KERNEL OPTIMIZATION FOR LOW-RANK MULTIFIDELITY ALGORITHMS

Mani Razi,[1] Robert M. Kirby,[2] & Akil Narayan[1,*]

[1]Scientific Computing and Imaging (SCI) Institute, Salt Lake City, Utah, 84112, USA

[2]School of Computing, and Scientific Computing and Imaging (SCI) Institute, Salt Lake City, Utah, 84112, USA

*Address all correspondence to: Akil Narayan, Department of Mathematics, and Scientific Computing and Imaging (SCI) Institute, University of Utah, Salt Lake City, Utah, 84112, USA, E-mail: akil@sci.utah.edu

Original Manuscript Submitted: 12/9/2019; Final Draft Received: 9/6/2020

One of the major challenges for low-rank multifidelity (MF) approaches is the assumption that low-fidelity (LF) and high-fidelity (HF) models admit "similar" low-rank kernel representations. Low-rank MF methods have traditionally attempted to exploit low-rank representations of linear kernels, which are kernel functions of the form $K(u,v) = v^T u$ for vectors u and v. However, such linear kernels may not be able to capture low-rank behavior, and they may admit LF and HF kernels that are not similar. Such a situation renders a naive approach to low-rank MF procedures ineffective. In this paper, we propose a novel approach for the selection of a near-optimal kernel function for use in low-rank MF methods. The proposed framework is a two-step strategy wherein (1) hyperparameters of a library of kernel functions are optimized, and (2) a particular combination of the optimized kernels is selected, through either a convex mixture (additive kernel approach) or through a data-driven optimization (adaptive kernel approach). The two resulting methods for this generalized framework both utilize only the available inexpensive low-fidelity data and thus no evaluation of high-fidelity simulation model is needed until a kernel is chosen. These proposed approaches are tested on five nontrivial real-world problems including multifidelity surrogate modeling for one- and two-species molecular systems, gravitational many-body problem, associating polymer networks, plasmonic nanoparticle arrays, and an incompressible flow in channels with stenosis. The results for these numerical experiments demonstrate the numerical stability efficiency of both proposed kernel function selection procedures, as well as high accuracy of their resultant predictive models for estimation of quantities of interest. Comparisons against standard linear kernel procedures also demonstrate increased accuracy of the optimized kernel approaches.

KEY WORDS: *kernel function optimization, low-rank approximation, multifidelity models, hyperparameter optimization, parameter estimation, surrogate modeling*

1. INTRODUCTION

Multifidelity (MF) approaches have become an active area of research in the literature for uncertainty quantification and several approaches have been introduced with widespread application in many areas of science and engineering. For example, such approaches have been utilized for molecular dynamics simulations, chaotic motion, finite element analysis of acoustic problems, finite volume analysis of heat-driven cavity flow, computational aerodynamics, particle-laden turbulence, multidisciplinary design optimization of wings, airfoil optimization, and plasmonics; see [1–9], respectively. MF models are computational meta-algorithms that are used to simultaneously leverage the strengths of low-fidelity models (i.e., low cost) and high-fidelity models (i.e., high accuracy).

The setup we consider in this paper is that we are given an ensemble of models, each of which has an identical set of parameters. Note that this can be a strong assumption, as many times differing models have differing parameter sets. Situations when models have different parameter sets are common, but we demonstrate that there are numerous examples in scientific computing when our assumption is valid. Trusted but computationally expensive models are called high-fidelity (HF) models. Relatively inexpensive computational models with low accuracy (relative to some baseline truth or HF models) are low-fidelity models. These characteristics that give rise to low- versus high-fidelity models usually reflect a trade-off between computational cost and accuracy. For example, these model ensembles are frequently produced by varying a discretization coarsening parameter, by making geometrical simplifications, or by making simplifications in physics.

Multilevel Monte Carlo-type methods are one popular class of MF methods, and several other types of strategies exist [10–12]. These methods are developed based on the approximation of correlation between low- and high-fidelity models with a type of reproducing kernel, or Gramian matrix. Kernel functions, which are used to construct Gramian or covariance matrices for different applications are, in essence, high-dimensional generalized inner product functions. These functions are applied to map the original data into higher-dimensional spaces and often act as the engine for automatic feature engineering. In this paper we consider a different branch of MF methods: low-rank MF methods [2]. Low-rank MF methods compute an inexpensive low-rank approximation to HF models thorough exploration of the low-fidelity model. Here, the low-rank property means that emulators have a small number of degrees of freedom.

In the context of model discrepancy, "fidelity" can also be interpreted as different values of quantized parameters that produce drastic changes in the way simulation models behave. This new form of quantized fidelity level has been recently studied for problems such as different particle numbers in plasmonic nanoparticle arrays, different number of species for competitive ecological systems, and different levels of density for associating polymer networks [9]. Although the low-fidelity models for most of aforementioned examples are typically obtained by computational simplification, data-fitted models can also be used in the context of multifidelity models.

The core idea in low-rank multifidelity methods is to utilize what is called the "kernel trick" in machine learning, which uses kernel functions to map model realizations to a kind of correlation. Until now, only linear kernels have been used in the literature for low-rank MF methods [1–3,9,13,14], although alternative kernels are popular for other kinds of MF approaches [15,16]. Linear kernel functions are those of the form $K(u, v) = v^T u$ for vectors u and v. The use of the linear kernel function in low-rank MF methods empirically produces accurate predictions. However, use of this kernel in MF methods has three major drawbacks:

- The rank of the kernel matrix equals the size of the model realization vectors. The result is that the number of usable HF simulations is limited to this rank.
- The linear kernel matrix can be ill-conditioned. Although such an event could indicate exploitable low-rank structure, it also results in inaccurate finite-precision computations. This limitation can sometimes require users to employ *ad hoc* approaches [9] to mitigate the stability issue.
- Low-rank structure of models may not be detectable using a linear kernel, but is perhaps detectable using an alternative kernel.

In this paper we propose and empirically explore a new procedure for selecting a kernel in low-rank MF methods. Kernel function selection strategies have been already used in the context of predictive estimation of the models' fidelity, e.g., [17]. To the best of the authors' knowledge, the details of the approach employed in this paper are distinct from existing kernel learning/selection strategies in the literature. The procedure in this paper is motivated by low-rank MF algorithms, which have different desiderata than standard kernel learning approaches. However, our approach does exhibit similarities to existing methods: we optimize hyperparameters in kernels, and optimize over multiple families of well-known kernels. Overall, our approach performs this optimization by attempting to minimize the discrepancy with respect to the linear kernel matrix, and a numerical rank regularization is employed to promote stability. We propose two approaches: the additive kernel approach constructs a kernel via a convex combination of hyperparameter-optimized kernel functions. The second, the adaptive kernel approach, selects a single hyperparameter-optimized kernel from the library using an inexpensive high-fidelity verification. Both new approaches perform optimization on LF data, and do not require any additional HF evaluations compared to the standard linear kernel approach, which relies on the use of a simple inner product for the construction of the Gramian matrix.

We focus on the bifidelity situation in this paper: We assume throughout well-defined low- and high-fidelity models exist, and that the low-fidelity model is inexpensive but also insufficiently accurate for the purpose of prediction. We assume that the HF model is an ideal computational simulation model that requires an onerous amount of time. Both LF and HF models are assumed to share a common set of tunable parameters (such as constitutive model constants or coefficients, and are often modeled as random variables for uncertainty quantification purposes). With a bifidelity template, the approaches in this paper can be extended to multifidelity [1,3].

To demonstrate the effectiveness of our approach, we test the new procedures on several nontrivial examples:

1. One- and two-species molecular dynamic (MD) simulation models: determination of MD model's statistical properties.

2. Polymer networks: investigation of the resultant strain of a graph of links in a polymer chain under a constant stress.

3. n-Body galaxy model: prediction of total energy and mean velocity and distance for the system.

4. Incompressible flow in two-dimensional channels with stenosis: estimation of velocity and wall shear stress profiles.

5. Plasmonic nano-particle arrays: estimation of extinction and scattering efficiencies for Vogel spiral configurations with a different number of nanoparticles.

This manuscript is organized as follows: In the following two sections, the theoretical foundation of both multifidelity models and the proposed kernel selection approaches is provided. The sections are followed by the results and discussion of five canonical test problems. Finally, concluding remarks are presented as the final section of this paper, which is Section 5.

2. LOW-RANK APPROXIMATION USING MULTIPLE LEVELS OF FIDELITY

Let a low-fidelity model, g_L, and a high-fidelity model, g_H, be given. These two models are functions of input parameters p. Consider a wave model with the wave speed, p, as a parameter. For a given parameter value p, g_L, and g_H output vectors,

$$D \subset \mathbb{R}^q, \quad g_L : D \to \mathbb{R}^m, \quad g_H : D \to \mathbb{R}^M.$$

Note that usually $m \neq M$, and in general we assume $m \ll M$ (for example, a coarse m-degree-of-freedom model, versus a refined M-degree-of-freedom model). We assume that $g_H(p)$ is a good predictor of some baseline "truth", but that $g_L(p)$ is a poor predictor. Qualitatively, g_L is inexpensive but inaccurate, and g_H is accurate but expensive.

The problem we consider is to construct an accurate emulator $\tilde{g}_H$ for g_H whose accuracy is comparable to g_H but whose cost is comparable to g_L, i.e.,

$$\mathrm{Cost}(\tilde{g}_H) \sim \mathrm{Cost}(g_L), \quad g_H(p) \approx \tilde{g}_H(p), \quad \forall\, p \in D.$$

The focus of this paper is on such a construction of $\tilde{g}_H$. The end-user goal for this emulator may be for computation of statistics of g_H (by modeling p as a random variable), for determining sensitivity of g_H to p, inference (identification of appropriate p to match data). Any of the end-user goals requires many queries of the g_H for several values of p, which can be prohibitively expensive if g_H is very expensive to query. This motivates the need to construct this emulator.

2.1 Low-Rank Multifidelity Model Construction

We describe the core idea behind low-rank MF methods, as outlined in [2]. The construction of $\tilde{g}_H$ proceeds in two steps. In the first step, the parameter space D is explored via the low-fidelity model to identify "important" points in parameter space. Since by assumption g_L is an inexpensive emulator, this is practically feasible. The procedure for accomplishing this exploration is essentially numerical linear algebra.

First we densely sample D with the ensemble $\{p_j\}_{j=1}^N$, where $N \gg 1$. These points may be selected in any reasonable way: as Monte Carlo samples, equispaced samples, low-discrepancy samples, etc. We next construct a Gramian matrix G_L with entries

$$(G_L)_{l,j} = K(g_L(p_l), g_L(p_j)), \quad l, j = 1, \dots, N, \quad K(u,v) = \langle u, v \rangle, \tag{1}$$

where $\langle \cdot, \cdot \rangle$ is the standard Euclidean inner product on $\mathbb{R}^m$. This Gramian is positive semidefinite, but is not definite if $N < m$. In this case, the first m steps of a pivoted Cholesky decomposition can be completed:

$$PG_L P^T = LL^T, \tag{2}$$

where L is a lower-triangular matrix and is zero for rows $m+1, \dots, N$. The permutation matrix $P \in \mathbb{R}^{N \times N}$ determines an "important" point selection; in particular, the vector

$$z := P \begin{pmatrix} 1 \\ 2 \\ \vdots \\ N \end{pmatrix},$$

is used to define an ordering $p_{z_1}, p_{z_2}, \dots, p_{z_N}$ of the N parameter points. For any $n \leq m$, the point set

$$\widehat{D} = \{p_{z_1}, \dots, p_{z_n}\} \subset D, \tag{3}$$

are the "important" points. For later purposes we will also require an $n \times n$ matrix $\widehat{G}_L$, which is a z-sliced version of G_L:

$$\left(\widehat{G}_L\right)_{i,j} = (G_L)_{z_i, z_j}, \quad i, j = 1, \dots, n.$$

The identification of $\widehat{D}$ is the main purpose of the first step. The second step will require evaluations of the HF model on $\widehat{D}$, so that n is usually chosen based on the available computational budget.

