Survey of multifidelity methods in uncertainty propagation, inference, and optimization

ACDL Technical Report TR16-1

Benjamin Peherstorfer* Karen Willcox* Max Gunzburger†

June 30, 2016

In many situations across computational science and engineering, multiple computational models are available that describe a system of interest. These different models have varying evaluation costs and varying fidelities. Typically, a computationally expensive high-fidelity model describes the system with the accuracy required by the current application at hand, while lower-fidelity models are less accurate but computationally cheaper than the high-fidelity model. Outer-loop applications, such as optimization, inference, and uncertainty quantification, require multiple model evaluations at many different inputs, which often leads to computational demands that exceed available resources if only the high-fidelity model is used. This work surveys multifidelity methods that accelerate the solution of outer-loop applications by combining high-fidelity and low-fidelity model evaluations. The overall premise of these multifidelity methods is that low-fidelity models are leveraged for speedup while the high-fidelity model is kept in the loop to establish accuracy and/or convergence guarantees. We categorize multifidelity methods according to three classes of strategies: adaptation, fusion, and filtering. The paper reviews multifidelity methods in the outer-loop contexts of uncertainty propagation, inference, and optimization.

Keywords: multifidelity; surrogate models; model reduction; multifidelity uncertainty quantification; multifidelity uncertainty propagation; multifidelity statistical inference; multifidelity optimization

*Department of Aeronautics & Astronautics, MIT, Cambridge, MA 02139

†Department of Scientific Computing, Florida State University, 400 Dirac Science Library, Tallahassee FL 32306-4120

1 Introduction

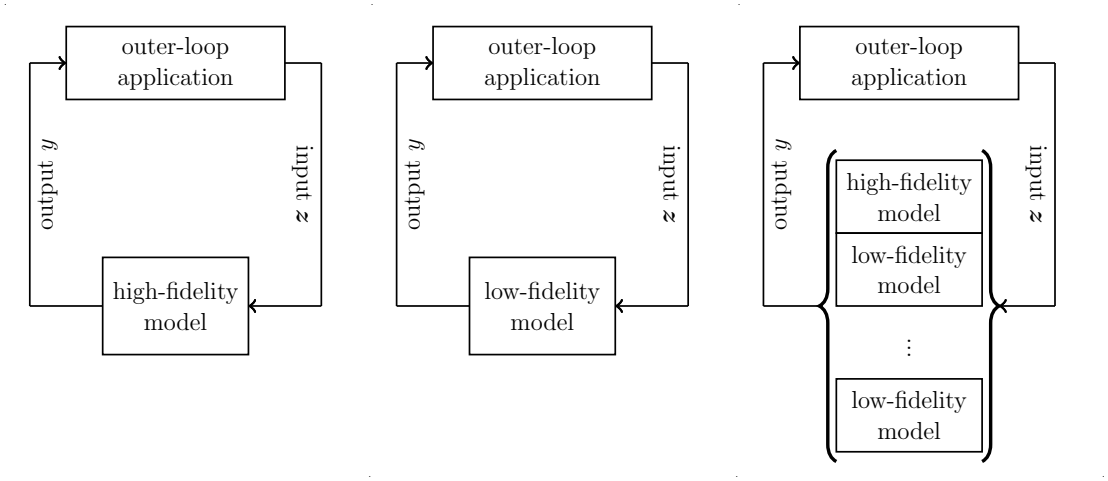
Section 1.1 introduces the concepts of multifidelity methods. Section 1.2 defines the three outer-loop applications of interest: uncertainty propagation, statistical inference, and optimization. Section 1.3 outlines the remainder of the paper.

1.1 Multifidelity methods

Models serve to support many aspects of computational science and engineering, from discovery to design to decision-making and more. In some of these settings, one primary purpose of a model is to characterize the input-output relationship of the system of interest—the input describes the relevant system properties and environmental conditions, and the output describes quantities of interest to the task at hand. In this context, evaluating a model means performing a numerical simulation that implements the model, computes a solution, and thus maps an input onto an approximation of the output. Model evaluations incur computational costs that typically increase with the accuracy of the approximation of the output. In many situations, several models are available that estimate the same output quantity with varying approximation qualities and varying computational costs. We define a *high-fidelity model* as a model that estimates the output with the accuracy that is necessary for the current task at hand, and we define a *low-fidelity model* as a model that estimates the output with a lower accuracy than the high-fidelity model (typically in favor of lower costs than the costs of the high-fidelity model).

