcting low fidelity surrogates determined using the Kaufman Algorithm. All points are the locations of low-fidelity (Mesh A/Mesh B) simulations. The N_f locations for high-fidelity simulations are selected from the locations of low-fidelity simulations. As an example, when 15 high-fidelity simulations are used for correcting low fidelity surrogates, N_f is set to 15 and the locations marked as 1-15 in this figure are selected for performing Mesh C computations.

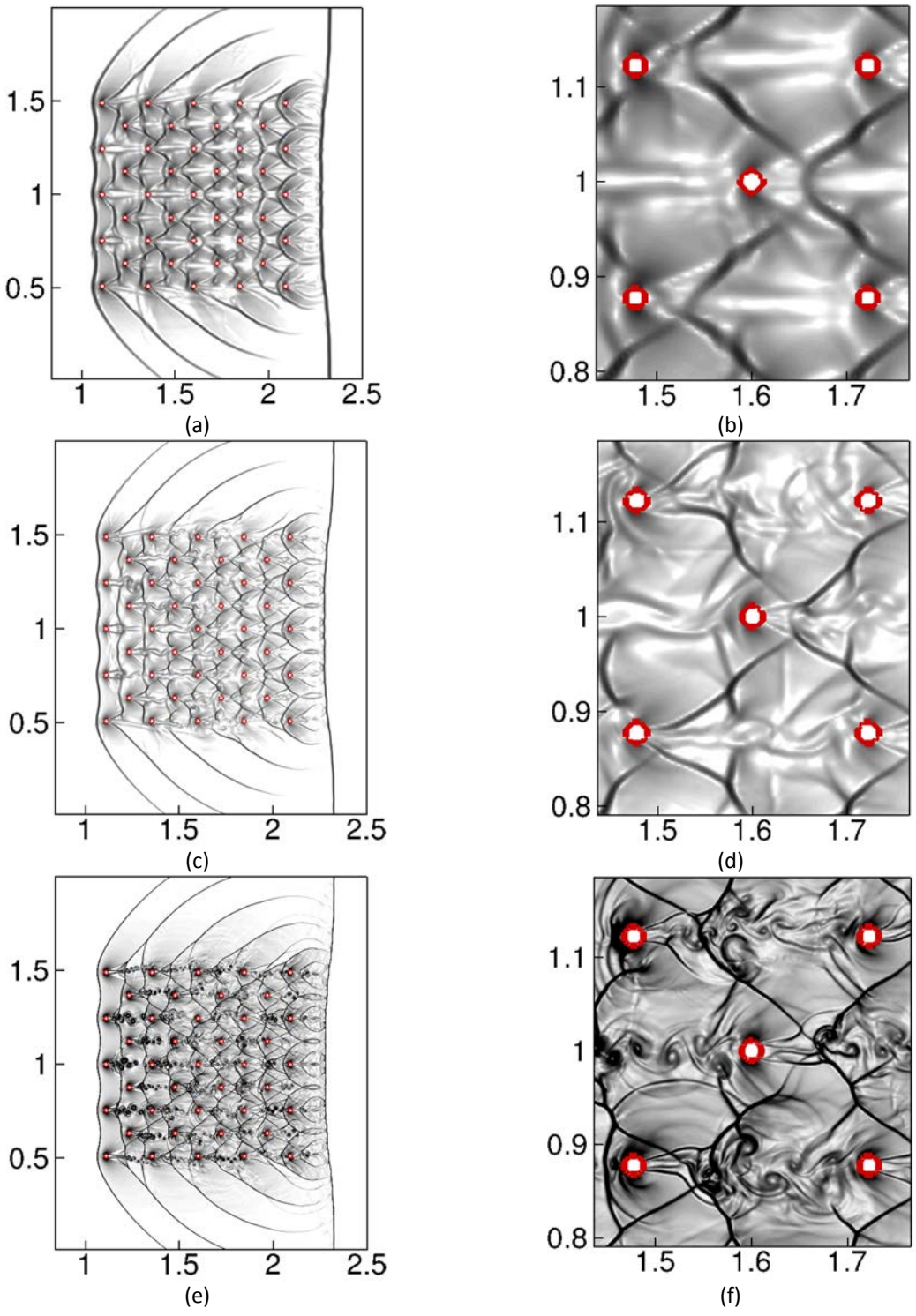


Figure 51 : Numerical Schlieren images of the interaction of a shock of Ma = 3.5 with a particle cluster of ϕ = 1% at $t^* = 1.32$. Figures (a),(c) and (e) show the numerical Schlieren images for the whole particle cluster computed using Mesh A, Mesh B and Mesh C respectively; Figures (b),(d) and (f) show the zoomed-in view of the Schlieren images near the five central particles computed using Mesh A, Mesh B and Mesh C respectively. Mesh A and Mesh B are low fidelity computations, while Mesh C is a high fidelity computation.

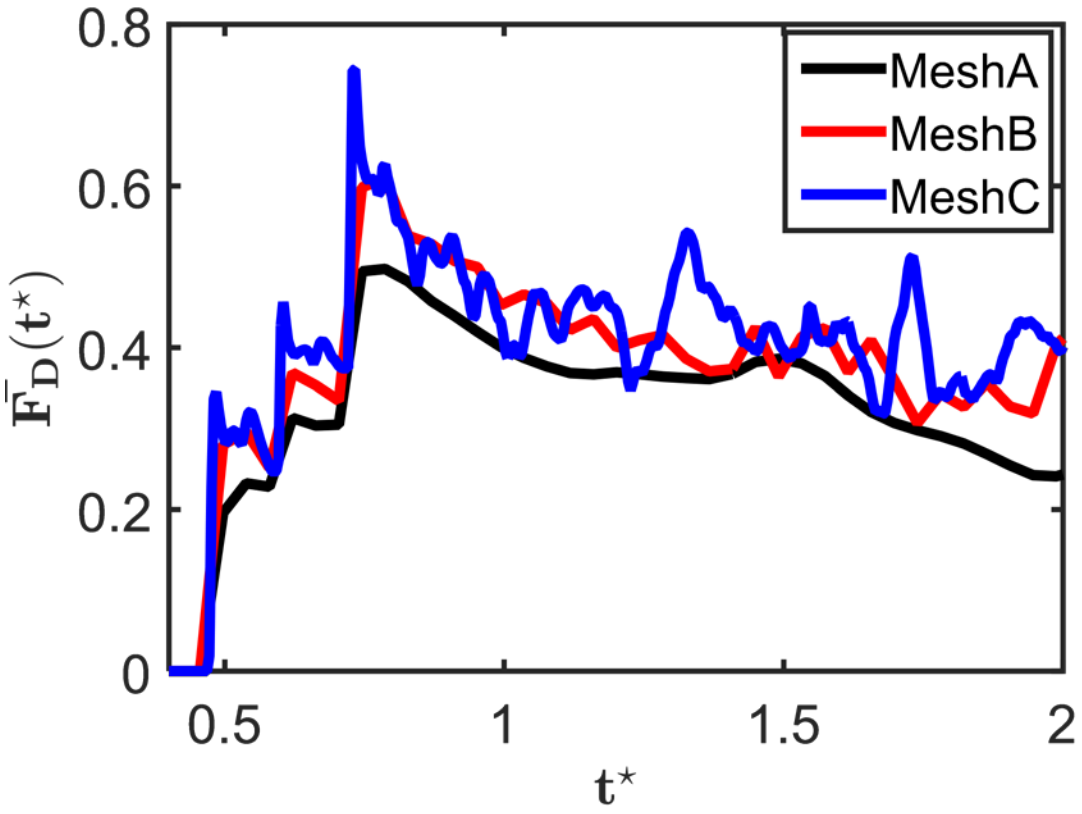


Figure 52 : Comparison of the evolution of volume-averaged drag, $\overline{F_D}(t^*)$ with time, t* for a shock of Ma = 3.5 interacting with a particle-cluster of volume fraction ϕ = 1% computed using low fidelity computations (Mesh A and Mesh B) and high fidelity computations (Mesh C).