This approach for selecting "important" points via pivots is computationally efficient [1,3] and is equivalent to the pivoted samples selected by a column-pivoted QR decomposition of a matrix of model realizations [2]. Pivots from an LU factorization are also used in the area of computational fluid dynamics and heat transfer [18] similarly. Alternative statistical strategies such as leverage-score sampling methods [19] and group matching methods [20,21] are also useful for the selection of optimal/important set of sampling points. For the purposes of simplicity in this paper, we focus exclusively on the Cholesky approach for identifying points. The second step of the MF procedure first computes the HF ensemble,

$$\{g_H(p_{z_i})\}_{i=1}^n,$$

which is feasible if n is small. The MF approximation is built as a linear expansion over this ensemble:

$$g_H(p) \approx \widetilde{g}_H(p) := \sum_{l=1}^{n} g_H(p_{z_l}) c_l(p), \tag{4}$$

where $\{c_l(p)\}_{l=1}^n$ are coefficients in a vector $c(p)$ that is computed via a least-squares projection on the low-fidelity model:

$$\widehat{G}_L c(p) = \begin{pmatrix} K(g_L(p), g_L(p_{z_1})) \\ \vdots \\ K(g_L(p), g_L(p_{z_n})) \end{pmatrix}. \tag{5}$$

Note that the right-hand side of the above equation is computable immediately simply by evaluating the $g_L(p)$, i.e., the low-fidelity model. The least-squares projection coefficients can be obtained as

$$c(p) = \widehat{G}_L^{-1} \begin{pmatrix} K(g_L(p), g_L(p_{z_1})) \\ \vdots \\ K(g_L(p), g_L(p_{z_n})) \end{pmatrix}. \tag{6}$$

Note in particular that the emulator is interpolatory on the trained samples $\{p_i\}_{i=1}^n$, i.e., $g_H(p_i) = \widetilde{g}_H(p_i)$ for all $i = 1, \ldots, n$. This property can be directly deduced by noting in Eq. (5) that if $p = p_{z_i}$, then the right-hand side is exactly a column of $\widehat{G}_L$, and so all the $c_j(p_i)$ coefficients vanish, except for $c_j(p_j) = 1$. This procedure can be generalized to include more levels of fidelity [2,3], which is advantageous when three or more models exist.

In summary, the first stage of the algorithm is "offline", which requires the computation of $N \gg 1$ realizations of the LF model. Some (inexpensive) linear algebra identifies a small set of $n \ll N$ parameter values where the HF model is evaluated in step 2. Then for each fixed $p \in D$, the low-rank MF surrogate in Eq. (4) is computed by first evaluating $g_L(p)$, and then by solving (5) for the vector $c(p)$.

While the MF surrogate $\widetilde{g}_H$ certainly achieves the cost requirement (evaluation requires only low-fidelity effort), understanding its accuracy is more subtle. The authors in [2] provide abstract error certification in general Hilbert spaces; however, the analysis is not useful in practice and the bound derived is not computable. In [14] a more practical analysis demonstrates that this surrogate is accurate when the LF and HF models correspond to time integration solvers with coarse and fine time steps, respectively. Finally, the authors in [4] provide a statistical strategy for computationally certifying error committed by $\widetilde{g}_H$.

2.2 Limitations and Drawbacks

Assuming $\widetilde{g}_H$ is accurate, the major limitations of this approach are described in Section 1. We give a more detailed discussion here. First, recall that since G_L has rank m at most, then when $m \ll N$ only the first $n = m$ pivots selected in the decomposition (2) are informative (the remaining $N - m$ pivots are essentially selected at random due to floating-point roundoff errors). To see why this can become an issue, consider g_L as a scalar output of a model (for example, the section lift coefficient in an aerodynamics model). Then the MF procedure only produces a single informative parameter value. This prevents the procedure from taking advantage of additional HF model runs when such runs are computationally feasible.

A second limitation is in the stability of the linear solver required in Eq. (5). When G_L is a low-rank matrix, then its singular values decay quickly, and the sliced matrix $\widehat{G}_L$ is ill-conditioned. This can numerically pollute the MF surrogate (4).

A final limitation that we will discuss is more subtle: one requirement for success of this procedure is that G_L is "similar" to G_H, where G_H is a HF Gramian defined similarly as Eq. (1) but with g_H instead of g_L. The choice of K in Eq. (1) to take uncentered inner products may not reveal similarity of G_L and G_H, but choosing a different function K may improve this approximation.

3. KERNEL FUNCTION SELECTION

As discussed in Section 2, kernel functions serve a key role in construction of the Gramian matrix, which is a building block of the low-rank multifidelity approximation approach. Equation (1) shows the appearance of the kernel K in development of the low-rank MF model. Section 2 shows the construction of the MF model with a linear kernel K as shown in Eq. (1). However, Section 2.2 lists some limitations of using this linear kernel. The main contribution of this paper is to mitigate some of these shortcomings by employing an alternative kernel function.

Note that there are several types of kernel functions that one could choose as an alternative to linear kernels. For simplicity, we consider several popular *radial* kernels as alternatives [a kernel K is radial if $K(u, v) = F(\|u - v\|)$ for some function F]. We list these alternative kernels in Table 1. Note that this list is not meant to be exhaustive, and only serves as a reasonably diverse set of kernels to demonstrate our approach.

Once a kernel is selected, say of index i, the entire procedure of Section 2 can be completed, by replacing K in Eq. (1) with K_i (for example, the procedure remains unchanged if we consider K_1 since this is a linear kernel in Table 1). There is one new detail in Table 1, namely that most of the kernels are now functions of an additional set of hyperparameters h, where the number of hyperparameters depends on the family. The hyperparameters are frequently parameters that affect the general shape of the kernel, and it is well-known in both the computational mathematics and statistics community that a good choice for these parameters is essential to building both a stable and accurate model [22,23]. Thus, we have complicated our procedure since we must now make additional choices for the family

TABLE 1: Kernel function library for $K(u, v; h)$. u and v are input vectors, while h is a vector of hyperparameters. $\| \cdot \|$ denotes the Euclidean length of an input vector, and $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product. Kernel functions 2–7 are two types of radial kernel functions

Index i	$\dim h$	Kernel type	
1	0	Linear	$K_i = \langle u, v \rangle$
2	1	Exponential	$K_i = \exp\left(-\frac{\|u-v\|}{h_1}\right)$
3	1	Squared exponential	$K_i = \exp\left(-\frac{\|u-v\|^2}{2h_1}\right)$
4	2	Rational quadratic	$K_i = \left(\frac{1+\|u-v\|^2}{2h_1^2 h_2}\right)^{-h_2}$
5	1	Matern 3/2	$K_i = \left(1 + \frac{\sqrt{3}}{h_1}\|u-v\|\right) \exp\left(-\frac{\sqrt{3}}{h_1}\|u-v\|\right)$
6	1	Matern 5/2	$K_i = \left(1 + \frac{\sqrt{5}}{h_1}\|u-v\| + \frac{5}{3h_1^2}\|u-v\|^2\right) \exp\left(-\frac{\sqrt{5}}{h_1}\|u-v\|\right)$
7	2	Compact RBF	$K_i = \max\left[0, 1 - \left(\frac{1}{h_1}\|u-v\|\right)^{h_2} \exp\left(-\frac{\|u-v\|^2}{2h_1^2}\right)\right]$

of kernel (quantified by the index i in Table 1), and we must choose values for the hyperparameters h. We first discuss the task of choosing values for the hyperparameters h.

3.1 Hyperparameter Selection

In this section, let i be a *fixed* family index associated to one of the kernel families in Table 1. Our task is to determine an appropriate value of the hyperparameter h. We formulate this selection as a simple optimization problem, where we impose that the following requirements should be met:

- To maintain accuracy, the optimized kernel should behave like the linear kernel K_1 (since when the Gramian G_L is well-conditioned, results in the literature [2,14] show that this procedure is effective).
- To maintain stability, the Gramian G_L resulting from a particular choice of kernel should reasonably be well-conditioned.

Our desire to maintain accuracy of the MF emulator is not always aligned with ensuring that a selected kernel behaves like the linear kernel. However, the current success of the linear kernel in the literature suggests that this may promote some accuracy [1,2,9]. In addition, the main purpose of the accuracy requirement is to ensure that the chosen hyperparameters do not force the kernel function to concentrate to a Dirac delta function: if the kernel focuses to the Dirac delta function, then the MF approximation will suffer poor accuracy. The choice of proximity to the linear kernel is thus just one choice of regularizer; we implement this as the Frobenius norm proximity to the linear kernel Gramian.

The need to maintain stability comes from the fact that solving (5) should be numerically stable. When the Gramian matrix is rank-deficient, the stability consideration regularizes the matrix so that its rank deficiency is mitigated. We implement this as the numerical (or stable) rank of the Gramian.

In summary, with $i \geq 2$ fixed, let $G_L^i(h)$ denote $N \times N$ the low-fidelity Gramian matrix associated with using $K = K_i(h)$ in Eq. (1). That is, $G_L^i(h)$ has entries

$$\left(G_L^i(h)\right)_{j,k} = K_i(u_L(p_j), u_L(p_k); h).$$

We choose a value for the hyperparameters h via the optimization problem

$$h^* = \arg\min_h f_{\text{obj}}(h) := \left\| G_L^1 - G_L^i(h) \right\|_F + \lambda \frac{1}{\sqrt{\text{srank}\big(G_L^i(h)\big)}}, \tag{7}$$

where λ is tunable parameter, $\| \cdot \|_F$ is the Frobenius norm on matrices, and $\text{srank}(\cdot)$ is the numerical/stable rank:

$$\text{srank}(A) = \frac{\|A\|_F^2}{\|A\|_2^2}.$$

The objective function f_{obj} in Eq. (7) thus seeks to simultaneously promote proximity to the linear Gramian ("accuracy") and maximization of the numerical rank ("stability"). The value of λ can be obtained by empirical testing. In practice we judge that this procedure finds an effective kernel if $h^* \sim \lambda/\sqrt{n}$, corresponding to a case when the error in the adaptive kernel Gramian balances with the numerical rank. However, we expect optimal values of this parameter to be problem-dependent. For all problems tested in this paper, we found that $\lambda = 0.1$ provided good results, obtained through grid search and evaluating by computing errors on testing data. This value of λ was computed using a logarithmic grid search over the first three test problems in this manuscript. Using this approach, λ is a regularization parameter that promotes the numerical stability of the solution, and the grid search we have employed is one of the standard approaches in machine learning [24,25] for tuning such a regularization parameter.

As often the number of hyperparameters is small (cf. Table 1) and the computational cost of high-fidelity model evaluation is significantly higher than that of the low-fidelity model, the process of hyperparameter optimization is inexpensive in contrast. Note that the optimization (7) operates *only* on the low-fidelity model, and the objective function is simple and efficient to evaluate if all the low-fidelity data are stored.

In order to deal with this problem, we propose two approaches that generalize the kernel function selection framework. For both of these proposed approaches we are using a heuristic optimization algorithm called the particle swarm algorithm [26,27] to minimize f_{obj} in Eq. (7). Next, this algorithm is briefly discussed.

3.2 Particle Swarm Optimization

The optimization problem (7) has $\dim h$ design variables, and for our choices of kernels this number is relatively small (see Table 1). Nevertheless, gradient-based optimization algorithms can perform poorly when the initial guess is far away from a basin of attraction. To mitigate this issue, we employ particle swarm optimization [26,27] to compute a reasonable initial guess, and subsequently perform a gradient-based interior point algorithm.

Particles swarm optimization (PSO) is a velocity-based evolutionary optimization approach, and was initially developed for modeling social behaviors in bird flocks using topological neighborhoods [28]. PSO is initialized using an ensemble of random points in the design/search space, with each ensemble member called a particle. For each particle there is an assigned position (design variable value) and velocity. The initial selection of particle positions and velocities is often random from a uniform distribution. The remaining portion of the algorithm updates particle positions and velocities using a notion of *fitness* of a particle, i.e., an estimate of its optimality relative to the ensemble.

Let $\{h_i\}_{i=1}^S$ and $\{v_i\}_{i=1}^S$ be the position and velocity, respectively, of the ith particle. For $n \in \mathbb{N}$, let h_i^n denote the position of the ith particle at PSO iteration n. At $n = 1$, all positions and velocities are randomly initialized. PSO updates proceed by

$$\begin{aligned} v_i^{n+1} &= v_i^n + k_1\rho(B_i^n - h_i^n) + k_2\gamma\big(B_g^n - h_i^n\big), \\ h_i^{n+1} &= h_i^n + v_i^{n+1}, \end{aligned} \tag{8}$$

where B_i^n and B_g^n denote the best solution for the ith particle in iterations $1, ..., n$, and the group's best solution for iterations $1, \ldots, n$, respectively. That is,

$$B_i^n := \min_{j\in[n]} f_{\text{obj}}(h_i^j), \quad B_g^n := \min_{(i,j)\in[S]\times[n]} f_{\text{obj}}(h_i^j), \quad [n] := \{1, 2, \ldots, n\}.$$

The parameters ρ and γ in Eq. (8) are uniform random variables on $[0, 1]$ (independent at each time and for each particle) and k_1 and k_2 are tunable parameters. We apply standard velocity clamping procedures to ensure velocities remain bounded and particles do not leave the design space: If some particle's velocity exceeds a maximum allowed velocity limit, it is set to the maximum allowed value of velocity [29]. PSO iterations continue until a stopping criterion is triggered. In our case, we limit the maximum number of iterations and lack of improvement in the group's best solution. There are many variations of PSO; for the purpose of this study we employ the MATLAB toolbox "particleswarm solver." This algorithm has been shown to be faster than standard evolutionary algorithms in approaching the vicinity of the minima [30,31].