We use the term *outer-loop application* to define computational applications that form outer loops around a model—where in each iteration an input is received and the corresponding model output is computed, and an overall outer-loop result is obtained at the termination of the outer loop. For example, in optimization, the optimizer provides at each iteration the design variables to evaluate (the input) and the model must evaluate the corresponding objective function value, the constraints values, and possibly the gradient information (the outputs). At the termination, an optimal design is obtained (the outer-loop result). Another outer-loop application is uncertainty propagation, which can be thought of conceptually as a loop over realizations of the input, requiring the corresponding model evaluation for each realization. In uncertainty propagation, the outer-loop result is the estimate of the statistics of interest. Other examples of outer-loop applications include inverse problems, data assimilation, control problems, and sensitivity analysis. See, e.g., [129, Chapter 10.1] for a discussion of outer-loop applications in uncertainty quantification and optimization, and [117] for specific examples of outer-loop applications in the context of petroleum production. Note that although it is helpful for the exposition to think of outer-loop applications as loops, they are often *not* implemented as such. For example, in uncertainty propagation, once the realizations of the input have been drawn, the model outputs can be typically computed in parallel.

The term *many-query application* is often used to denote applications that evaluate a model many times [181], a categorization that applies to most (if not all) outer-loop applications. We distinguish between many-query and outer-loop applications by consid-



(a) single-fidelity approach with high-fidelity model

(b) single-fidelity approach with low-fidelity model

(c) multifidelity approach with high-fidelity model and multiple low-fidelity models

Figure 1: Multifidelity methods combine the high-fidelity model with low-fidelity models. The low-fidelity models are leveraged for speedup and the high-fidelity model is kept in the loop to establish accuracy and/or convergence guarantees on the outer-loop result.

ering the latter to be the class of applications that target a specific outer-loop result. In contrast, many-query applications do not necessarily target a specific outer-loop result (and thus the set of outer-loop applications is essentially a subset of the set of many-query applications). For example, performing a parameter study is many-query but does not necessarily lead to a specific outer-loop result. This distinction is important in the discussion of multifidelity methods, since accuracy and/or convergence will be assessed relative to a specific outer-loop result.

The accuracy of the outer-loop result, as required by the problem at hand, can be achieved by using the high-fidelity model in each iteration of the outer loop; however, evaluating the high-fidelity model in each iteration often leads to computational demands that exceed the available resources. Simply replacing the high-fidelity model with a low-fidelity model can result in significant speedups but leads to a lower—and typically unknown—approximation quality of the outer-loop result. This is clearly unsatisfactory and motivates the need for multifidelity methods.

We survey here multifidelity methods for outer-loop applications. We consider the class of multifidelity methods that have two key properties: (1) They leverage low-fidelity models to obtain computational speedups, and (2) they use recourse to the high-fidelity model to establish accuracy and/or convergence guarantees on the outer-loop result, see Figure 1. Thus, multifidelity methods use low-fidelity models to reduce the runtime where possible, but recourse to the high-fidelity model to preserve the accuracy of the outer-loop result that would be obtained with a method that uses only the high-fidelity model. The two key ingredients of multifidelity methods are (1) low-fidelity models that

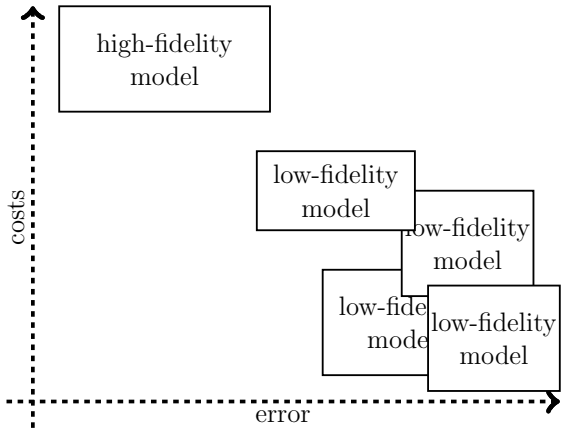


Figure 2: In many situations, different types of low-fidelity models are available, e.g., coarse-grid approximations, projection-based reduced models, data-fit interpolation and regression models, machine-learning-based models, and simplified models. The low-fidelity models vary with respect to error and costs. Multifidelity methods leverage these heterogeneous types of low-fidelity models for speedup.

provide useful approximations of the high-fidelity input-output relationship, and (2) a model management strategy that distributes work among the models while providing theoretical guarantees that establish the accuracy and/or convergence of the outer-loop result.