Spatio-temporal averaged drag	$\widetilde{F_D}(Ma, \phi)$
Mesh A	0.426
Mesh B	0.512
Mesh C	0.522

Table 6 : Value of $\widetilde{F_D}(Ma, \phi)$ computed using the low-fidelity (Mesh A and Mesh B) and high-fidelity (Mesh C) computations for a shock of Ma = 3.5 interacting with a particle-cluster of volume fraction ϕ = 1% Ma = 3.5 and ϕ = 1%

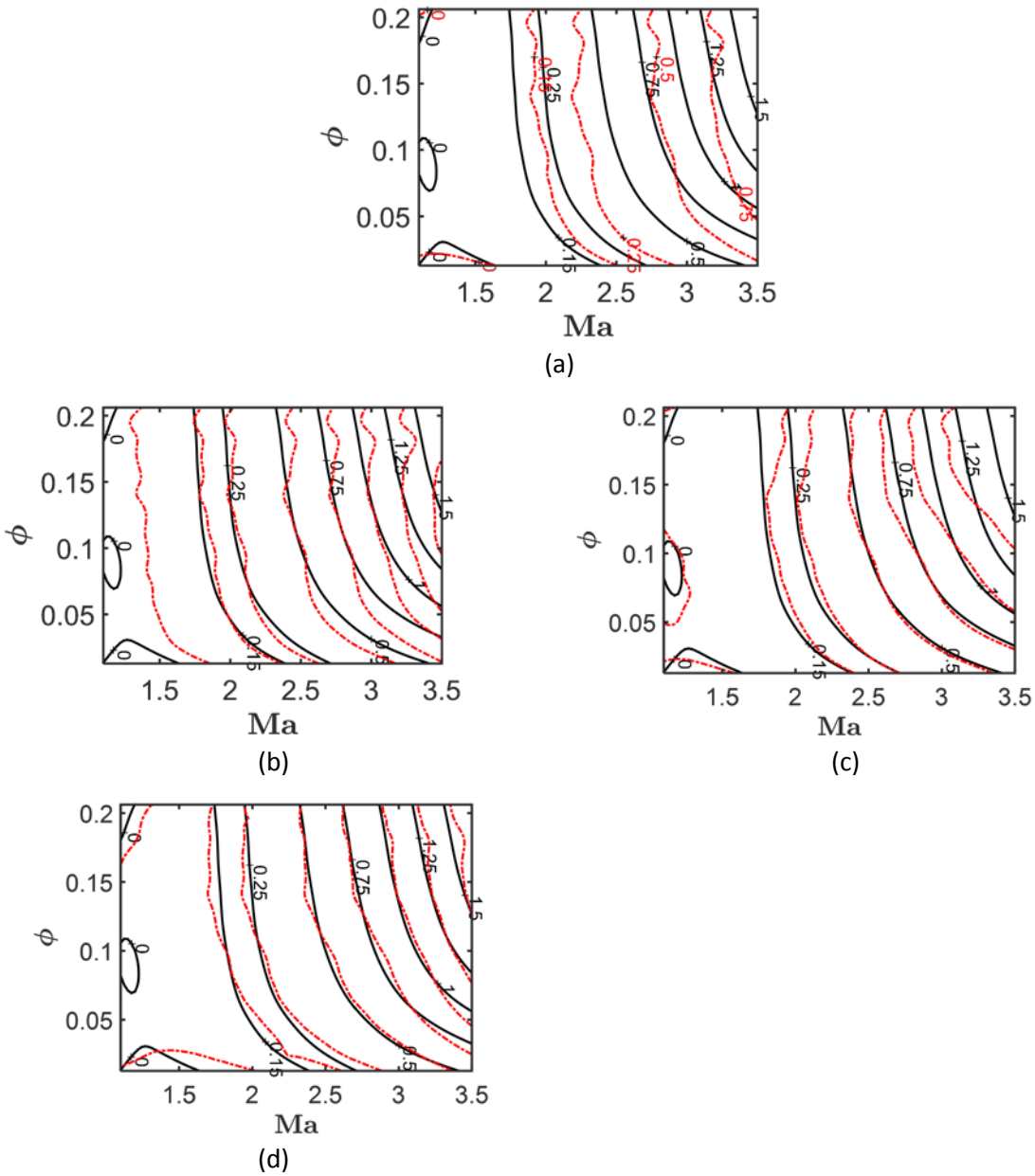


Figure 53 : Comparison of the contour of the high fidelity (Mesh C) hypersurface of $\widetilde{F_D}$ – shown in solid lines - with those of the low fidelity (Mesh A) hypersurface –shown in dashed lines - (a) before correction (b) after correction using space mapping (c) after correction using RBFs (d) after correction using the MBKG method and €. The number of high fidelity computations used for correcting the low fidelity surrogate is 15.