However, PSO is slow when it comes to converging to a local minimum once a basin of attraction is reached. Therefore, we use lose criteria to terminate the PSO iterations, and subsequently use the best particle as the initial guess into a gradient-based interior point algorithm (with finite difference approximated gradient); the output of the gradient-based algorithm is our computed value of h^* in Eq. (7).

3.3 Additive Kernel Approach

We have discussed in Sections 3.1 and 3.2 how we select the hyperparameter value h for each kernel index i in Table 1. This section and the following section propose two strategies for choosing an overall kernel K for use in the MF procedure of Section 2.1.

Our first approach, the additive kernel approach, chooses an overall kernel K as an additive mixture of all kernels in the library:

$$K(u,v) = \sum_{i=1}^{L} w_i K_i(u,v;h_i^*), \quad (w_1, \ldots w_L) \in \left\{ x \in \mathbb{R}^L \;\middle|\; \sum_{\ell=1}^{L} x_\ell = 1, \;\; x_\ell \in [0,1] \;\; \forall i \right\}, \tag{9}$$

where L, h_i^*, and w_i are the number of kernel functions in the library ($L = 7$ in this paper), the ith optimized hyperparameter value h^* output from Eq. (7), and its corresponding weight. The optimizations associated with computing the quantities in Eq. (9) are convex, so several types of algorithms are appropriate. We again employ PSO for an initial guess, followed by an interior-point algorithm for accelerated convergence to a minimum.

The full MF procedure is exactly as described in Section 2.1, with K in Eq. (1) the kernel computed in Eq. (9). We emphasize again that the choice of kernel only depends on low-fidelity data. Only after a kernel function is chosen do we query the HF model through the procedure in Section 2.1.

3.4 Adaptive Approach for the Selection of an Optimal Kernel Function

Our second procedure for selecting a kernel is the adaptive kernel approach. In this procedure, we select only one kernel K_i as the kernel of choice in the MF procedure. However, the index of this choice i depends on the HF simulation budget n. In brief, with n fixed, we select i as the kernel index that results in the best approximation of a size-n least-squares approximation of the LF model.

In detail, we first fix i and construct $\widetilde{g}_{L,i}$ as an emulator for the LF model using kernel i:

$$\widetilde{g}_L(p) = \sum_{l=1}^{n} g_L\big(p_{z_{l,i}}\big) c_{l,i}(p), \tag{10}$$

where $\{z_{l,i}\}_{l=1}^{n}$ are the pivots selected from the Cholesky decomposition of G_L in Eq. (1) with $K = K(\cdot,\cdot;h_i^*)$. The coefficients $\{c_{l,i}\}_{l=1}^{n}$ are entries in the least-squares coefficient vector $c_i(p)$ from a linear system similar to Eq. (5):

$$\widehat{G}_L c_i(p) = \begin{pmatrix} K\big(g_L(p), g_L(p_{z_{1,i}})\big) \\ \vdots \\ K\big(g_L(p), g_L(p_{z_{n,i}})\big) \end{pmatrix},$$

where again $K = K(\cdot,\cdot;h_i^*)$. The function $\widetilde{g}_L$ is an emulator for g_L built via a least-squares approximation. One error metric for this least-squares emulator is

$$\epsilon_i := \text{med}_{j\in[N]\setminus\{z_{1,i},\ldots,z_{n,i}\}} \| g_L(p_j) - \widetilde{g}_L(p_j) \|,$$

where med denotes the median. Note that the error $\|g_L(p_j) - \widetilde{g}_L(p_j)\| = 0$ when $j \in \{p_{z_{1,i}}, \dots, p_{z_{n,i}}\}$ represents the indices of the training samples. The median has been chosen as the summary statistic for the distribution of the residuals for multifidelity emulators since the errors do not necessarily follow a normal (or symmetric) distribution and it is most often skewed. Moreover, the median is an effective statistic for summarizing distributions [32] and thus provides a more robust measure of error for our purpose. We emphasize that the data used to train the emulator are the set of parameters $\{p_{z_i}\}_{i=1}^{n}$. In this testing phase we compute errors over the set of realizations corresponding to the parameters $\{z_i\}_{i=1}^{N} \backslash \{p_{z_i}\}_{i=1}^{n}$ that the emulator was *not* trained with.

To choose the "best" kernel index i^*, we choose

$$i^* = \underset{i=1,\dots,n}{\arg\min}\, \epsilon_i.$$

Finally, we choose K in Eq. (1) as $K = K_{i^*}(h^*)$. Like the additive kernel approach, this approach analyzes only the LF model, and hence is computationally feasible. However, this approaches differs from the additive approach in that the optimal i^* is recomputed when n is changed.

4. RESULTS AND DISCUSSION

In order to demonstrate the two proposed kernel function selection methods for constructing an accurate surrogate, we consider their implementation on five nontrivial technical problems from different areas of engineering and science. These problems have been selected to provide intuition about the feasibility and benefits of the application of both approaches to real-world problems with low-dimensional output parameter space. The success of these approaches can be directly measured by the accuracy of their MF surrogate model. Our chosen error metric is the median relative error:

$$\text{error} = \underset{p \in \{p_1,\dots,p_N\} \backslash \{p_{z_1},\dots,p_{z_n}\}}{\text{median}} \frac{\|g_H(p) - \widetilde{g}_H(p)\|_{\mathbb{R}^M}}{\|g_H(p)\|_{\mathbb{R}^M}}, \tag{11}$$

where $\{z_1, \dots, z_n\}$ are the indices where the HF model was evaluated to construct $\widetilde{g}_H$. Thus, our metric is evaluated over data from a full size-N discretized parameter set, and uses the high-fidelity model g_H as the oracle "truth." We also compare results obtained with the use of the linear kernel function to emphasize the enhancement provided by the additive kernel and adaptive kernel approaches. We also measure the cost of all procedures in units of the cost of a single HF realization. We include the cost of optimization in the kernel selection process as well. Thus, our cost metric is the "effective number of high-fidelity samples," defined as

$$\begin{aligned}\text{Effective Number of High-Fidelity Samples} = {} & \text{Number of HF Samples} \\ & + \lceil * \rceil \frac{\text{Cost(Kernel Optimization)}}{\text{Cost(One HF Simulation)}},\end{aligned} \tag{12}$$

where $\lceil * \rceil$ is the ceiling function. Many of our plots will measure accuracy versus cost, using the accuracy and cost metrics identified above.

4.1 Test Problem 1: Molecular Systems

The classical computational algorithm for performing molecular dynamics (MD) simulations has been developed based upon the Lagrangian methodology of tracking particle dynamics via Newton's equations of motion. Numerical integration of these equations provides an accurate evaluation of the time evolution of a molecular system, and hence the system's quantities of interest. While there are many integration schemes available in the literature, we settle on the velocity Verlet algorithm [33] due to its popularity, reasonable accuracy, and simplicity of implementation. The Lennard-Jones (LJ) potential was used for simulating the interatomic interactions in the MD system. The potential function also has parameters ϵ; the potential well depth; and σ, the length scale for the pairwise interatomic interaction.

In the process of MD simulation using any interatomic interaction model, when the size of the integration time step is large, the major challenge is the stability of the MD integration scheme. If the time step is too large, then iterations may predict near-collocation of molecules/atoms which will lead to large (unphysical) repulsive forces and large displacements on the next time step, and this instability snowballs to future time steps. To prevent such divergence of the integration scheme, we clamp the magnitude of repulsive interactions for closely approaching atoms. For our LJ potential, this capping can be implemented straightforwardly by modifying the potential at short distance.

The MF situation we consider here is models whose fidelity is defined by the choice of integration time step. The parameters $p \in \mathbb{R}^2$ we consider are the temperature and the density of the molecular system. For every parameter value, quantities of interest are computed using the low-fidelity model (here, the MD simulation with a large time step Δt) and concatenated into an output vector. The quantities of interest we consider here are the scalar averaged total energy E and diffusion coefficient D:

$$g(p_j) = \left[\frac{E(p_j)}{\sqrt{\overline{E}}}, \frac{D(p_j)}{\sqrt{\overline{D}}}\right]^T. \tag{13}$$

We have rescaled the parameters by their averaged values over the ensemble:

$$\begin{aligned}\overline{E} &= \frac{1}{N}\sum_{j=1}^{N} \|E(p_j)\|_2^2,\\ \overline{D} &= \frac{1}{N}\sum_{j=1}^{N} \|D(p_j)\|_2^2.\end{aligned} \tag{14}$$

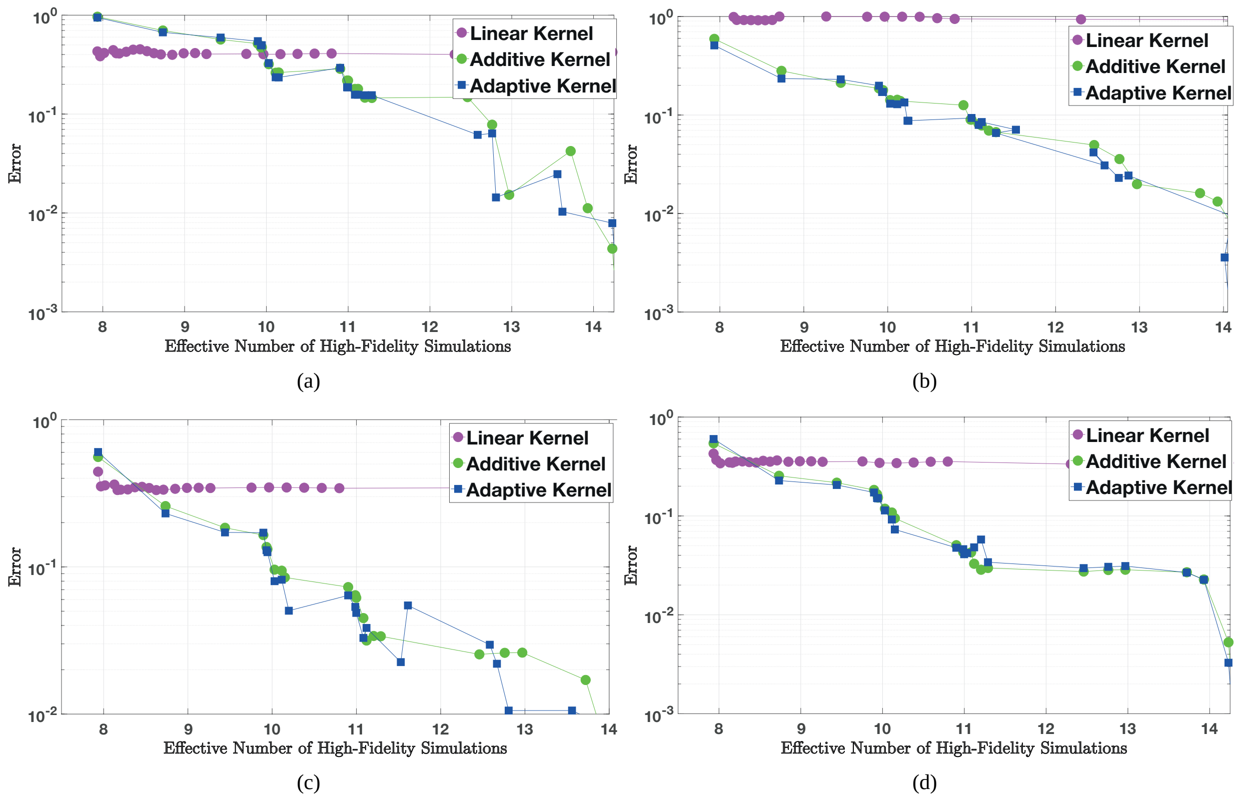
Note that in MD simulations there are other important quantities of interest, such as the radial distribution function (RDF) and mean squared displacement (MSD). For this test problem, we consider a uniform grid of temperature and density defined as $T \times \rho : [500, 1000]$ K $\times$ $[36.27, 701.29]$ kg/m^3 ($\rho^* : [0.05, 0.95]$) with $N = 114$ sample points, where ρ and ρ* are density and dimensionless density equal to Nm/V and $N\sigma^3/V$, respectively. We also assume σ, ε, m or molecular mass to be 3 Å, 1 kcal/mol, and 12.01 g/mol, respectively. Here, the boundary conditions in all sides of cubic simulation box (with the width of 27.05 Å) are considered to be periodic. Here the mass of each particle (m) and simulation box size ($L = V^{1/3}$) are set to 12.01 g/mol and 27.05 Å, respectively.