The multifidelity methods we survey are applicable to a broad range of problems, but of particular interest is the setting of a high-fidelity model that corresponds to a fine-grid discretization of a partial differential equation (PDE) that governs the system of interest. In this setting, coarse-grid approximations have long been used as cheaper approximations. Varying the discretization parameters generates a hierarchy of low-fidelity models. We are here interested in richer and more heterogeneous sets of models, including projection-based reduced models [191, 181, 90, 21], data-fit interpolation and regression models [82, 80], machine-learning-based models such as support vector machines [201, 57, 39], and other simplified models [131, 149], see Figure 2. In a broader sense, we can think of the models as information sources that describe the input-output relationships of the system of interest. In that broader sense, expert opinions, experimental data, and historical data are potential information sources. We survey techniques for constructing low-fidelity models and discuss information sources beyond models in Section 3.

Model management serves two purposes. First is to balance model evaluations among the models (i.e., to decide which model to evaluate when). Second is to guarantee the same accuracy in the outer-loop result as if only the high-fidelity model were used. We distinguish between three types of model management strategies (see Figure 3): (1) *adapting* the low-fidelity model with information from the high-fidelity model, (2) *fusing* low- and high-fidelity model outputs, and (3) *filtering* to use the high-fidelity model only when indicated by a low-fidelity filter. The appropriate model management strategy

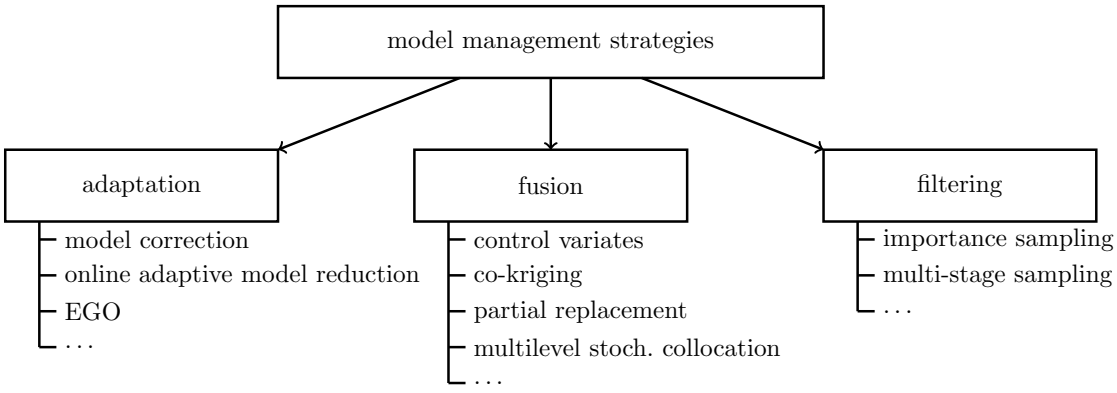


Figure 3: We distinguish between three model management strategies: adaptation, fusion, and filtering.

for the task at hand typically depends on the nature of the outer-loop application. We survey model management techniques that fall into these three categories in Section 4.

Comparison to multilevel methods Multilevel methods have a long history in computational science and engineering, e.g., multigrid methods [30, 96, 32, 198], multilevel preconditioners [29, 65], and multilevel function representations [212, 16, 66, 34]. Multilevel methods typically derive a hierarchy of low-fidelity models of the high-fidelity model by varying a parameter. For example, the parameter could be the mesh width and thus the hierarchy of low-fidelity models would be the hierarchy of coarse-grid approximations. A common approach in multilevel methods is to describe the approximation quality and the costs of the low-fidelity model hierarchy with rates, and then to use these rates to distribute work among the models. In this paper, we consider more general low-fidelity models with properties that cannot necessarily be well described by rates. Even though many multilevel methods are applicable to more heterogeneous models than coarse-grid approximations, describing the model properties by rates only, and consequently distributing work with respect to rates, can be too coarse a description and can miss important aspects of the models. Furthermore, in our setting, low-fidelity models are often given and cannot be easily generated on request by varying a (e.g., discretization) parameter. The multifidelity techniques that we describe here explicitly take such richer sets of models into account.