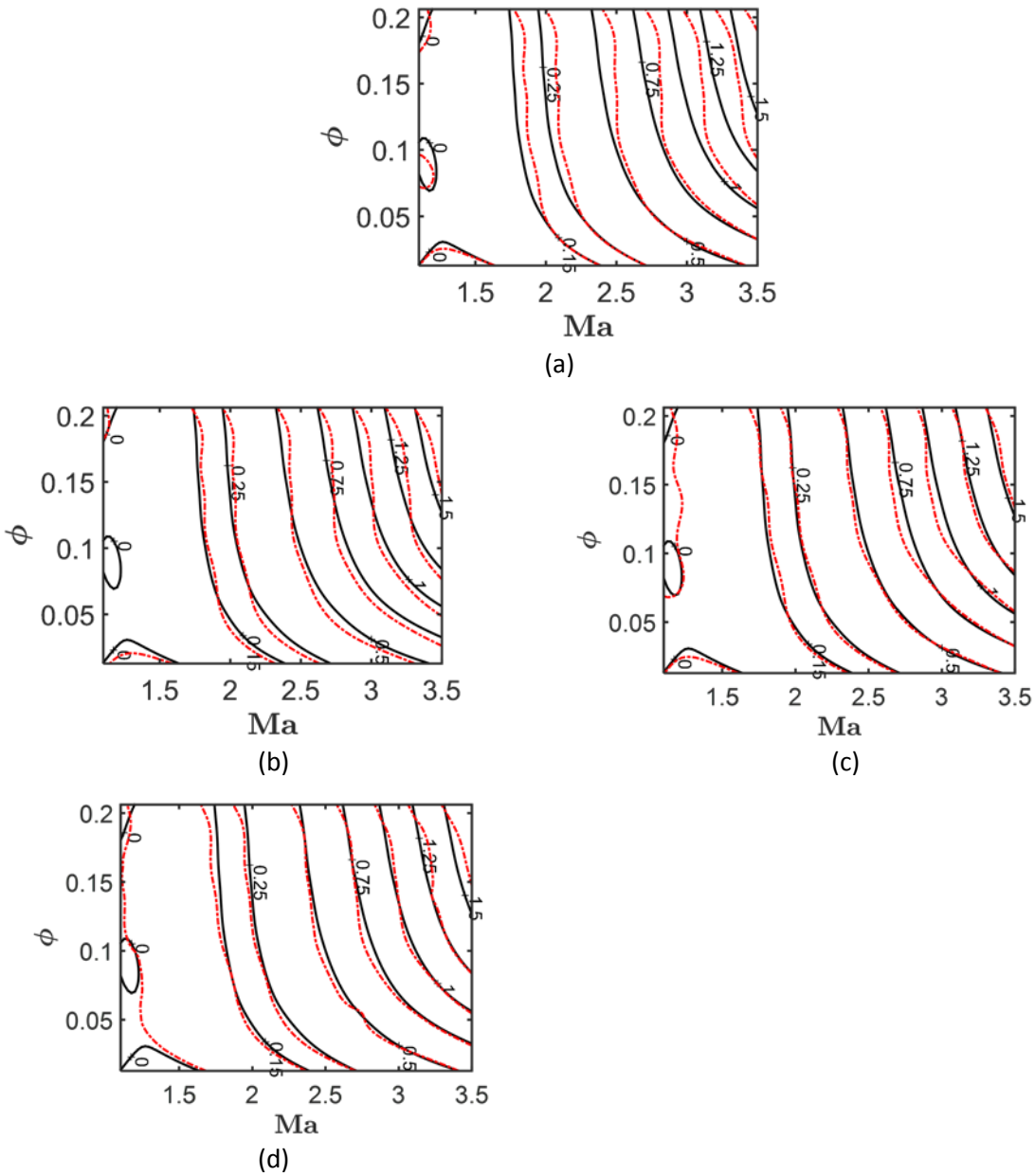
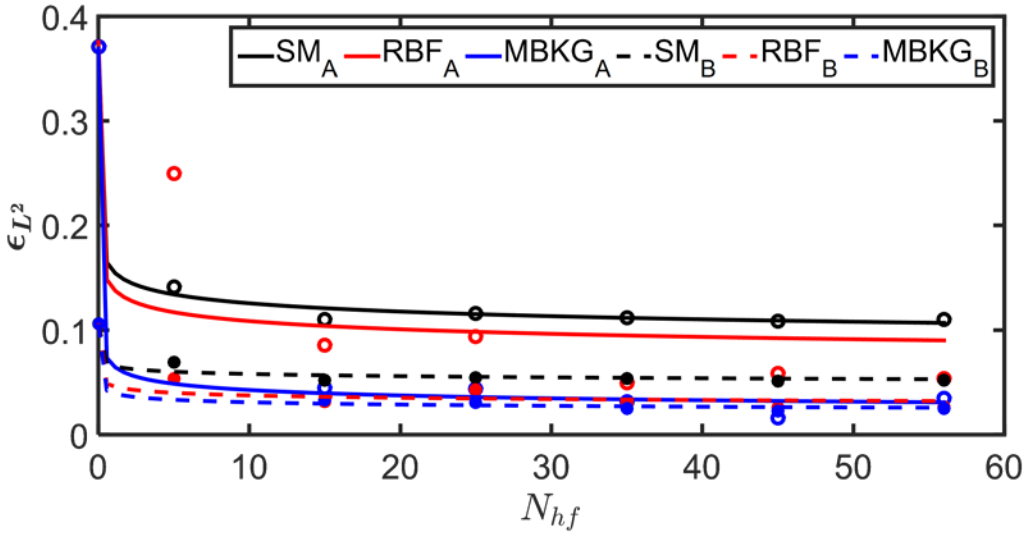
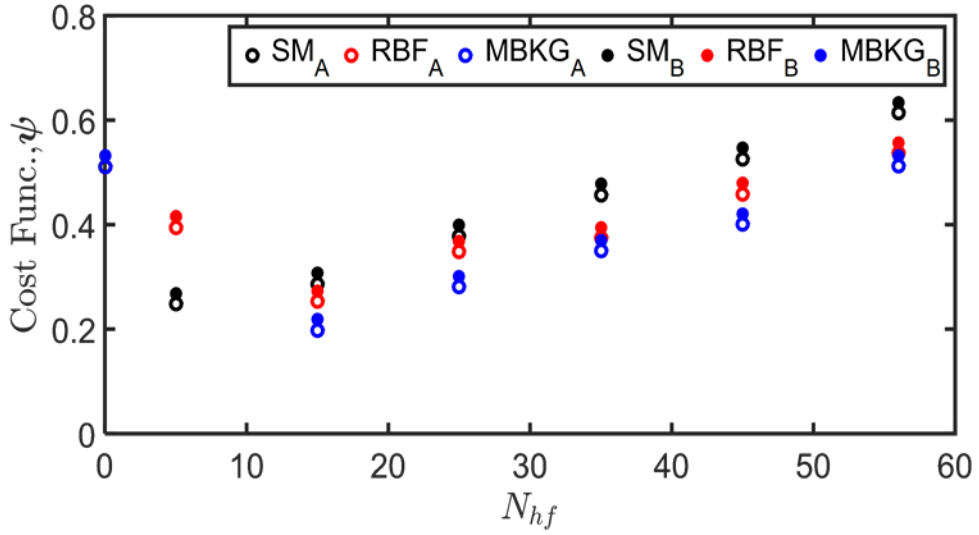


Figure 54: Comparison of the contour of the high fidelity (Mesh C) hypersurface of $\widetilde{F_D}$ – shown in solid lines - with those of the low fidelity (Mesh B) hypersurface –shown in dashed lines - (a) before correction (b) after correction using space mapping (c) after correction using RBFs (d) after correction using the MBKG method and €. The number of high fidelity computations used for correcting the low fidelity surrogate is 15.



(a)



(b)

Figure 55 : (a)Variation of the L2 error and the compute time for correcting low fidelity surrogate models with the computational time used for correcting the surrogate model. (b) Variation of the Cost Function, given by Equation (4) with the number of high fidelity computations, N_{HF}. High fidelity computations are performed using Mesh C computations, while low fidelity surrogates are created using Mesh A or Mesh B computations as indicated in the figure.

CHAPTER 6. CONCLUSIONS

The present work presents a framework for multiscale modeling of multimaterial flows using surrogate modeling techniques in the particular context of shocks interacting with clusters of particles. The work builds a framework for bridging scales in shock-particle interaction by using ensembles of resolved mesoscale computations of shocked particle laden flows. The information from mesoscale models is "lifted" by constructing metamodels of the closure terms - the thesis analyzes several issues pertaining to surrogate-based multiscale modeling frameworks.

First, to create surrogate models, 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. 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.

After this, closure laws for drag are constructed in the form of surrogate models derived from real-time resolved mesoscale computations of shock-particle interactions. The mesoscale computations are performed to calculate the drag force on a cluster of particles for different values of Mach Number and particle volume fraction. Two Kriging-based methods, viz. the Dynamic Kriging Method (DKG) and the Modified Bayesian Kriging Method (MBKG) are evaluated for their ability to construct surrogate models with sparse data; i.e. using the least number of mesoscale simulations. It is shown that unlike the DKG method, the MBKG method converges monotonically even with noisy input data and is therefore more suitable for surrogate model construction from numerical experiments.

In macroscale models for shock-particle interactions, Subgrid Particle Reynolds' Stress Equivalent (SPARSE) terms arise because of velocity fluctuations due to fluid-particle interaction in the subgrid/meso scales. Mesoscale computations are performed to calculate the SPARSE terms and the kinetic energy of the fluctuations for different values of Mach Number and particle volume fraction. Closure laws for SPARSE terms are constructed using the MBKG method. It is found

that the directions normal and parallel to those of shock propagation are the principal directions of the SPARSE tensor. It is also found that the kinetic energy of the fluctuations is independent of the particle volume fraction and is 12-15% of the incoming shock kinetic energy for higher Mach Numbers.

Finally, the thesis addresses the cost of performing large ensembles of resolved mesoscale computations for constructing surrogates. Variable fidelity techniques are used to construct an initial surrogate from ensembles of coarse-grid, relative inexpensive computations, while the use of resolved high-fidelity simulations is limited to the correction of initial surrogate. Different variable-fidelity techniques, viz the Space Mapping Method, RBFs and the MBKG methods are evaluated based on their ability to correct the initial surrogate. It is found that the MBKG method uses the least number of resolved mesoscale computations to correct the low-fidelity metamodel. Instead of using 56 high-fidelity computations for obtaining a surrogate, the MBKG method constructs surrogates from only 15 resolved computations, resulting in drastic reduction of computational cost.