The LF model is defined by an MD simulation with time step $\Delta t = 20$ fs, and the HF model is the same MD simulation with $\Delta t = 1$ fs. In this situation, the dimension of the vector $g_L(p)$ is 2, and hence the naive MF procedure in Section 2.1 with the linear kernel can only take $n = 2$ meaningful HF samples. Figure 1 demonstrates that the additive/adaptive kernel approaches circumvent this limitation. We see that the linear kernel error stagnates since after $n = 2$ the "important" parameter values are chosen essentially randomly, and furthermore the least-squares problem (5) becomes ill-conditioned. However, both the additive and adaptive kernel approaches produce more reasonable errors, with improvements gained as one is able to invest more HF effort.

As shown in Figs. 1(a) and 1(c), both proposed approaches work well although the error is not monotonically decreasing with respect to increased effort. This lack of monotonic error is due to the fact that we use kernel evaluations of the low-fidelity model on the right-hand side of Eq. (5) in the place of the (unavailable) high-fidelity kernel evaluation. In panes (b) and (d) we show the error of the RDF and MSD quantities of interest for this procedure, which were not used in the formation of the MF surrogate. Thus, this procedure does not overfit energy and diffusion at the cost of other quantities.

Another reason for the nonmonotonic reduction of the error with respect to the increase in effective high-fidelity simulation is variable computational costs of molecular simulation, which are directly proportional to molecular density or the number of particles in the simulation box. Hence, addition of one high-fidelity sample does not linearly increase the cumulative computational cost.

Here, both proposed approaches outperform the naive multifidelity method that applies the linear kernel function. The rank deficiency of the constructed Gramian matrix for the linear kernel approach is the main reason for such poor



(a) (b)

(c) (d)

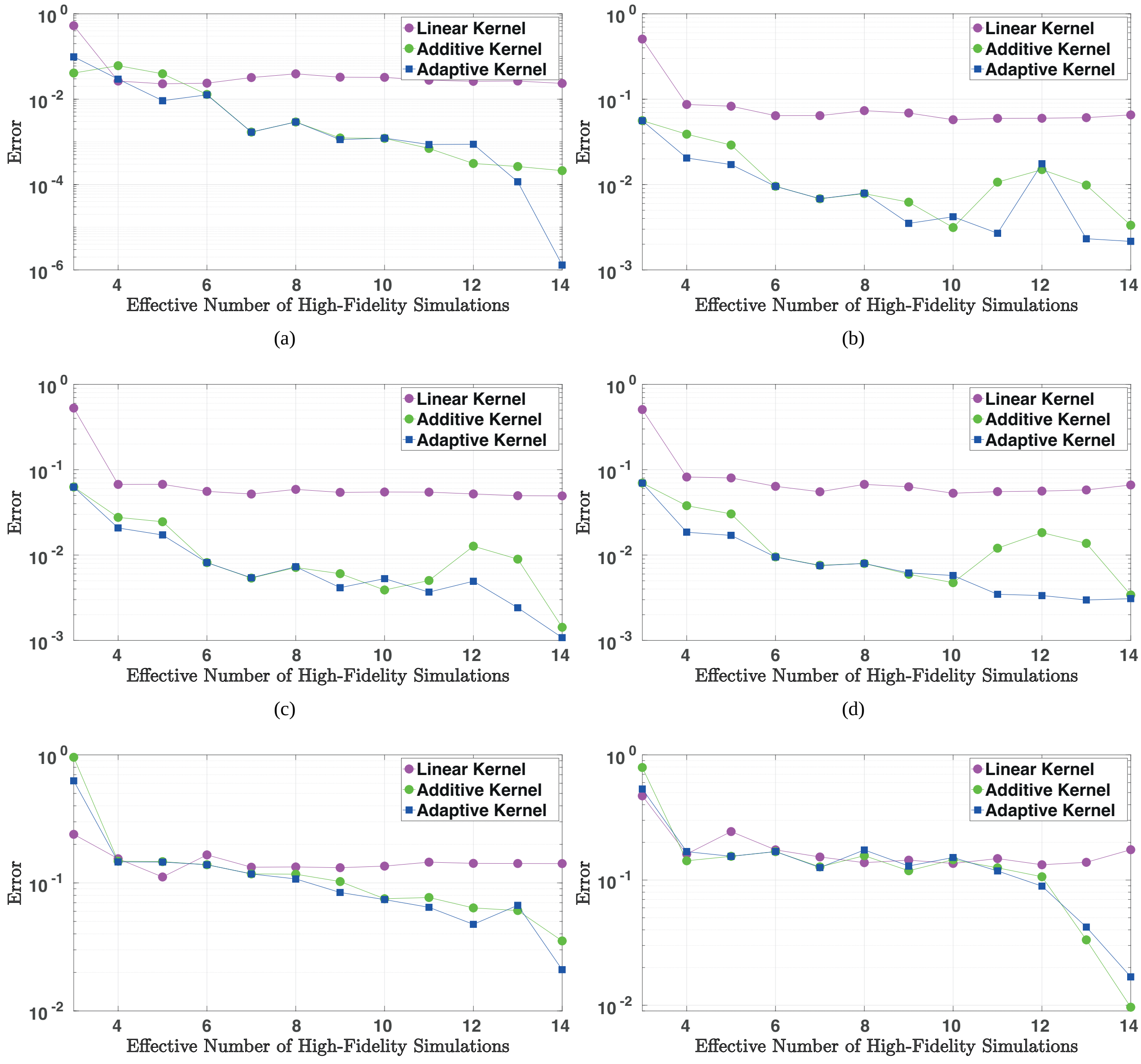
FIG. 1: Median error of the multifidelity model constructed based upon the results from the low-rank low-fidelity MD model for one-species LJ data with $\Delta t = 20$ fs in the prediction of the properties for the high-fidelity model with $\Delta t = 1$ fs; Test Problem 1. (a) Total energy, (b) RDF, (c) self-diffusion coefficient, and (d) MSD.

performance. This ill-conditioned Gramian matrix itself results from the low dimension of the model output space for low- and high-fidelity models.

Additionally, the results illustrated in Fig. 1 indicate about two orders of magnitude in error reduction when the effective number of high-fidelity simulations increases from 8 to 14. This is supporting evidence for the success of the proposed multifidelity kernel optimization approach in handling a difficult predictive task for a complex system with underlying stochastic behavior.

Next, we consider a more difficult MD simulation with a two-component glass-forming system. In this case, the three output parameters of both LF and HF models are the diffusion coefficients D_{A} and D_{B} for species A and B, along with the total energy E.

We perform output normalization similar to Eq. (14). Here, the length scale for pairwise interatomic interactions between molecules are set as $\sigma_{\mathrm{A}} = 3.5$ Å for interactions between molecules of species A, $\sigma_{\mathrm{B}} = 3.888$ Å for B-B interactions, and $\sigma_{\mathrm{AB}} = \sqrt{\sigma_{\mathrm{A}}\sigma_{\mathrm{B}}}$ for A-B interactions between both types of molecules. For this experiment, we have $N = 24$ points in the density-temperature (parameter) space. The number of molecules in the MD simulation for each species is 512. Once again, the boundary conditions are assumed to be periodic for all sides of the simulation box. The temperature and simulation box length for these sample points vary between 180 and 290 K, and 37.62 and 38.35 Å, respectively. Once again, the results depicted in Fig. 2 show high accuracy of both adaptive and additive models for prediction of the three radial distribution functions involving interactions of molecules of type A and B (R_{AA}, R_{AB}, and R_{BB}) as well as the corresponding diffusion coefficients and system total energy. However, these results indicate that, if performing 12 high-fidelity simulations is computationally affordable, the multifidelity model constructed based on the proposed adaptive kernel function selection approach is slightly more accurate compared to



(a)

(b)

(c)

(d)

(e)

(f)

FIG. 2: Median error of the multifidelity model constructed based upon the results from the low rank low-fidelity MD model for two-species LJ data with $\Delta t = 20$ fs in the prediction of the properties for the high-fidelity model with $\Delta t = 1$ fs; Test Problem 1. (a) Total energy, (b) RDF A-A interaction, (c) RDF A-B interaction, (d) RDF B-B interaction, (e) self-diffusion coefficient for species A, and (f) self-diffusion coefficient for species B.

the one constructed using the additive approach and significantly more accurate when compared to the multifidelity solution with linear kernel function.

Furthermore, as the number of particles is fixed for each sample, a more monotonic trend of convergence can be observed for this case (see Fig. 2). Due to the smaller number of training low-fidelity data points, the accuracy of the approach degrades. On the other hand, while radial distribution functions are not involved in the construction of the multifidelity models, the accuracy of the emulator for their estimation is quite good: around 1% error with only 1/3 of the high-fidelity samples.

4.2 Test Problem 2: The n-Body Galaxy Model

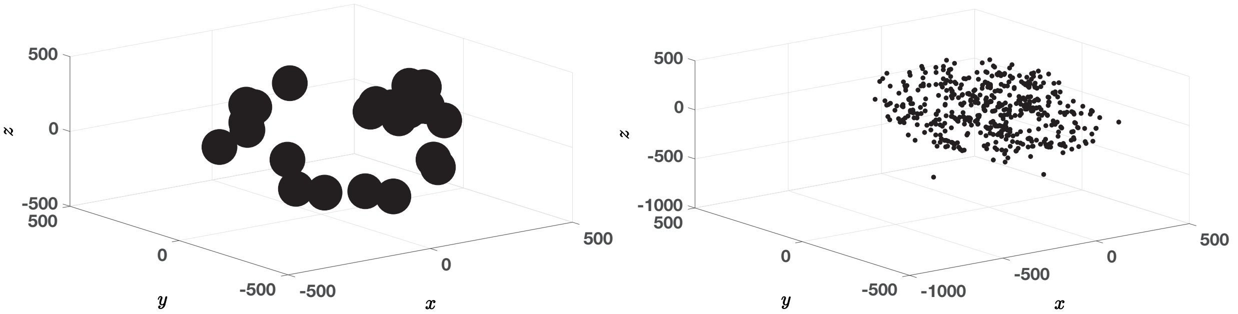
Galaxy systems often can be modeled as an n-body problem, in which the gravitational dynamics governs the interaction between objects. The interaction force between the objects can be driven from Newton's law of gravity as [34]

$$F_i = -\sum_{\substack{j=1\\ j\neq i}}^{n} \frac{G m_i m_j (r_i - r_j)}{|r_i - r_j|^3}, \tag{15}$$

where G, m_i, and r_i are the gravitational constant, ith object mass, and distance of the ith object from the origin, respectively. Considering a cutoff radius can simplify the calculation, as the force between each object and any other object outside this radius is automatically set to zero. For the purpose of this study, the GalaxSee n-body model [35] is used as the simulation module.

The parameter space for this case involves the system's total mass ($M \in [50, 500]$ solar mass) and the initial rotation ($\Gamma \in [0, 0.9]$) about a central axis in the n-body cluster imposed on the system. Fidelity here is based on quantized parameters in the model [9]: We consider galaxy systems of 25 and 500 objects as the low- and high-fidelity models, respectively. The quantities of interest for this test problem are total energy, mean distance from the origin, and the mean velocity of the objects in galaxy systems. As illustrated in Fig. 3, the evolutions of both low- and high-fidelity models are significantly different, and thus we do not expect to be able to use MF simulations to compute detailed trajectories; however, our quantities of interest are averaged metrics, and so using a MF surrogate procedure can be effective. The results provided in Fig. 4 indicate success of the kernel selection procedures for constructing MF emulators. This level of accuracy is obtained despite the apparent failure of the standard linear kernel MF surrogate. Thus, the kernel selection procedure takes advantage of the optimization process to promote numerical stability and accuracy. Here, the adaptive kernel approach provides us with a slightly more accurate predictive model. This could be the result excluding several "improper" choices of kernel function. In the additive kernel approach for this problem, every kernel function in the library has a nonzero contribution in construction of the additive kernel.