Comparison to traditional model reduction Traditionally, model reduction [10, 181, 21] first constructs a low-fidelity reduced model, and then replaces the high-fidelity model with the reduced model in an outer-loop application. Replacing the high-fidelity model often leads to significant speedups, but it also means that the accuracy of the outer-loop result depends on the accuracy of the reduced model. In some settings, error bounds or error estimates are available for the reduced model outputs [181, 203, 89], and it may be possible to translate these error estimates on the model outputs into error estimates on the outer loop result. In contrast, multifidelity methods establish accu-

racy and convergence guarantees—instead of providing error bounds and error estimates only—by keeping the high-fidelity model in the loop and thus trading some speedup for guarantees—even if the quality of the low-fidelity model is unknown.

1.2 Outer-loop applications

We focus on three outer-loop applications for which a range of multifidelity methods exist: uncertainty propagation, statistical inference, and optimization.

Uncertainty propagation In uncertainty propagation, the model input is described by a random variable and one is interested in statistics of the model output. Using Monte Carlo simulation to estimate statistics of the model output often requires a large number of model evaluations to achieve accurate approximations of the statistics. A multifidelity method that combines outputs from computationally cheap low-fidelity models with outputs from the high-fidelity model can lead to significant reductions in runtime and provide unbiased estimators of the statistics of the high-fidelity model outputs [86, 126, 148, 145, 194, 159]. Note that we consider probabilistic approaches to uncertainty propagation only; other approaches to uncertainty propagation are, e.g., fuzzy set approaches [24] and worst-case scenario analysis [13].

Statistical inference In inverse problems, an indirect observation of a quantity of interest is given. A classical example is that limited and noisy observations of a system output are given and one wishes to estimate the input of the system [138]. In statistical inference, the unknown input is modeled as a random variable and one is interested in sampling the distribution of this random variable to assess the uncertainty associated with the input estimation. Markov chain Monte Carlo (MCMC) methods provide one way to sample the distribution of the input random variable. MCMC is an outer-loop application that requires evaluating the high-fidelity model many times. Multifidelity methods in MCMC typically use multi-stage adaptive delayed acceptance formulations that leverage low-fidelity models to speed up the sampling [51, 72, 60, 63].

Optimization The goal of optimization is to find an input that leads to an optimal model output with respect to a given objective function. Optimization is typically solved using an iterative process that requires evaluations of the model in each iteration. Multifidelity optimization reduces the runtime of the optimization process by using low-fidelity models to accelerate the search [26, 114, 81, 80] or by using a low-fidelity model in conjunction with adaptive corrections and a trust-region model-management scheme [2, 4, 26, 134, 169]. Other multifidelity optimization methods build a surrogate using evaluations from multiple models, and then optimize using this surrogate. For example, efficient global optimization (EGO) is a multifidelity optimization method that adaptively constructs a low-fidelity model by interpolating the objective function corresponding to the high-fidelity model with Gaussian process regression (kriging) [109].

1.3 Outline of the paper

The remainder of this paper focuses on the two key ingredients of multifidelity methods: the construction of low-fidelity models, and model management strategies. Section 2 first introduces notation and terminology that is used throughout the presentation. Section 3 discusses low-fidelity models, including simplified models, projection-based reduced models, and data-fit models. Section 4 overviews the model management strategies of adaptation, fusion and filtering. Sections 5–7 survey specific techniques in the context of uncertainty propagation, inference, and optimization, respectively. The outlook in Section 8 closes the survey.

2 Notation and terminology

This section introduces notation and terminology that is used throughout the rest of the manuscript.