REFERENCES

[1] V.M. Boiko, Shockwave Interaction with a Cloud of Particles, ShockWaves. 7 (1997) 275–285.

[2] J. Wagner, S. Beresh, S. Kearney, W. Trott, J. Castaneda, B. Pruett, M. Baer, Interaction of a Planar Shock with a Dense Field of Particles, in: K. Kontis (Ed.), 28th Int. Symp. Shock Waves, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012: pp. 115–120. http://link.springer.com/chapter/10.1007/978-3-642-25685-1_19 http://link.springer.com/chapter/10.1007/978-3-642-25685-1_19#page-1 http://link.springer.com/content/pdf/10.1007/978-3-642-25685-1_19.pdf http://www.springerlink.com/index/10.1007/978-3-642-25685-1.

[3] R. Hambli, Numerical procedure for multiscale bone adaptation prediction based on neural networks and finite element simulation, Finite Elem. Anal. Des. 47 (2011) 835–842. doi:10.1016/j.finel.2011.02.014.

[4] R. Hambli, Multiscale prediction of crack density and crack length accumulation in trabecular bone based on neural networks and finite element simulation, Int. J. Numer. Methods Biomed. Eng. 27 (2011) 461–475. doi:10.1002/cnm.

[5] R. Hambli, Apparent damage accumulation in cancellous bone using neural networks., J. Mech. Behav. Biomed. Mater. 4 (2011) 868–78. doi:10.1016/j.jmbbm.2011.03.002.

[6] J.F. Unger, S. Eckardt, Multiscale Modeling of Concrete, Arch. Comput. Methods Eng. 18 (2011) 341–393. doi:10.1007/s11831-011-9063-8.

[7] J.F. Unger, C. Konke, Neural networks as material models within a multiscale approach, Comput. Struct. 87 (2009) 1177–1186. doi:10.1016/j.compstruc.2008.12.003.

[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, Phys. Fluids. 11 (1999) 378. doi:10.1063/1.869887.

[9] G. DeOliveira, A.K. Kapila, D.W. Schwendeman, J.B. Bdzil, W.D. Henshaw, C.M. Tarver, Detonation diffraction, dead zones, and the ignition-and-growth model., in: Thirteen. Symp. Int. Detonation, 2006: p. Vol 16. http://www.overtureframework.org/publications/DetSymp06Rev2.pdf.

[10] M.R. Baer, J.W. Nunziato, A 2-Phase Mixture Theory for the Deflagration-to-Detonation Transition (Ddt) in Reactive Antigranulocytes-Materials, Int. J. Multiph. Flow. 12 (1986) 861–889.

[11] M.R. Baer, Modeling heterogeneous energetic materials at the mesoscale, Thermochim. Acta. 384 (2002) 351–367. doi:10.1016/S0040-6031(01)00794-8.

[12] Y. Abdelhamid, U. El Shamy, Multiscale Modeling of Flood-Induced Scour in a Particle Bed, Bridges. 10 (2014) 740–749.

[13] G.E. Karniadakis, M. Israeli, S.A. Orszag, High-order splitting methods for the incompressible Navier-Stokes equations, J. Comput. Phys. 97 (1991) 414–443. doi:10.1016/0021-9991(91)90007-8.

[14] E. Vanden-eijnden, W. Ren, X. Li, B. Engquist, E. Weinan, HMM_review.pdf, n.d. http://www.math.nus.edu.sg/ matrw/multiscale/publications/HMM_review.pdf (accessed October 30, 2014).

[15] Y. Ling, J.L. Wagner, S.J. Beresh, S.P. Kearney, S. Balachandar, Interaction of a planar shock wave with a dense particle curtain: Modeling and experiments, Phys. Fluids. 24 (2012) 113301. doi:10.1063/1.4768815.

[16] 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, J. Comput. Phys. 228 (2009) 1365–1379. doi:10.1016/j.jcp.2008.10.037.

[17] O. Ortiz, Eulerian-Eulerian model of 1-D compressible particle-laden flow: running shock impinging on a cloud of particles, Thesis, 2011. http://scholarworks.calstate.edu/handle/10211.10/1368 (accessed May 26, 2016).

[18] C.T. Crowe, M.P. Sharma, D.E. Stock, The Particle-Source in Cell (PSI-Cell) model for gas-droplet flows, J Fluids Eng. 6 (1977) 325–332.

[19] B. Shotorban, F. Mashayek, A stochastic model for particle motion in large-eddy simulation, J. Turbul. 7 (2006) N18. doi:10.1080/14685240600595685.

[20] 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 AIAAASMESAEASEE Jt. Propuls. Conf., 2011.

[21] S. Davis, O. Sen, G. Jacobs, H.S. Udaykumar, Multiscale Modeling of Particles Embedded in High Speed Flows, in: APS Shock Compression Condens. Matter Meet. Abstr., 2015: p. 4001.

[22] G.B. Jacobs, J.S. Hesthaven, Implicit-explicit time integration of a high-order particle-in-cell method with hyperbolic divergence cleaning, Comput. Phys. Commun. 180 (2009) 1760–1767. doi:10.1016/j.cpc.2009.05.020.