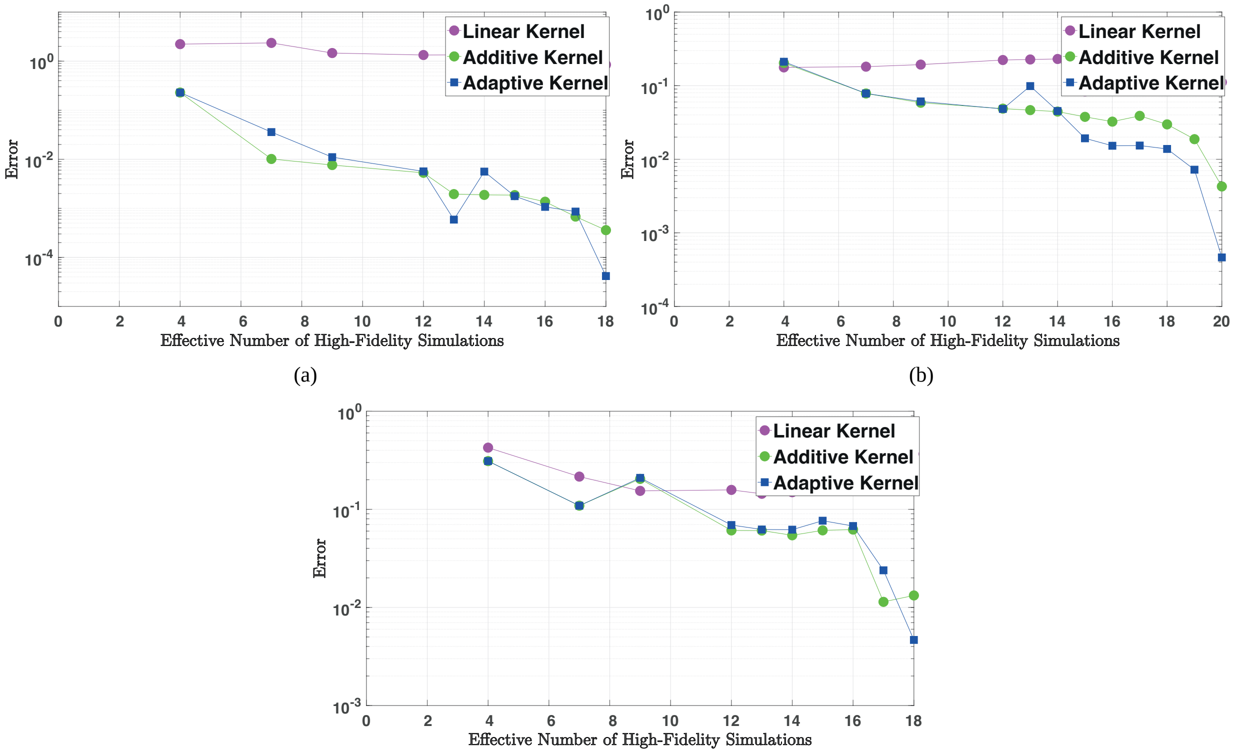
As shown in Fig. 4, the cost of multifidelity simulations scales proportionally to the cost of each added high-fidelity simulation sample. By increasing the number of high-fidelity samples from 4 to 18 for the construction of the multifidelity emulator, the prediction error of the emulator models reduces by around two orders of magnitude. This is the case for all outputs of the model, and for two of the proposed approaches. This level of accuracy is achieved despite the complexity of the dynamical system, but is possible by considering only averaged characteristics of the system. We expect significant decline of accuracy if particle trajectories or velocities are directly predicted.



(a)

(b)

FIG. 3: Galaxy systems with 5 and 500 bodies as the low- and high-fidelity systems; $M = 50$; $\Gamma = 0$. Galaxy with (a) 25 and (b) 500 objects after 14,000 megayears.



(a)

(b)

(c)

FIG. 4: Median error of the multifidelity model constructed based upon the results from the low-rank low-fidelity galaxy (n body) model with $n = 25$ bodies in the prediction of the properties for the high-fidelity model of the galaxy with $n = 500$ bodies after 14,000 megayears; Test Problem 2; 36 data points were tested for each case to measure the error. (a) Total energy, (b) mean distance of objects in the n-body system relative to the center of the system, and (c) mean velocity of n-body system.

4.3 Test Problem 3: Associating Polymer Network

Telechelic associated polymer networks in their equilibrium condition can be modeled as a simulated gel network (SGN) with its nodes characterized as "peers" and "superpeers"; for details, see [36]. The number of links and nodes in the network depends on the temperature. For this test problem, the fractions of superpeers (n_s) and peers (n_p) at nondimensional temperature (T) of 0.55 based on Table 1 of Ref. [36] are set to be 0.604 and 0.396, respectively. The following system of equations and constraints models the network that exhibits the same properties as those of an SGN:

$$\begin{aligned} n_s\langle k\rangle_s &= 2l_{ss} + l_{ps}, \\ n_p\langle k\rangle_s &= l_{ps} + 2l_{pp}, \\ l_{\text{tot}} &= l_{pp} + l_{ps} + l_{ss}, \\ l_{ss} &= (0.85 \pm 0.01)l_{\text{tot}}, \\ l_{pp} &= (0.02 \pm 0.01)l_{\text{tot}}, \\ l_{ps} &= (0.13 \pm 0.01)l_{\text{tot}}, \end{aligned} \tag{16}$$

where l_{ss}, l_{ps}, and l_{pp} represent the probability of having a connection between two superpeers, between one peer and a superpeer, and between two peers, respectively. Here $\langle k\rangle$ denotes the average degree (number of connections for each node), and for $T = 0.55$ is set to be $\langle k\rangle = 7.131$ and $\langle k\rangle = 0.997$ for superpeers and peers, respectively. We consider 1200 nodes for the model network and randomly generate links based on the calculated values of l_{ss}, l_{ps}, and l_{pp}.

In order to study the topological change in the simulated associating polymers under constant stress, we consider the following governing equations for the nodal strain evolution in the network [37]:

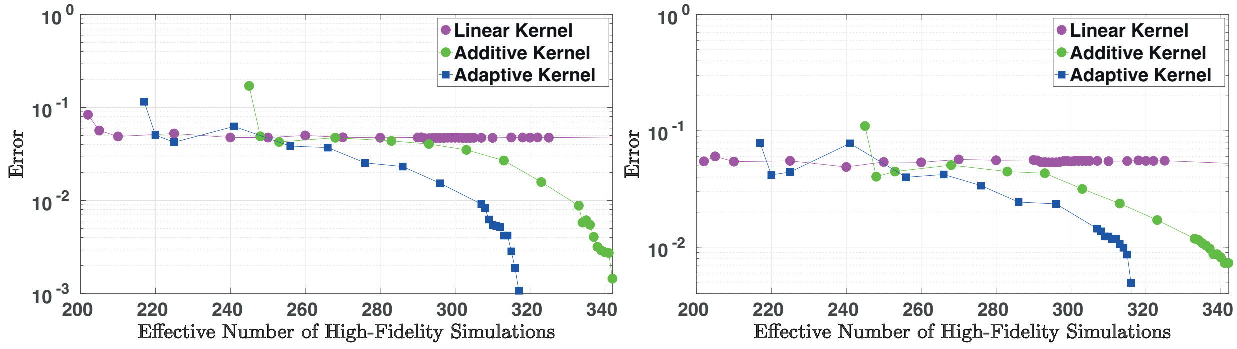
$$\frac{d\varepsilon}{dt} = \frac{1}{\eta} A\sigma_0, \tag{17}$$

where η, σ_0, ε, and A are a coefficient of viscosity, vectors of constant stress, time-dependent strain, and the network's adjacency matrix, respectively. For this test problem, we consider the effective resistances between any two nodes in a network and use a graph sparsification process [38] to obtain a low-fidelity network model. Hence, the high- and low-fidelity models in this case are fully dense and sparsified networks, respectively. Here, the parameter space is obtained by uniform random variation of the viscosity coefficient and constant in $[0.05, 2.5]$ and between 0 and 2.5, respectively. Two scalar quantities of interest for this test problem are the mean and standard deviation of the resulting strain field at $t = 10$. In this case the output quantities of interest are two-dimensional vectors. Once again, as shown in Fig. 5, the dimension of the outputs being small directly impacts the accuracy of the standard (linear kernel) multifidelity model. However, the MF models using the new modified kernel approaches perform much better. Similar to the result of the previous test problems, the adaptive approach appears to be more cost-effective and results in a more accurate predictive model. Once again, this better accuracy is the result of selection of the optimal kernel function exclusively in the adaptive approach instead of estimation of the contribution factor for each kernel function in the function library in the additive approach.

Here, as the numbers of nodes in both low- and high-fidelity networks are the same, the computational cost of the simulation for both are roughly the same. This is the main reason behind the large number of effective high-fidelity simulations required to obtain a reasonable level of accuracy for the MF emulators.

4.4 Test Problem 4: Plasmonic Nanoparticle Arrays

In a wide range of real-world technical problems in engineering, the quantities of interest are scalar quantities. As mentioned previously in this paper, using the linear kernel function in such a situation is essentially ineffective, producing an MF emulator with only one degree of freedom, and results in a surrogate model with a significant inaccuracy. Designing a plasmonic nanoparticle array with an optimal scattering and/or extinction efficiency is among these real-world scenarios. For the calculation of these efficiency parameters as well as simulation of the optical response of a set of identical nonmagnetic metallic nanospheres with sizes much smaller than the wavelength of light (here 25



(a)

(b)

FIG. 5: Median error of the multifidelity model constructed based upon the results from the low-rank low-fidelity model (sparse model polymer network) in the prediction of the mean and standard deviation of the resultant strain field at $t = 10$ for the high-fidelity model (dense model polymer network); Test Problem 3; 200 data points were tested for each case to measure the error. (a) Strain field mean and (b) strain field standard deviation.

nm), the coupled dipole approximation (CDA) method [39] is used. Using this method, for N metallic nanoparticles described by the same volumetric polarizability $\alpha(\omega)$ and located at vector positions r_i, one can compute the local field $E_{loc}(r_i)$ as

$$\left(1 - \frac{\alpha k^2}{\epsilon_0} \sum_{j=1, j\neq i}^{N} \tilde{\mathbf{G}}_{ij}\right) \mathbf{E}_{loc}(\mathbf{r}_i) = \mathbf{E}_0 \mathbf{r}_i, \tag{18}$$

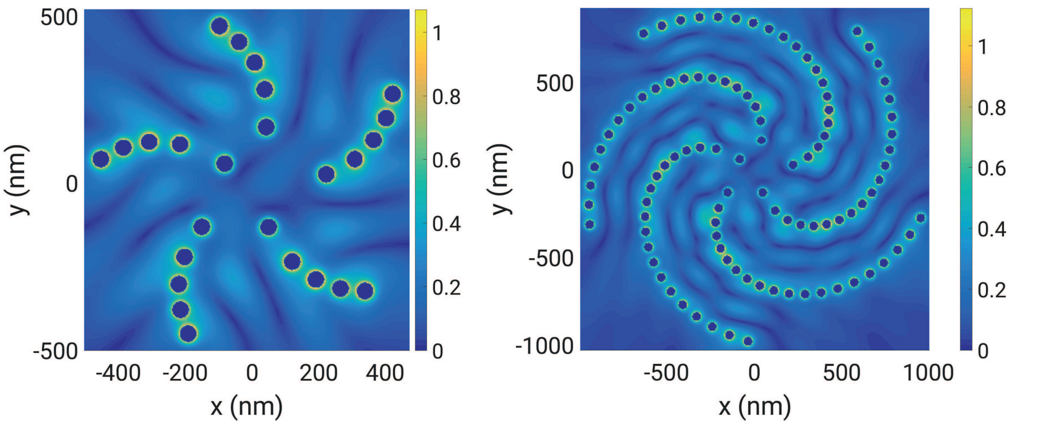
where $\mathbf{E}_0(r_i)$ is the incident field in vector form, k is the wave number in the background medium, ϵ_0 is the dielectric permittivity of vacuum ($\epsilon_0 = 1$ in the CGS unit system), and $\tilde{\mathbf{G}}_{ij}$ is constructed from 3×3 block of the overall a $3N \times 3N$ Green's matrix for the ith and jth particles and the summation runs through all jth particles except for $j = i$.

By solving (18), the total local fields $[\mathbf{E}_{loc}(r_i)]$ are computed, and consequently the local electric dipole polarizations across the array can be determined. The scattering and extinction cross sections can be computed using the resulting polarizations. Finally, the corresponding extinction (Q_{ext}) and scattering (Q_{sc}) efficiencies are obtained by normalization of scattering and extinction cross sections with respect to the total projected area of the array (i.e., sum of the areas of the particles projected perpendicularly to the direction of the excitation beam).

In particular we consider Vogel spirals arrays [40,41], which can uniquely identify useful nanoparticle array configurations with only four parameters: the number of particles (α_{vs}), divergence angle, incident wavelength, and scaling factor.