[23] G.B. Jacobs, W.S. Don, T. Dittmann, High-order resolution Eulerian–Lagrangian simulations of particle dispersion in the accelerated flow behind a moving shock, Theor. Comput. Fluid Dyn. 26 (2010) 37–50. doi:10.1007/s00162-010-0214-6.
[24] T. Dittmann, G. Jacobs, W.-S. Don, Dispersion of a cloud of particles by a moving shock: Effects of shape, angle of incidence and aspect ratio, in: American Institute of Aeronautics and Astronautics, n.d. http://arc.aiaa.org/doi/abs/10.2514/6.2011-441.
[25] C. Lu, Multi-scale modeling of shock interactions with a cloud of particles using an artificial neural network for model representation, Procedia IUTAM. (2012).
[26] V.M. Boiko, S.V. Poplavskii, Drag of nonspherical particles in a flow behind a shock wave, Combust. Explos. Shock Waves. 41 (2005) 71–77. doi:10.1007/s10573-005-0008-0.
[27] E. Loth, Compressibility and Rarefaction Effects on Drag of a Spherical Particle, AIAA J. 46 (2008) 2219–2228. doi:10.2514/1.28943.
[28] G.G. Stokes, On the effect of the internal friction of fluids on the motion of pendulums., Trans. Camb. Philos. Soc. 9 (1851).
[29] R. Clift, J.R. Grace, M.E. Weber, Bubbles, drops, and particles, Academic Press, 1978.
[30] G. Tedeschi, H. Gouin, M. Elena, Motion of tracer particles in supersonic flows, Exp. Fluids. 26 (1999) 288–296.
[31] Z.-G. Feng, E.E. Michaelides, Drag Coefficients of Viscous Spheres at Intermediate and High Reynolds Numbers, J. Fluids Eng. 123 (2001) 841–849.
[32] J. Ghaboussi, D.A. Pecknold, M.F. Zhang, R.M. Haj-Ali, Autoprogressive training of neural network constitutive models, Int. J. Numer. Methods Eng. 42 (1998) 105–126.
[33] R. Haj-ali, D.A. Pecknold, J. Ghaboussi, G.Z. Voyiadjis, SIMULATED MICROMECHANICAL MODELS USING ARTIFICIAL NEURAL NETWORKS, J. Eng. Mech. (2001) 730–738.
[34] C. Lu, ARTIFICIAL NEURAL NETWORK FOR BEHAVIOR LEARNING FROM MESO-SCALE SIMULATIONS, APPLICATION TO MULTI-SCALE MULTIMATERIAL FLOWS, The University of Iowa, 2010. http://ir.uiowa.edu/cgi/viewcontent.cgi?article=2035&context=etd&unstamped=1&sei-redir=1&referer=http://scholar.google.com/scholar?hl=en&q=Artificial+neural+network+for+behavior+learning+from+meso-scale+simulations%2C+application+to+multi-scale+multimaterial+flows&btnG=&as_sdt=1%2C16&as_sdtp=#search="Artificial neural network behavior learning from meso-scale simulations, application multi-scale multimaterial flo.
[35] O. Sen, S. Davis, G. Jacobs, H.S. Udaykumar, Meso-scale simulation of shocked particle laden flows and construction of metamodels, in: APS Shock Compression Condens. Matter Meet. Abstr., 2015: p. 6002.
[36] O. Sen, H.S. Udaykumar, G. Jacobs, Effect of Microstructural Geometry for Computing Closure Models in Multiscale Modeling of Shocked Particle Laden Flow, Bull. Am. Phys. Soc. (2016).
[37] L. Zhao, K.K. Choi, I. Lee, R.J. Carver, C. Author, A Metamodeling Method Using Dynamic Kriging and Sequential Sampling, in: Am. Inst. Aeronaut. Astronaut., 2010: pp. 1–18.
[38] 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 Conf. Struct. Multidiscip. Optim., 2013: pp. 1–10.
[39] T.W. Simpson, T.M. Mauery, J.J. Korte, F. Mistree, Kriging Models for Global Approximation in Simulation-Based Multidisciplinary Design Optimization, AIAA J. 39 (2001) 2233–2241.
[40] T.W. Simpson, J.D. Peplinski, P.N. Koch, J.K. Allen, Metamodels for Computer-based Engineering Design : Survey and recommendations, Eng. Comput. 17 (2001) 129–150.
[41] R. Jin, X. Du, W. Chen, The use of metamodeling techniques for optimization under uncertainty, Struct. Multidiscip. Optim. 25 (2003) 99–116. doi:10.1007/s00158-002-0277-0.
[42] G.G. Wang, S. Shan, Review of Metamodeling Techniques in Support of Engineering Design Optimization, J. Mech. Des. 129 (2006) 370–380. doi:10.1115/1.2429697.
[43] S.A. Smolyak, Quadrature and interpolation formulas for tensor products of certain classes of functions, Dokl Akad Nauk SSSR. 4 (1963) 240–243.
[44] X. Ma, N. Zabaras, An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations, J. Comput. Phys. 228 (2009) 3084–3113.
[45] J. Park, I.W. Sandberg, Universal Approximation using Radial-Basis-Function Networks, Neural Comput. 3 (1991) 246–257.
[46] S. Chen, P.M. Grant, C.F.N. Cowan, Orthogonal least-squares algorithm for training multioutput radial basis function networks, IEEE Proc.-F. 139 (1992) 378–384.

[47] X. Tong, E. Luke, M. Remotigue, J. Kang, Multiphase simulations of blast-soil interactions, in: ASME 2013 Fluids Eng. Div. Summer Meet., American Society of Mechanical Engineers, 2013: pp. V01CT25A005–V01CT25A005. http://trafficlight.bitdefender.com/info?url=http%3A//nuclearengineering.asmedigitalcollection.asme.org/pdfaccess.ashx%3FResourceID%3D6236801%26PDFSource%3D13&language=en_US (accessed November 29, 2015).
[48] N. Gaul, Modified Bayesian Kriging for noisy response problems and Bayesian confidence-based reliability-based design optimization, Theses Diss. (2014). http://ir.uiowa.edu/etd/1322.
[49] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, USA, 1996.
[50] L. Leifsson, S. Koziel, A. Bekasiewicz, Fast Low-fidelity Wing Aerodynamics Model for Surrogate-based Shape Optimization, Procedia Comput. Sci. 29 (2014) 811–820. doi:10.1016/j.procs.2014.05.073.
[51] S. Koziel, L. Leifsson, I. Couckuyt, T. Dhaene, Robust variable-fidelity optimization of microwave filters using co-Kriging and trust regions, Microw. Opt. Technol. Lett. 55 (2013) 765–769. doi:10.1002/mop.27447.
[52] L. Leifsson, S. Koziel, Multi-fidelity design optimization of transonic airfoils using physics-based surrogate modeling and shape-preserving response prediction, J. Comput. Sci. 1 (2010) 98–106. doi:10.1016/j.jocs.2010.03.007.
[53] L. Leifsson, S. Koziel, Aerodynamic Shape Optimization by Variable-Fidelity Computational Fluid Dynamics Models: A Review of Recent Progress, J. Comput. Sci. (2015). doi:10.1016/j.jocs.2015.01.003.
[54] L. Huang, Z. Gao, D. Zhang, Research on multi-fidelity aerodynamic optimization methods, Chin. J. Aeronaut. 26 (2013) 279–286. doi:10.1016/j.cja.2013.02.004.
[55] S.E. Gano, J.E. Renaud, J.D. Martin, T.W. Simpson, Update Strategies for Kriging Models for Use in Variable Fidelity Optimization, (2005) 18–21.
[56] H. Dong, B. Song, P. Wang, S. Huang, Multi-fidelity information fusion based on prediction of kriging, Struct. Multidiscip. Optim. (2014). doi:10.1007/s00158-014-1213-9.
[57] Z.-H. Han, S. Görtz, R. Zimmermann, Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function, Aerosp. Sci. Technol. 25 (2013) 177–189. doi:10.1016/j.ast.2012.01.006.
[58] E. Weinan, B. Engquist, Multiscale Modeling and Computation, Not. AMS. 50 (2003) 1062–1070.
[59] B. Engquist, E. Weinan, Multiscale Modeling and Computation, Volume 50, Number 9 - fea-engquist.pdf, n.d. http://www.ams.org/journals/notices/200309/fea-engquist.pdf (accessed November 3, 2014).
[60] E.L. Lee, C.M. Tarver, Phenomenological model of shock initiation in heterogeneous explosives, Phys. Fluids 1958-1988. 23 (1980) 2362–2372.
[61] B. Fornberg, C. Piret, A Stable Algorithm for Flat Radial Basis Functions on a Sphere, Si. 30 (2007) 60–80.
[62] G.E. Fasshauer, Solving differential equations with radial basis functions: multilevel methods and smoothing, Adv. Comput. Math. 11 (1999) 139–159. doi:10.1023/A:1018919824891.
[63] J.A. Leonard, M.A. Kramer, L.H. Ungar, Using radial basis functions to approximate a function and its error bounds, IEEE Trans. Neural Netw. 3 (1992) 624–627. doi:10.1109/72.143377.
[64] D.G. Krige, A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand, J. Chem. Metall. Min. Soc. South Afr. (1952) 201–215.
[65] F. Trochu, A Contouring Program Based on Dual Kriging Interpolation, Eng. Comput. 9 (1993) 160–177.
[66] S. ren N. Lophaven, J. Sø ndergaard, H.B. Nielsen, Kriging Toolbox, 2002.
[67] J.M. Hyman, Patch dynamics for multiscale problems, Comput. Sci. Eng. 7 (2005) 47–53.
[68] G. Samaey, I.G. Kevrekidis, D. Roose, Patch dynamics with buffers for homogenization problems, J. Comput. Phys. 213 (2006) 264–287. doi:10.1016/j.jcp.2005.08.010.
[69] Z. Chen, On the heterogeneous multiscale method with various macroscopic solvers, Nonlinear Anal. Theory Methods Appl. 71 (2009) 3267–3282. doi:10.1016/j.na.2009.01.229.
[70] E. Weinan, W. Ren, E. Vanden-Eijnden, W. E, A General Strategy for Designing Seamless Multiscale Methods, J. Comput. Phys. 228 (2009) 5437–5453. doi:10.1016/j.jcp.2009.04.030.
[71] W.Q. Ren, E. Weinan, Heterogeneous multiscale method for the modeling of complex fluids and micro-fluidics, J. Comput. Phys. 204 (2005) 1–26. doi:10.1016/j.jcp.2004.10.001.
[72] R.E. Bank, T.F. Dupont, H. Yserentant, The hierarchical basis multigrid method, Numer. Math. 52 (1988) 427–458. doi:10.1007/BF01462238.
[73] U. Ghia, K.N. Ghia, C.T. Shin, High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method, J. Comput. Phys. 48 (1982) 387–411. doi:10.1016/0021-9991(82)90058-4.
[74] M. Dorobantu, B. Engquist, Wavelet-based numerical homogenization, Siam J. Numer. Anal. 35 (1998) 540–559.