Here, the number of particles in the array defines the level of fidelity. Varying this fidelity parameter can result in significantly different outputs (compare the two panels in each of Figs. 6 and 7). Also the limited dimension of the output parameter space for this case makes the choice of the linear kernel function prohibitive. The results shown in Figs. 8 and 9 confirm this deduction for MF models of silver and aluminum nanoparticle arrays. The number N of particles in the low- and high-fidelity models for both cases are $N = 25$ and $N = 100$, respectively for Fig. 6, and are $N = 50$ and $N = 1000$ for Fig. 7, respectively. Both of the novel kernel selection approaches are successfully applied to develop surrogate models with much higher accuracy (see Figs. 8 and 9). The computational cost of the kernel selection optimization for this test problem is insignificant in comparison to the cost of running one high-fidelity simulation. Hence, the prediction error of the multifidelity emulator coverages more smoothly and monotonically. Here, the accuracy of both new kernel selection MF methods appears to be similar.

Due to the complexity of plasmonic models for large-scale arrays of particles, the prediction accuracy is lower in Fig. 6 compared to Fig. 7. Another reason for this lower accuracy is the more pronounced difference between parametric dependence of the low- and high-fidelity nanoparticle arrays and corresponding plasmonic models. Similar



(a) (b)

FIG. 6: Calculated local extinction field from CDA; Test Problem 4; silver; $\lambda = 513.228$ nm, $a_{vs} = 100.019$, and $\alpha_{vs} = 2.5358$ rad. (a) $N = 25$ and (b) $N = 100$.

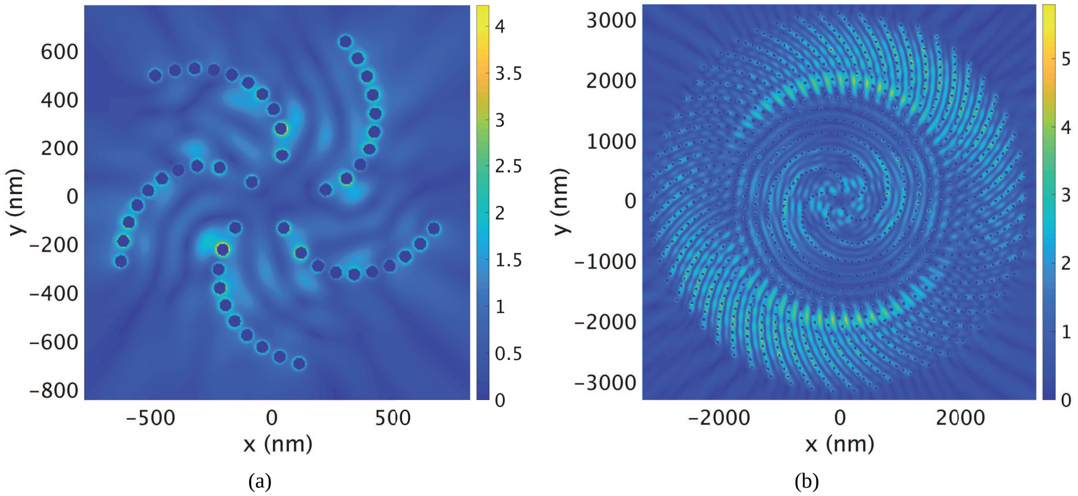


FIG. 7: Calculated local extinction field from CDA; Test Problem 4; aluminum; λ $=$ 213.228 nm, a_{vs} $=$ 100.019, and α_{vs} $=$ 2.5358 rad. (a) N $=$ 25 and (b) N $=$ 1000.

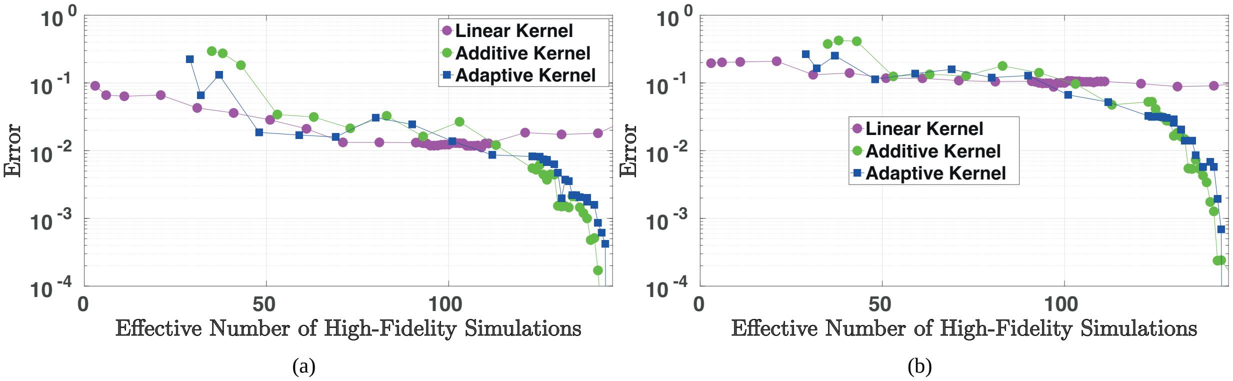
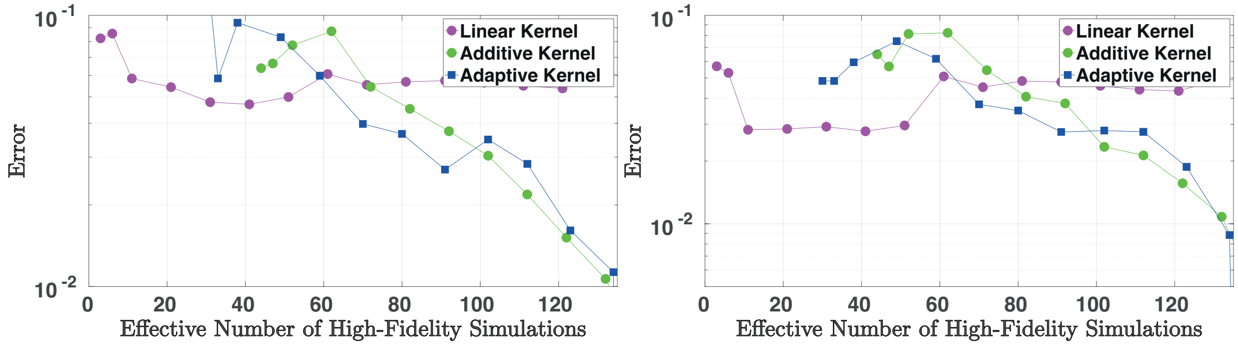


FIG. 8: Median error of the multifidelity model constructed based upon the results from the low-rank low-fidelity model with 25 nanoparticles (N_L $=$ 25) in the prediction of the extinction and scattering efficiencies for the high-fidelity models with 100 silver nanoparticles (N_H $=$ 100); Test Problem 4; 250 data points were tested for each case to measure the error. (a) Q_{ext} and (b) Q_{sc}.

parametric dependence between low- and high-fidelity models is beneficial for the selection of a better set of high-fidelity sampling points.

4.5 Test Problem 5: An Incompressible Flow in Two-Dimensional Channels with Stenosis

Our previous examples have investigated the two proposed kernel selection schemes when the model output space has small dimension. In most of the cases above, the adaptive kernel approach has a small advantage. However, it is also necessary to investigate the performance of these approaches for cases in which the dimension of output space is high enough to pick a sufficient number of “important” samples for the construction of the MF model even with the linear kernel function. The objective of considering this test problem is to provide some insights about the competitiveness of the adaptive kernel approach compared to the linear kernel function method, when the Gramian matrix for both is not ill-conditioned. Since the previous examples suggest that the adaptive kernel procedure is slightly more accurate than the additive method, we focus on highlighting results for the adaptive kernel only. Note that the additive kernel is slightly less computationally expensive than the adaptive kernel method.



(a)

(b)

FIG. 9: Median error of the multifidelity model constructed based upon the results from the low rank low-fidelity model with 50 nanoparticles ($N_L = 50$) in the prediction of the extinction and scattering efficiencies for the high-fidelity models with 1000 aluminum nanoparticles ($N_H = 1000$); Test Problem 4; 250 data points were tested for each case to measure the error. (a) Q_{ext} and (b) Q_{sc}.

In many fluid dynamics problems a large amount of data can be extracted from a solution, and thus are a good example of high-dimensional output space. Here, we consider a two-dimensional incompressible stenotic flow in a large channel with a length and width of 11 and 1 m, respectively. The shape of the curve of stenosis in the middle of the channel follows the equation (see Fig. 10)

$$y(x, y_0) = 1 - \frac{y_0}{2}(1 + \cos(2\pi(x - 5))). \tag{19}$$

The parameter space for this problem is two-dimensional, comprised of the flow Reynolds number (Re) and the parameter y_0: $(\text{Re}, y_0) \in [0.1, 0.6] \times [10, 500]$. The inlet flow stream has a velocity of 1 m/s and for simplicity the fluid kinematic viscosity ν is set to be $1/\text{Re}$. The goal of constructing predictive models for this problem is to estimate horizontal and vertical velocity (u and v, respectively) profiles in the middle of the channel ($x = 5.5$ m) as well as the shear stress on the top and bottom walls of the channel ($y = 0$ and 1, respectively) at $t = 30$ sec. The HF model in this case solves the full unsteady incompressible Navier-Stokes equation, which can be written in vector form as follows [42]:

$$\frac{\partial U}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} + H = 0, \tag{20}$$

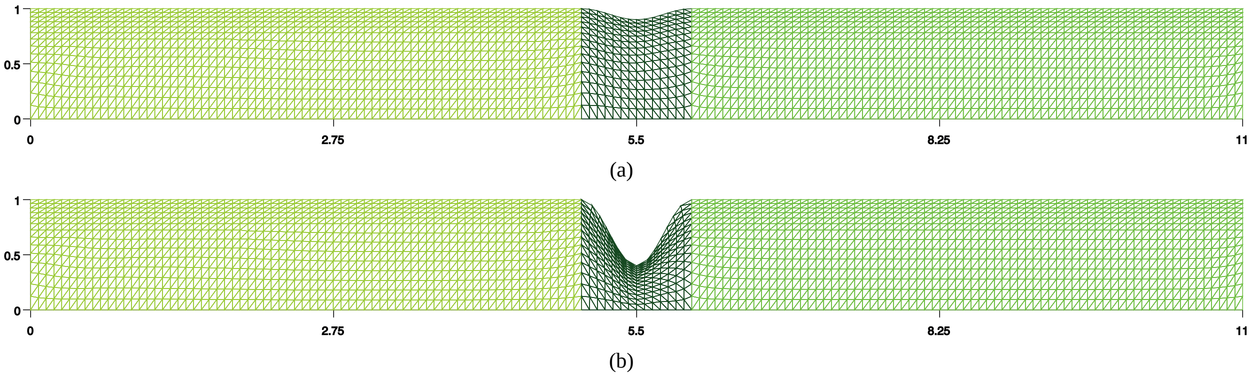
where

$$\begin{aligned} U &= [0, u, v]^T, \\ E &= \left[u, u^2 + P - \nu\frac{\partial u}{\partial x}, uv - \nu\frac{\partial v}{\partial x}\right]^T, \\ F &= \left[v, uv - \nu\frac{\partial u}{\partial y}, v^2 + P - \nu\frac{\partial v}{\partial y}\right]^T, \\ H &= [-f_x, -f_y, -f_z]^T. \end{aligned}$$

Here, P and f denote pressure and external force, respectively. The LF model is the coupled linearized Navier-Stokes equation, which is obtained by decomposing the solution into a basic state and perturbed state:

$$\frac{\partial V}{\partial t} + W \cdot \nabla V + V \cdot \nabla W = -\nabla P + \nu \nabla^2 V + f, \tag{21}$$

where W and $W + \varepsilon V$ are the basic and perturbed states, respectively, and $\varepsilon \ll 1$.



(a)

(b)

FIG. 10: Different geometrical configurations tested for two-dimensional stenotic flows; stenosis shape function dependence on the axial coordinate x: $f(x) = (1 - (f_0/2))(1 + \cos(2\pi(x - 5)))$; Test Problem 5. (a) $f_0 = 0.1$ and (b) $f_0 = 0.6$.