[75] H.S. Fu, † B.H., A wavelet multiscale method for the inverse problems of a two-dimensional wave equation, Inverse Probl. Sci. Eng. 12 (2004) 643–656. doi:10.1080/10682760410001694203.
[76] F. Ebrahimi, M. Sahimi, Grid coarsening, simulation of transport processes in, and scale-up of heterogeneous media: Application of multiresolution wavelet transformations, Mech. Mater. 38 (2006) 772–785. doi:10.1016/j.mechmat.2005.06.013.
[77] O. Sen, S. Davis, G. Jacobs, H.S. Udaykumar, Evaluation of convergence behavior of metamodeling techniques for bridging scales in multi-scale multimaterial simulation, J. Comput. Phys. 294 (2015) 585–604. doi:10.1016/j.jcp.2015.03.043.
[78] B. Fornberg, T.A. Driscoll, G. Wright, R. Charles, Observations on the behavior of radial basis function approximations near boundaries, Comput. Math. Appl. 43 (2002) 473–490. doi:10.1016/S0898-1221(01)00299-1.
[79] N.J. Gaul, H. Cowles, K.K. Choi, D. Lamb, Modified Bayesian Kriging for Noisy Response Problems for Reliability Analysis, in: Boston, Massachusetts, USA, n.d.
[80] A. Kapahi, S. Sambasivan, H.S. Udaykumar, A three-dimensional sharp interface Cartesian grid method for solving high speed multi-material impact, penetration and fragmentation problems, J. Comput. Phys. 241 (2013) 308–332. doi:10.1016/j.jcp.2013.01.007.
[81] A. Kapahi, H.S. Udaykumar, Dynamics of void collapse in shocked energetic materials: physics of void–void interactions, Shock Waves. 23 (2013) 537–558.
[82] A. Kapahi, J. Mousel, S. Sambasivan, H.S. Udaykumar, Parallel, sharp interface Eulerian approach to high-speed multi-material flows, Comput. Fluids. 83 (2013) 144–156. doi:10.1016/j.compfluid.2012.06.024.
[83] N.K. Rai, A. Kapahi, H.S. Udaykumar, Treatment of contact separation in Eulerian high-speed multimaterial dynamic simulations, Int. J. Numer. Methods Eng. 100 (2014) 793–813. doi:10.1002/nme.4760.
[84] S. Sambasivan, An Evaluation of Ghost-Fluid Methods for Strong Shock Interations with Immersed Solid Interfaces, University of Iowa, Iowa City, 2009.
[85] S.K. Sambasivan, H.S.S. UdayKumar, Sharp interface simulations with Local Mesh Refinement for multi-material dynamics in strongly shocked flows, Comput. Fluids. 39 (2010) 1456–1479. doi:10.1016/j.compfluid.2010.04.014.
[86] S.K. Sambasivan, H.S. UdayKumar, Ghost Fluid method for strong shock interactions Part 2: Immersed solid boundaries, Aiaa J. 47 (2009) 2923–2937.
[87] R.P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), J. Comput. Phys. 152 (1999) 457–492.
[88] J.A. Sethian, Level-Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science., Cambrige University Press, Cambrige, 1999.
[89] R.P. Fedkiw, Coupling an Eulerian fluid calculation to a Lagrangian solid calculation with the ghost fluid method, J. Comput. Phys. 175 (2002) 200–224. doi:10.1006/jcph.2001.6935.
[90] A.J.M. Ferreira, E.J. Kansa, G.E. Fasshauer, V.M.A. Leitão, eds., Progress on Meshless Methods, Springer Netherlands, Dordrecht, 2009. http://www.springerlink.com/index/10.1007/978-1-4020-8821-6.
[91] S.L. Dingman, D.M. Seely-Reynolds, R.C. Reynolds, Application of kriging to estimating mean annual precipitation in a region of orographic influence 1, (1988) 329–339.
[92] A.J.M. Ferreira, E.J. Kansa, G.E. Fasshauer, V.M.A. Leitão, eds., Progress on Meshless Methods, Springer Netherlands, Dordrecht, 2009. http://www.springerlink.com/index/10.1007/978-1-4020-8821-6.
[93] M. Stein, Large Sample Properties of Simulations using Latin Hypercube Sampling, Technometrics. 29 (1987) 143–151.
[94] 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, SAND Rep. SAND2002-0099 Sandia Natl. Lab. Albuq. 2002. (n.d.). http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.470.2044&rep=rep1&type=pdf (accessed April 26, 2016).
[95] T. Bayes, R. Price, An essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, frs communicated by mr. price, in a letter to john canton, amfrs, Philos. Trans. 1763 (n.d.).
[96] A. Gelman, J.B. Carlin, H.S. Stern, D.B. Rubin, Bayesian data analysis, USA: Chapman & Hall/CRC, 2014., Boca Raton, FL, n.d.
[97] M.S. Hamada, A. Wilson, S.C. Reese, H. Martz, Bayesian reliability, Springer Science & Business Media, 2008.

[98] M.K. Cowles, Applied Bayesian statistics: with R and OpenBUGS examples., Springer Science & Business Media, 2013.
[99] D. Drikakis, D. Ofengeim, E. Timofeev, P. Voionovich, COMPUTATION OF NON-STATIONARY SHOCK-WAVE/CYLINDER INTERACTION USING ADAPTIVE-GRID METHODS, J. Fluids Struct. 11 (1997) 665–692. doi:10.1006/jfls.1997.0101.
[100] A.E. Bryson, R.W. Gross, Diffraction of Strong Shocks by Cones, Cylinders and Spheres, J. Fluid Mech. 10 (1961) 1–16.
[101] K. Takayama, K. Itoh, Unsteady drag over cylinders and aerofoils in transonic shock tube flows, in: n.d.
[102] A. Chaudhuri, A. Hadjadj, A. Chinnayya, On the use of immersed boundary methods for shock/obstacle interactions, J. Comput. Phys. 230 (2011) 1731–1748. doi:10.1016/j.jcp.2010.11.016.
[103] W.M. Harris, W.K.S. Chiu, Determining the representative volume element size for three-dimensional microstructural material characterization. Part 1: Predictive models, J. Power Sources. 282 (2015) 552–561. doi:10.1016/j.jpowsour.2015.02.035.
[104] P. Romero, E. Masad, Relationship between the Representative Volume Element and Mechanical Properties of Asphalt Concrete, J. Mater. Civ. Eng. 13 (2001) 77–84.
[105] J. Joos, M. Ender, T. Carraro, A. Weber, E. Ivers-Tiffée, Representative volume element size for accurate solid oxide fuel cell cathode reconstructions from focused ion beam tomography data, Electrochimica Acta. 82 (2012) 268–276. doi:10.1016/j.electacta.2012.04.133.
[106] C. Heinrich, M. Aldridge, A.S. Wineman, J. Kieffer, A.M. Waas, K. Shahwan, The influence of the representative volume element (RVE) size on the homogenized response of cured fiber composites, Model. Simul. Mater. Sci. Eng. 20 (2012). http://iopscience.iop.org/article/10.1088/0965-0393/20/7/075007/pdf (accessed July 6, 2016).
[107] B. Abdallah, F. Willot, D. Jeulin, Stokes Flow Through a Boolean Model of Spheres: Representative Volume Element, Transp. Porous Media. 109 (2015) 711–726. doi:10.1007/s11242-015-0545-2.
[108] Y.J. Cho, W.J. Lee, Y.H. Park, Effect of boundary conditions on the numerical solutions of representative volume element problems for random heterogeneous composite microstructures, Met. Mater. Int. 20 (2014) 1085–1093. doi:10.1007/s12540-014-6012-8.
[109] M. Mehrabadi, S. Tenneti, R. Garg, S. Subramaniam, Pseudo-turbulent gas-phase velocity fluctuations in homogeneous gas–solid flow: fixed particle assemblies and freely evolving suspensions, J. Fluid Mech. 770 (n.d.) 210–246.
[110] M. Sommerfeld, H. Grönig, Decay of shock waves in a dusty-gas shock tube with different configurations, Shock Tubes Waves. (1984) 470–477.
[111] O. Igra, J. Falcovitz, L. Houas, G. Jourdan, Review of methods to attenuate shock/blast waves, Prog. Aerosp. Sci. 58 (2013) 1–35. doi:10.1016/j.paerosci.2012.08.003.
[112] F. Aizik, G. Ben-Dor, T. Elperin, O. Igra, M. Mond, H. Groenig, Attenuation law of planar shock waves propagating through dust-gas suspensions, AIAA J. 33 (1995) 953–955. doi:10.2514/3.12382.
[113] P. Kosinski, On Shock Wave Propagation in a Branched Channel with Particles, Shock Waves. 15 (2006) 13–20.
[114] P. Kosinski, Numerical analysis of shock wave interaction with a cloud of particles in a channel with bends, Int. J. Heat Fluid Flow. 28 (2007) 1136–1143. doi:10.1016/j.ijheatfluidflow.2006.11.003.
[115] P. Kosinski, A. Christian Hoffmann, R. Klemens, Dust lifting behind shock waves: comparison of two modelling techniques, Chem. Eng. Sci. 60 (2005) 5219–5230. doi:10.1016/j.ces.2005.04.035.
[116] Y. Ling, J.L. Wagner, S.J. Beresh, S.P. Kearney, S. Balachandar, Interaction of a planar shock wave with a dense particle curtain: Modeling and experiments, Phys. Fluids. 24 (2012) 113301. doi:10.1063/1.4768815.
[117] M. Parmar, A. Haselbacher, S. Balachandar, Modeling of the unsteady force for shock–particle interaction, Shock Waves. 19 (2009) 317–329. doi:10.1007/s00193-009-0206-x.
[118] M.A. van der Hoef, M. van Sint Annaland, N.G. Deen, J. a. M. Kuipers, Numerical Simulation of Dense Gas-Solid Fluidized Beds: A Multiscale Modeling Strategy, Annu. Rev. Fluid Mech. 40 (2008) 47–70. doi:10.1146/annurev.fluid.40.111406.102130.
[119] B.H. Xu, A.B. Yu, Numerical simulation of the gas-solid flow in a fluidized bed by combining discrete particle method with computational fluid dynamics, Chem. Eng. Sci. 52 (1997) 2785–2809.
[120] W. Shuai, Z. Tianyu, L. Guodong, L. Huilin, S. Liyan, Multi-scale heat transfer in fluidized bed reactors by Eulerian CFD modeling, Fuel. 139 (2015) 646–651. doi:10.1016/j.fuel.2014.09.027.
[121] G. Moula, W. Nastoll, O. Simonin, R. Andreux, Multiscale Study of Reactive Dense Fluidized Bed for FCC Regenerator, Oil Gas Sci. Technol. – Rev. D'IFP Energ. Nouv. 68 (2013) 1073–1092. doi:10.2516/ogst/2013183.

[122] H. Baek, G.E. Karniadakis, A convergence study of a new partitioned fluid–structure interaction algorithm based on fictitious mass and damping, J. Comput. Phys. 231 (2012) 629–652. doi:10.1016/j.jcp.2011.09.025.
[123] G.E. Karniadakis, M. Israeli, S.A. Orszag, High-order splitting methods for the incompressible Navier-Stokes equations, J. Comput. Phys. 97 (1991) 414–443. doi:10.1016/0021-9991(91)90007-8.
[124] D.A. Fedosov, H. Noguchi, G. Gompper, Multiscale modeling of blood flow: from single cells to blood rheology, Biomech. Model. Mechanobiol. 13 (2013) 239–258. doi:10.1007/s10237-013-0497-9.
[125] A.G. Hoekstra, S. Alowayyed, E. Lorenz, N. Melnikova, L. Mountrakis, B. van Rooij, A. Svitenkov, G. Závodszky, P. Zun, Towards the virtual artery: a multiscale model for vascular physiology at the physics–chemistry–biology interface, Phil Trans R Soc A. 374 (2016) 20160146. doi:10.1098/rsta.2016.0146.
[126] W. Wu, W. Rodi, T. Wenka, 3D numerical modeling of flow and sediment transport in open channels., J. Hydraul. Eng. 126 (2000) 4–15.
[127] E.A. Zedler, R. Street, Large-eddy simulation of sediment transport: currents over ripples., J. Hydraul. Eng. 127 (n.d.) 444–452.
[128] C.S. Hill, Interactions Between Channel Topography and Hydrokinetic Turbines: Sediment Transport, Turbine Performance, and Wake Characteristics, PhD Thesis. (2015). http://adsabs.harvard.edu/abs/2015PhDT.....176H (accessed November 5, 2016).
[129] N.G. Deen, S.H.L. Kriebitzsch, M.A. van der Hoef, J.A.M. Kuipers, Direct numerical simulation of flow and heat transfer in dense fluid–particle systems, Chem. Eng. Sci. 81 (2012) 329–344. doi:10.1016/j.ces.2012.06.055.
[130] O.V. Vysokomornaya, G.V. Kuznetsov, P.A. Strizhak, Heat and mass transfer in the process of movement of water drops in a high-temperature gas medium, J. Eng. Phys. Thermophys. 86 (2013) 62–68. doi:10.1007/s10891-013-0805-3.
[131] A.E. Bergles, R.M. Manglik, CURRENT PROGRESS AND NEW DEVELOPMENTS IN ENHANCED HEAT AND MASS TRANSFER, J. Enhanc. Heat Transf. 20 (2013). doi:10.1615/JEnhHeatTransf.2013006989.
[132] J. Chang, G. Wang, J. Gao, K. Zhang, H. Chen, Y. Yang, CFD modeling of particle–particle heat transfer in dense gas-solid fluidized beds of binary mixture, Powder Technol. 217 (2012) 50–60. doi:10.1016/j.powtec.2011.10.008.
[133] S.B. Pope, Small scales, many species and the manifold challenges of turbulent combustion, Proc. Combust. Inst. 34 (2013) 1–31. doi:10.1016/j.proci.2012.09.009.
[134] J. Sierra-Pallares, D.L. Marchisio, M.T. Parra-Santos, J. García-Serna, F. Castro, M.J. Cocero, A computational fluid dynamics study of supercritical antisolvent precipitation: Mixing effects on particle size, AIChE J. 58 (2012) 385–398. doi:10.1002/aic.12594.
[135] Y. Wang, L. Wu, B. Li, W. Zhang, Y. Hu, Numerical Simulation and Analysis of the Mixing Process of Rotary Pressure Exchangers with Different Sizes and Structures, J. Chem. Eng. Jpn. 49 (2016) 573–578. doi:10.1252/jcej.15we140.
[136] S. Davis, G.B. Jacobs, S. Oishik, H.S. UdayKumar, SPARSE: A Subgrid Particle Averaged Reynolds Stress Equivalent Model : Testing with A Priori Closure, Proc R Soc A. (n.d.).
[137] 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, J. Comput. Phys. 228 (2009) 1365–1379. doi:10.1016/j.jcp.2008.10.037.
[138] G.B. Jacobs, J.S. Hesthaven, High-order nodal discontinuous Galerkin particle-in-cell method on unstructured grids, J. Comput. Phys. 214 (2006) 96–121. doi:10.1016/j.jcp.2005.09.008.
[139] 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 AIAAASMESAEASEE Jt. Propuls. Conf., 2011.
[140] T. Dittmann, G. Jacobs, W.-S. Don, Dispersion of a cloud of particles by a moving shock: Effects of shape, angle of incidence and aspect ratio, in: American Institute of Aeronautics and Astronautics, n.d. http://arc.aiaa.org/doi/abs/10.2514/6.2011-441.
[141] S. Balachandar, J.K. Eaton, Turbulent Dispersed Multiphase Flow, Annu. Rev. Fluid Mech. 42 (2010) 111–133. doi:10.1146/annurev.fluid.010908.165243.
[142] A. Haselbacher, M. Parmar, S. Balachandar, Modeling of the unsteady force for shock-particle interaction, Shock Waves. 19 (2009) 317–329. doi:10.1007/s00193-009-0206-x.
[143] R. Hambli, Apparent damage accumulation in cancellous bone using neural networks., J. Mech. Behav. Biomed. Mater. 4 (2011) 868–78. doi:10.1016/j.jmbbm.2011.03.002.