Both Eqs. (20) and (21) can be cast into the form of $L(U) = B$ for a linear operator L- and U-independent term B. Via a Galerkin discretization, this can subsequently be cast in matrix form:

$$A\hat{U} = \hat{B}, \tag{22}$$

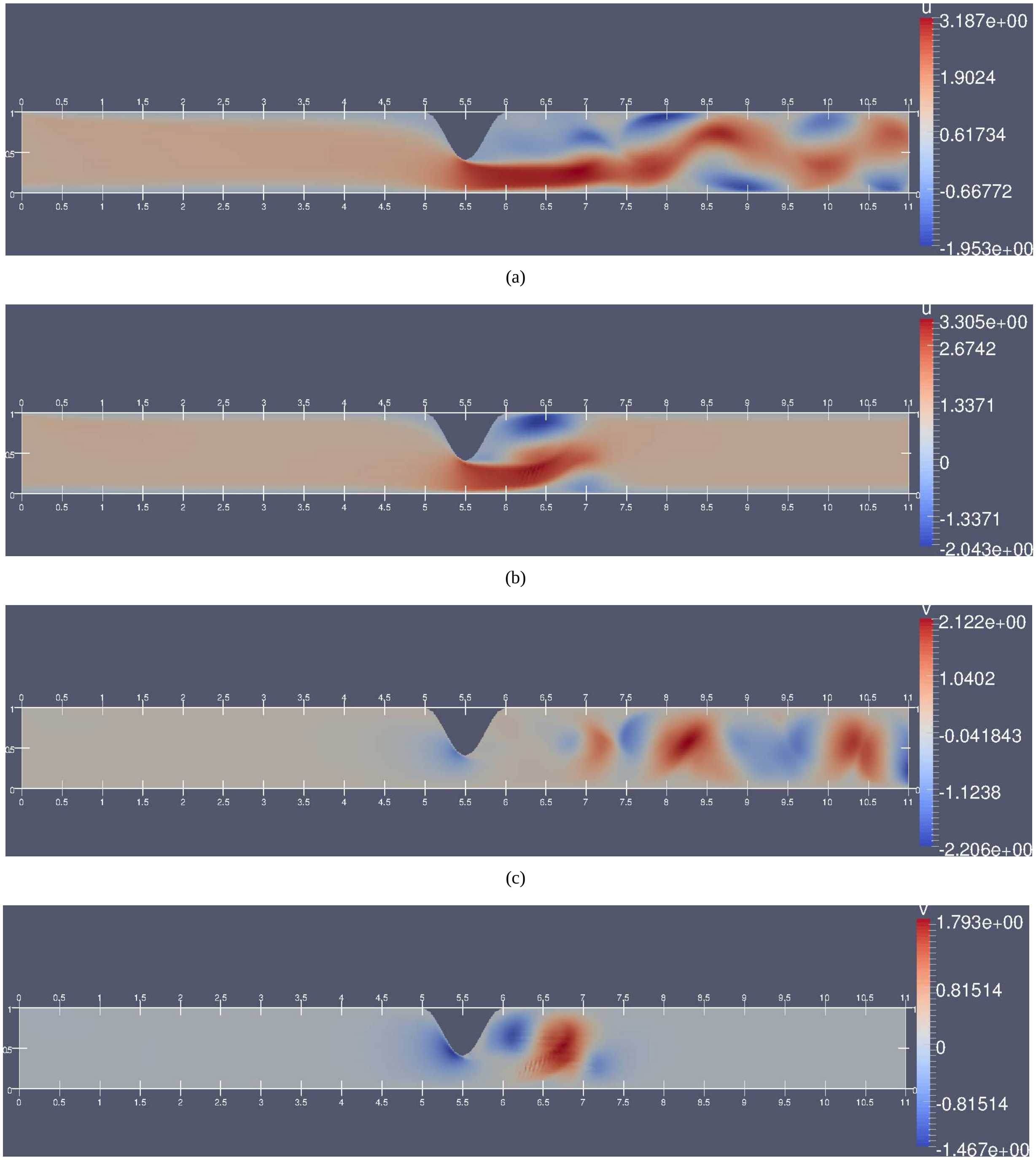
where

$$\begin{aligned} \hat{B}_m &= \int_\Omega \Phi_m B dx, \\ A_{m,n} &= \int_\Omega \Phi_m L(\Phi_n) B dx, \end{aligned} \tag{23}$$

where Φ_m are basis functions for the Galerkin method. We solve this problem using Nektar++, which is an open-source spectral/hp element software [43]. Fourth-order polynomials are used to approximate the finite element solution of the problem. Both low- and high-fidelity models use the same mesh but they are different with respect to their corresponding solution models, and as shown in Fig. 11 this difference leads to a significant difference in the prediction of the flow field in the channel at $t = 30$ sec. The accuracy of the quantities of interest obtained from low-rank MF models with respect to the effective number of high-fidelity model evaluations is illustrated in Fig. 12. In this figure, the adaptive kernel MF method produces accuracy comparable to the linear kernel MF method for all quantities of interest. Figure 13 provides a clear picture for the inaccuracy of the low-fidelity model and the resulting enhancement in the predictions of the MF models. These results indicate that there is no penalty for considering a more complex kernel selection procedure in this example. Since the more complex kernel selection procedure does perform better than the linear kernel method for other examples, this suggests that using the adaptive kernel approach rather than the linear kernel approach is a better choice in general.

5. CONCLUDING REMARKS

Multifidelity surrogate modeling is an active area of research in the field of uncertainty quantification, with a wide range of applications in different areas of science and engineering. In the context of low-rank multifidelity modeling, one commonly used technique is to exploit the “kernel trick” to express correlations in parameter space. Such a technique is the building block of low-rank MF learning algorithms and the crucial reason behind their success in providing accurate predictions of quantities of interest for complex systems. While linear kernel functions have been historically used and have demonstrated success, several drawbacks arise such as rank deficiency and ill-conditioning of Gramian matrices when the dimension of the output space is small. The result is often numerical instability and inaccuracy of the emulator.



(a)

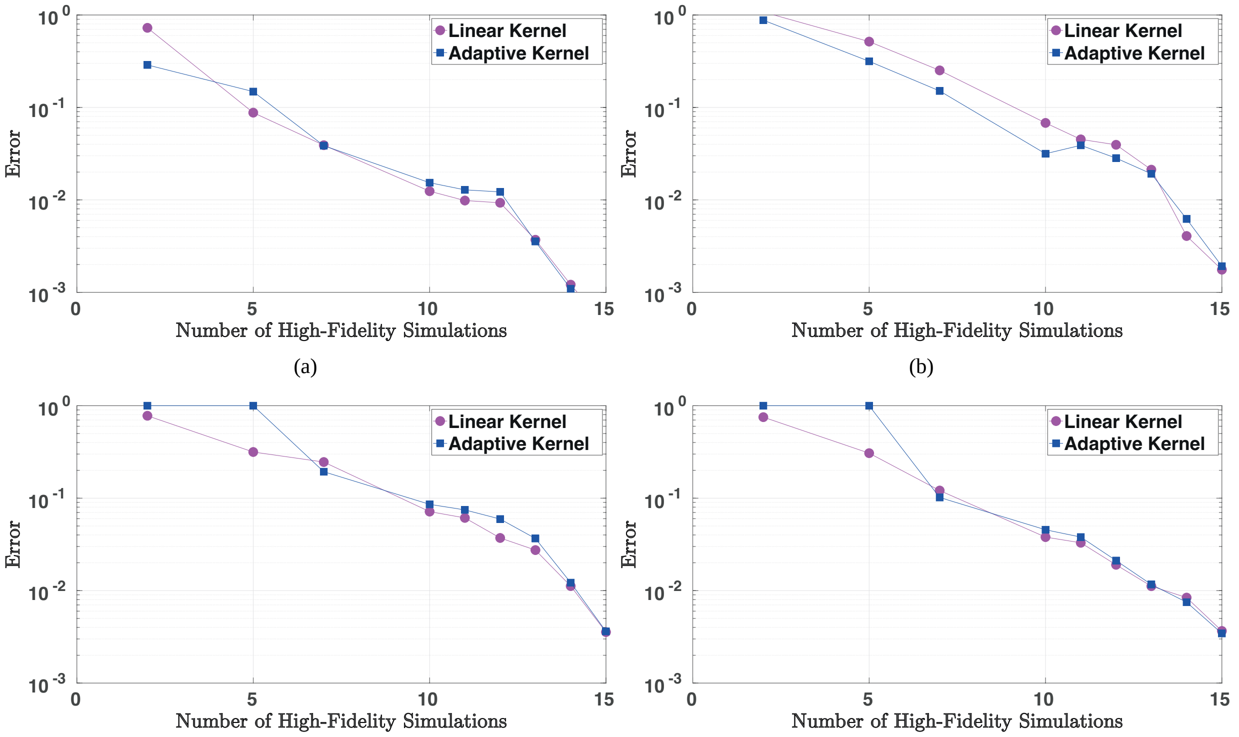
(b)

(c)

(d)

FIG. 11: Two-dimensional stenotic flows; $t = 30$ sec, Re $= 500$, and $f_0 = 0.6$; Test Problem 5. (a) u; full Navier-Stokes equation, (b) u; coupled linearized Navier-Stokes, (c) v; full Navier-Stokes equation, and (d) v; coupled linearized Navier-Stokes.

To address this issue, two novel approaches are proposed in this paper for data-driven kernel function selection from a library existing of popular kernel functions. The selection procedure is an optimization but only relies on



(a)

(b)

(c)

(d)

FIG. 12: Median error of the multifidelity model constructed based upon the results from the full-rank low-fidelity model with a coupled linearized Navier-Stokes equation solution in the prediction of the flow properties in a two-dimensional channel with stenosis for the corresponding high-fidelity models with full Navier-Stokes equation solution; Test Problem 5; $t = 30$ sec; 25 data points were tested for each case to measure the error. (a) Horizontal velocity (u) profile at $x = 6.5$ m, (b) vertical velocity (v) profile at $x = 6.5$ m, (c) top wall shear stress, and (d) bottom wall shear stress.

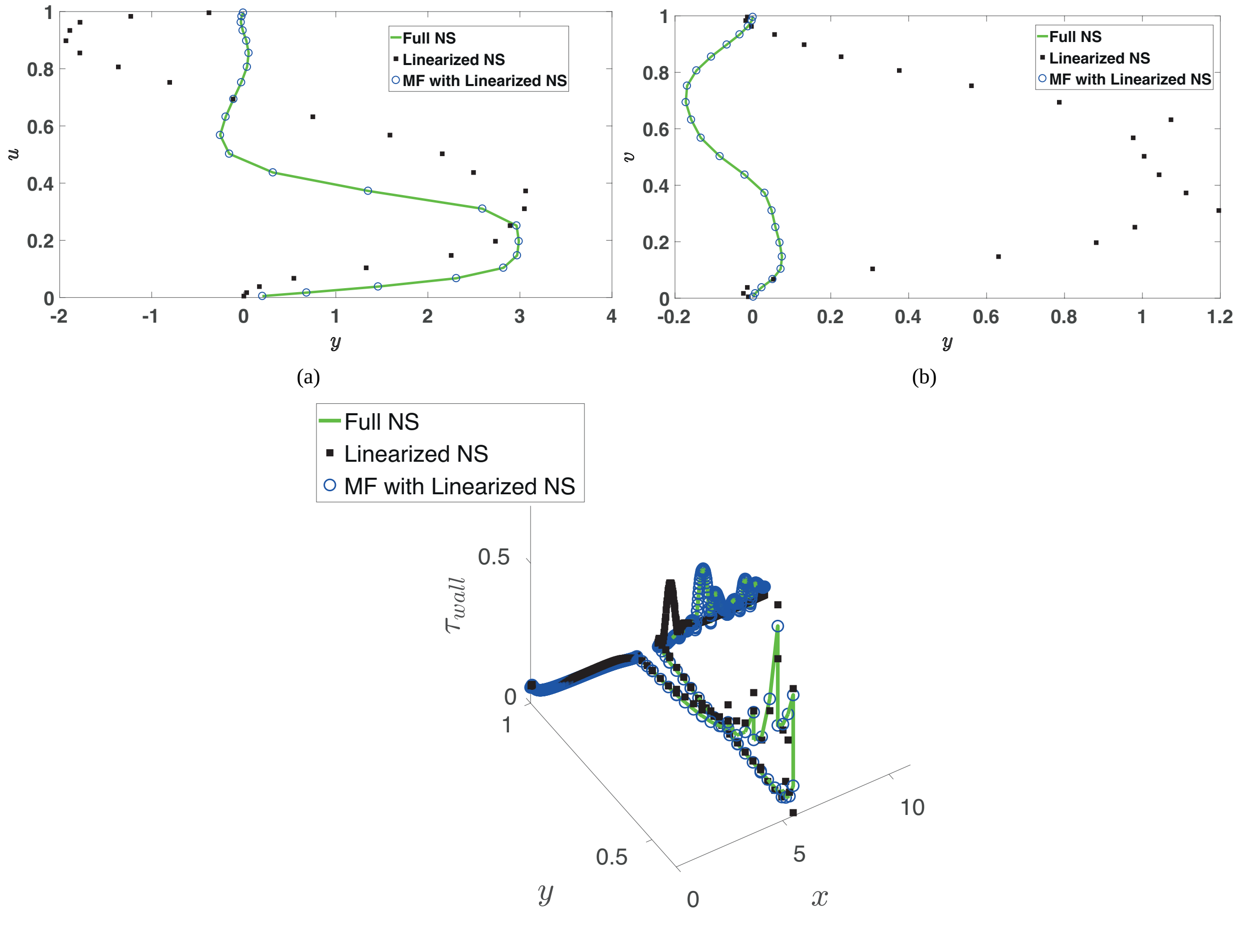
inexpensive low-fidelity data. Compared to the cost of a single HF evaluation, this optimization is negligible in practice. We have made some heuristic choices for hyperparameter optimization objective function (7) and in the choice of our optimizer (PSO with gradient descent), but alternative optimization techniques can be employed without changing the essential nature of our proposed methods. In addition, the kernel library (see Table 1) may be grown if more kernel families are desired, or shrunk if some are deemed unnecessary. We have proposed two techniques, an additive kernel approach, and an adaptive kernel approach, for selection of the final kernel.