[144] R. Hambli, Multiscale prediction of crack density and crack length accumulation in trabecular bone based on neural networks and finite element simulation, Int. J. Numer. Methods Biomed. Eng. 27 (2011) 461–475. doi:10.1002/cnm.
[145] R. Hambli, H. Katerchi, C.-L. Benhamou, Multiscale methodology for bone remodelling simulation using coupled finite element and neural network computation., Biomech. Model. Mechanobiol. 10 (2011) 133–145. doi:10.1007/s10237-010-0222-x.
[146] S. Jung, J. Ghaboussi, Neural network constitutive model for rate-dependent materials, Comput. Struct. 84 (2006) 955–963. doi:10.1016/j.compstruc.2006.02.015.
[147] O. Sen, N.J. Gaul, K.K. Choi, G. Jacobs, H.S. Udaykumar, Evaluation of Kriging Based Surrogate Models Constructed from Mesoscale Computations of Shock Interaction with Particles, J. Comput. Phys. (n.d.).
[148] O. Sen, S. Davis, G. Jacobs, H.S. Udaykumar, Meso-scale simulation of shocked particle laden flows and construction of metamodels, in: APS Shock Compression Condens. Matter Meet. Abstr., 2015: p. 6002.
[149] S. Davis, O. Sen, G. Jacobs, H.S. Udaykumar, Multiscale Modeling of Particles Embedded in High Speed Flows, in: APS Shock Compression Condens. Matter Meet. Abstr., 2015: p. 4001.
[150] S. Davis, O. Sen, G. Jacobs, H.S. Udaykumar, Coupling of micro-scale and macro-scale Eulerian-Lagrangian models for the computation of shocked particle-laden flows, in: ASME 2013 Int. Mech. Eng. Congr. Expo., American Society of Mechanical Engineers, 2013: pp. V07AT08A011–V07AT08A011.
[151] E. Roux, P.-O. Bouchard, Kriging metamodel global optimization of clinching joining processes accounting for ductile damage, J. Mater. Process. Technol. 213 (2013) 1038–1047. doi:10.1016/j.jmatprotec.2013.01.018.
[152] S. Volpi, M. Diez, N.J. Gaul, H. Song, U. Iemma, K.K. Choi, E.F. Campana, F. Stern, Development and validation of a dynamic metamodel based on stochastic radial basis functions and uncertainty quantification, Struct. Multidiscip. Optim. 51 (2014) 347–368. doi:10.1007/s00158-014-1128-5.
[153] 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 Conf. Struct. Multidiscip. Optim., 2013: pp. 1–10.
[154] C.T. Crowe, Multiphase flows with droplets and particles, (1988).
[155] E. Weinan, The Heterogeneous Multiscale Method and the "Equation free" approach to multiscale Modeling, n.d. https://web.math.princeton.edu/ weinan/pdf files/HMM-eqfree.pdf (accessed November 3, 2014).
[156] G.B. Jacobs, J.S. Hesthaven, Implicit-explicit time integration of a high-order particle-in-cell method with hyperbolic divergence cleaning, Comput. Phys. Commun. 180 (2009) 1760–1767. doi:10.1016/j.cpc.2009.05.020.
[157] 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, J. Comput. Phys. 228 (2009) 1365–1379. doi:10.1016/j.jcp.2008.10.037.
[158] O. Sen, S. Davis, G. Jacobs, H.S. Udaykumar, Evaluation of convergence behavior of metamodeling techniques for bridging scales in multi-scale multimaterial simulation, J. Comput. Phys. 294 (2015) 585–604.
[159] O. Sen, H.S. Udaykumar, G. Jacobs, Effect of Microstructural Geometry for Computing Closure Models in Multiscale Modeling of Shocked Particle Laden Flow, in: APS Meet. Abstr., 2016.
[160] A. Haselbacher, M. Parmar, S. Balachandar, Improved Drag Correlation for Spheres and Application to Shock-Tube Experiments, Aiaa J. 48 (2010) 1273–1276. doi:10.2514/1.J050161.
[161] A.I.J. Forrester, A. Sóbester, A.J. Keane, Multi-fidelity optimization via surrogate modelling, Proc. R. Soc. Lond. Math. Phys. Eng. Sci. 463 (2007) 3251–3269. doi:10.1098/rspa.2007.1900.
[162] Z.-H. Han, R. Zimmermann, S. Goretz, A New Cokriging Method for Variable-Fidelity Surrogate Modeling of Aerodynamic Data, in: 48th AIAA Aerosp. Sci. Meet. New Horiz. Forum Aerosp. Expo., American Institute of Aeronautics and Astronautics, n.d. http://arc.aiaa.org/doi/abs/10.2514/6.2010-1225 (accessed October 30, 2016).
[163] S. Koziel, S. Ogurtsov, I. Couckuyt, T. Dhaene, Variable-Fidelity Electromagnetic Simulations and Co-Kriging for Accurate Modeling of Antennas, IEEE Trans. Antennas Propag. 61 (2013) 1301–1308. doi:10.1109/TAP.2012.2231924.
[164] B. Choi, J. Lee, T. Park, Alternate Learning Algorithm on Multilayer Perceptrons, (2006) 63–67.
[165] G.E. Fasshauer, M.J. Mccourt, Stable Evaluation of Gaussiam Radial Basis Function Interpolants, Siam J Sci Comput. 34 (2012).
[166] N. Benoudjit, M. Verleysen, On the Kernel Widths in Radial-Basis Function Networks, Neural Process. Lett. 18 (2003) 139–154.

[167] A. Alexandridis, H. Sarimveis, K. Ninos, A Radial Basis Function network training algorithm using a non-symmetric partition of the input space – Application to a Model Predictive Control configuration, Adv. Eng. Softw. 42 (2011) 830–837. doi:10.1016/j.advengsoft.2011.05.026.
[168] R. Ahmad, H. Jamaluddin, Radial Basis Function (RBF) for Non–Linear Dynamic System Identification, J. Teknol. 36 (2002) 39–54. doi:10.11113/jt.v36.561.