To demonstrate the effectiveness of our methods, we have applied these new techniques to five nontrivial problems in disparate areas of science and engineering: molecular dynamics simulations, n-body galaxy models, associating polymer networks, plasmonic nanoparticle arrays, and incompressible flow in two-dimensional channels with stenosis. Our numerical experiments suggest the following:

1. The standard linear kernel is sometimes unreliable when models have a low-dimensional output space (cf., Fig. 5).

2. The new kernel selection methods have succeeded in producing accurate and relatively inexpensive surrogates when the linear kernel fails for low-dimensional output spaces, and can achieve errors that are smaller by one to two orders of magnitude for identical cost (cf., Fig. 8).

3. The kernel selection and optimization procedure optimizes only over the low-fidelity model, and hence is very inexpensive compared to a single high-fidelity model evaluation.



(a)

(b)

(c)

FIG. 13: Comparison of properties predicted for two-dimensional stenotic flows as obtained using high-fidelity model [full Navier-Stokes (NS) equation solution], low-fidelity model (coupled linearized Navier-Stokes equation solution), and multifidelity approach based on low-fidelity data and 16 samples from high-fidelity; Test Problem 5; $t = 30$ sec, Re $= 500$, and $f_0 = 0.6$. (a) u profile at the middle of the channel, (b) v profile at the middle of the channel, and (c) τ_{wall} for the top ($y = 1$).

4. When the model output space is high-dimensional, the linear kernel procedure works well, but the novel kernel selection procedures do not produce worse emulators (cf., Fig. 12).

5. The proposed methods can succeed on a diverse range of problems in science and engineering capability.

Our results also suggest a small preference for the adaptive kernel approach, due to a slightly better overall accuracy, along with its better computational scalability as the number L of kernel functions in the library grows. For very large L, the complexity of a high-dimensional (convex) optimization for the alternative additive kernel approach can severely impact the efficiency of this procedure. Hence, the authors generally recommend the implementation of the adaptive kernel approach if the user is free to choose.

There are still open questions and challenges with this approach. Low-rank MF methods are limited, in general, to cases in which the low-fidelity and high-fidelity models share a common parameter space and have similar parametric variations. Also, these procedures do not provide an approach for construction of a low-fidelity model given a trusted high-fidelity model. Finally, the procedures we have developed do not directly tackle the curse of dimensionality for high-dimensional parameter spaces. Future work will be devoted to tackling these challenges.

ACKNOWLEDGMENTS

M. Razi and R.M. Kirby acknowledge that their part of this research was sponsored by ARL under cooperative agreement number W911NF-12-2-0023. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of ARL or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. M. Razi and A. Narayan also acknowledge partial support from AFOSR no. FA9550-15-1-0467.

REFERENCES

1. Razi, M., Narayan, A., Kirby, R.M., and Bedrov, D., Fast Predictive Models based on Multi-Fidelity Sampling of Properties in Molecular Dynamics Simulations, *Comput. Mater. Sci.*, **152**(C):125–133, 2018.
2. Narayan, A., Gittelson, C., and Xiu, D., A Stochastic Collocation Algorithm with Multifidelity Models, *SIAM J. Sci. Comput.*, **36**(2):A495–A521, 2014.
3. Zhu, X., Narayan, A., and Xiu, D., Computational Aspects of Stochastic Collocation with Multifidelity Models, *SIAM/ASA J. Uncertainty Quantif.*, **2**(1):444–463, 2014.
4. Hampton, J., Fairbanks, H.R., Narayan, A., and Doostan, A., Practical Error Bounds for a Non-Intrusive Bi-Fidelity Approach to Parametric/Stochastic Model Reduction, *J. Comput. Phys.*, **368**:315–332, 2018.
5. Skinner, R., Doostan, A., Peters, E., Evans, J., and Jansen, K.E., An Evaluation of Bi-Fidelity Modeling Efficiency on a General Family of NACA Airfoils, in *Proc. of 35th AIAA Applied Aerodynamics Conf.*, p. 3260, 2017.
6. Jofre, L., Geraci, G., Fairbanks, H., Doostan, A., and Iaccarino, G., Multi-Fidelity Uncertainty Quantification of Irradiated Particle-Laden Turbulence, *Comput. Phys.*, arXiv:1801.06062, 2018.
7. Allaire, D. and Willcox, K., A Mathematical and Computational Framework for Multifidelity Design and Analysis with Computer Models, *Int. J. Uncertainty Quantif.*, **4**(1):1–20, 2014.
8. Lam, R., Allaire, D.L., and Willcox, K.E., Multifidelity Optimization Using Statistical Surrogate Modeling for Non-Hierarchical Information Sources, in *Proc. of 56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conf.*, p. 0143, 2015.
9. Razi, M., Narayan, A., and Kirby, R.M., Fast Predictive Multi-Fidelity Prediction with Models of Quantized Fidelity Levels, *J. Comput. Phys.*, **376**:992–1008, 2019.
10. Giles, M.B., Multilevel Monte Carlo Methods, *Acta Numer.*, **24**:259–328, 2015.
11. Fernndez-Godino, M.G., Park, C., Kim, N.H., and Haftka, R.T., Review of Multi-Fidelity Models, *Stat. Appl.*, arXiv:1609.07196, 2016.
12. Peherstorfer, B., Willcox, K., and Gunzburger, M., Survey of Multifidelity Methods in Uncertainty Propagation, Inference, and Optimization, *SIAM Rev.*, **60**(3):550–591, 2018.
13. Zhu, X., Linebarger, E.M., and Xiu, D., Multi-Fidelity Stochastic Collocation Method for Computation of Statistical Moments, *J. Comput. Phys.*, **341**:386–396, 2017.
14. Keshavarzzadeh, V., Kirby, R., and Narayan, A., Convergence Acceleration for Time-Dependent Parametric Multifidelity Models, *SIAM J. Numer. Anal.*, **57**(3):1344–1368, 2019.
15. Perdikaris, P., Venturi, D., Royset, J., and Karniadakis, G., Multi-Fidelity Modelling via Recursive Co-Kriging and Gaussian–Markov Random Fields, *Proc. R. Soc. A*, **471**(2179):20150018, 2015.
16. Perdikaris, P., Venturi, D., and Karniadakis, G.E., Multifidelity Information Fusion Algorithms for High-Dimensional Systems and Massive Data Sets, *SIAM J. Sci. Comput.*, **38**(4):B521–B538, 2016.
17. Mehmani, A., Chowdhury, S., Meinrenken, C., and Messac, A., Concurrent Surrogate Model Selection (COSMOS): Optimizing Model Type, Kernel Function, and Hyper-Parameters, *Struct. Multidisc. Optim.*, **57**(3):1093–1114, 2018.
18. Anderson, D. and Gu, M., An Efficient, Sparsity-Preserving, Online Algorithm for Low-Rank Approximation, in *Proc. of 34th Int. Conf. on Machine Learning*, pp. 156–165, 2017.
19. Perry, D.J. and Whitaker, R.T., Augmented Leverage Score Sampling with Bounds, in *Proc. of Joint European Conf. on Machine Learning and Knowledge Discovery in Databases*, New York: Springer, pp. 543–558, 2016.

20. Lozano, A., Swirszcz, G., and Abe, N., Group Orthogonal Matching Pursuit for Logistic Regression, in *Proc. of 14th International Conf. on Artificial Intelligence and Statistics*, pp. 452–460, 2011.
21. Perry, D., Kirby, R., Narayan, A., and Whitaker, R., Allocation Strategies for High Fidelity Models in the Multifidelity Regime, *SIAM/ASA J. Uncertainty Quantif.*, **7**(1):203–231, 2019.
22. Fasshauer, G.F., *Meshfree Approximation Methods with Matlab*, Singapore: World Scientific Publishing Company, 2007.
23. Rasmussen, C.E., Gaussian Processes in Machine Learning, in *Advanced Lectures on Machine Learning*, Berlin: Springer, pp. 63–71, 2004.
24. Bergstra, J. and Bengio, Y., Random Search for Hyper-Parameter Optimization, *J. Mach. Learn. Res.*, **13**(1):281–305, 2012.
25. Klein, A., Falkner, S., Bartels, S., Hennig, P., and Hutter, F., Fast Bayesian Optimization of Machine Learning Hyperparameters on Large Datasets, in *Proc. of 20th Int. Conf. on Artificial Intelligence and Statistics*, pp. 528–536, 2017.
26. Kennedy, J., Particle Swarm Optimization, in *Encyclopedia of Machine Learning*, Berlin: Springer, pp. 760–766, 2011.
27. Clerc, M., *Particle Swarm Optimization*, Vol. 93, Hoboken, NJ: John Wiley & Sons, 2010.
28. Deepa, S. and Sugumaran, G., Model Order Formulation of a Multivariable Discrete System Using a Modified Particle Swarm Optimization Approach, *Swarm Evol. Comput.*, **1**(4):204–212, 2011.
29. Shahzad, F., Baig, A.R., Masood, S., Kamran, M., and Naveed, N., Opposition-Based Particle Swarm Optimization with Velocity Clamping (OVCPSO), in *Advances Computational Intelligence*, Berlin: Springer, pp. 339–348, 2009.
30. Angeline, P.J., Using Selection to Improve Particle Swarm Optimization, in *Proc. of 1998 IEEE World Congress on Computational Intelligence*, IEEE, pp. 84–89, 1998.
31. Angeline, P.J., Evolutionary Optimization versus Particle Swarm Optimization: Philosophy and Performance Differences, *International Conference on Evolutionary Programming*, Springer, pp. 601–610, 1998.
32. Dittmann, I. and Maug, E.G., Biases and Error Measures: How to Compare Valuation Methods, *ERIM Report Series, Ref. No. ERS-2006-011-F&A*, p. 2006-07, 2008.
33. Lee, J.G., *Computational Materials Science: An Introduction*, Boca Raton, FL: CRC Press, 2016.
34. Trenti, M. and Hut, P., Gravitational N-Body Simulations, *Astrophys.*, arXiv:0806.3950, 2008.
35. Jacobs, P., List, P., Ludin, M., Weeden, A., and Panoff, R.M., The Blue Waters Student Internship Program: Promoting Competence and Confidence for Next Generation Researchers in High-Performance Computing, in *Proc. of Workshop on Education for High-Performance Computing*, IEEE Press, pp. 49–55, 2014.
36. Billen, J., Wilson, M., Rabinovitch, A., and Baljon, A.R., Topological Changes at the Gel Transition of a Reversible Polymeric Network, *Europhys. Lett.*, **87**(6):68003, 2009.
37. Christensen, R., *Theory of Viscoelasticity: An Introduction*, Cambridge: Academic Press, 1982.
38. Spielman, D.A. and Srivastava, N., Graph Sparsification by Effective Resistances, *SIAM J. Comput.*, **40**(6):1913–1926, 2011.
39. Guérin, C.A., Mallet, P., and Sentenac, A., Effective-Medium Theory for Finite-Size Aggregates, *J. Opt. Soc. Am. A*, **23**(2):349–358, 2006.
40. Christofi, A.C., Pinheiro, F.A., and Dal Negro, L., Probing Scattering Resonances of Vogel's Spirals with the Green's Matrix Spectral Method, *Opt. Lett.*, **41**(9):1933–1936, 2016.
41. Razi, M., Wang, R., He, Y., Kirby, R.M., and Dal Negro, L., Optimization of Large-Scale Vogel Spiral Arrays of Plasmonic Nanoparticles, *Plasmonics*, **14**(1):253–261, 2019.
42. Anderson, D., Tannehill, J.C., and Pletcher, R.H., *Computational Fluid Mechanics and Heat Transfer*, Boca Raton, FL: CRC Press, 2016.
43. Cantwell, C.D., Moxey, D., Comerford, A., Bolis, A., Rocco, G., Mengaldo, G., De Grazia, D., Yakovlev, S., Lombard, J.E., and Ekelschot, D., Nektar++: An Open-Source Spectral/hp Element Framework, *Comput. Phys. Commun.*, **192**:205–219, 2015.