2.1 Notation

Let $\mathcal{D} \subseteq \mathbb{R}^d$ be the input domain, with dimension $d \in \mathbb{N}$, and let $\mathcal{Y} \subseteq \mathbb{R}^{d'}$ be the output domain, with dimension $d' \in \mathbb{N}$. Let further $\boldsymbol{z} \in \mathcal{D}$ be the input and $\boldsymbol{y} \in \mathcal{Y}$ the output, respectively. A model is a function $f : \mathcal{D} \to \mathcal{Y}$ that maps an input to an output. The costs of evaluating the model f are denoted by $c \in \mathbb{R}_+$, where $\mathbb{R}_+$ is the set of positive real numbers $\mathbb{R}_+ = \{x \in \mathbb{R} : x > 0\}$. The high-fidelity model f_{hi} provides an approximation of the output of interest with the accuracy required by the problem at hand and has costs $c_{\text{hi}} \in \mathbb{R}_+$. The output of a low-fidelity model f_{lo} is a poorer approximation of the output of interest than the high-fidelity model output, but the costs c_{lo} of the low-fidelity model are lower than the costs c_{hi} of the high-fidelity model, $c_{\text{lo}} < c_{\text{hi}}$. If there are $k \in \mathbb{N}$ low-fidelity models, we denote them as $f_{\text{lo}}^{(1)}, \ldots, f_{\text{lo}}^{(k)}$. A single-fidelity method uses only one model and a multifidelity method uses multiple models, see Figure 1.

2.2 Models based on systems of equations

Evaluating a model f often requires solving a system of equations that stems from the discretization of the governing equations of the system of interest. We denote these discretized systems in a general form as

$$\boldsymbol{A}(\boldsymbol{z})\boldsymbol{u}(\boldsymbol{z}) + \boldsymbol{g}(\boldsymbol{u}(\boldsymbol{z}); \boldsymbol{z}) = \boldsymbol{0}\,, \tag{1}$$

with $N \in \mathbb{N}$ degrees of freedom. The operator $\boldsymbol{A}(\boldsymbol{z}) \in \mathbb{R}^{N \times N}$ is a linear operator depending on the input $\boldsymbol{z}$, and $\boldsymbol{g} : \mathbb{R}^N \times \mathcal{D} \to \mathbb{R}^N$ is a nonlinear function. The state $\boldsymbol{u}(\boldsymbol{z}) \in \mathbb{R}^N$ is an N-dimensional vector. As an example, finite element discretizations of parametrized nonlinear PDEs lead to systems such as (1). The model output at input $\boldsymbol{z} \in \mathcal{D}$ is derived from the state $\boldsymbol{u}(\boldsymbol{z})$ with the output function $\ell : \mathbb{R}^N \to \mathcal{Y}$, i.e., $f(\boldsymbol{z}) = \ell(\boldsymbol{u}(\boldsymbol{z}))$. Thus, a model evaluation $f(\boldsymbol{z})$ entails solving the system (1).

Some of the methodologies surveyed in this paper exploit the structure of models of the form (1); however, the paper covers a much broader range of models. Many of the types of low-fidelity models, in particular, do not require solving a system of equations of the form (1) to evaluate the model. Furthermore, many of the multifidelity methodologies discussed in the following are applicable to more general nonlinear problems than (1).

2.3 Black-box models

Black-box models can be evaluated at the inputs in $\mathcal{D}$ to obtain outputs in $\mathcal{Y}$; however, no details on how the outputs are computed are available. For example, in the case of black-box models that are based on systems such as (1), the operator $\boldsymbol{A}(\boldsymbol{z})$ and the nonlinear function $\boldsymbol{g}$ are unavailable. Black-box models arise, for example, if the model implementation is closed-source software or if the implementation is too complex to understand the details within a reasonable amount of time.

3 Methods for constructing the low-fidelity models

We distinguish between three types of low-fidelity models, see Figure 4. The first type are simplified models that arise in a problem-dependent way from the high-fidelity model (e.g., by simplifying physics assumptions or linearization) or from its implementation (e.g., loosened residual tolerances). Section 3.1 discusses simplified models.

The second type of low-fidelity models are computed in a systematic way by projecting the governing equations of the high-fidelity model onto a problem-dependent, low-dimensional reduced space. The construction of such projection-based reduced models is typically intrusive, which means that knowledge of the governing equations of the high-fidelity model are required, e.g., the operators and nonlinear terms in case of a model (1) based on a PDE. Section 3.2 presents model reduction techniques for constructing projection-based reduced models.

Section 3.3 introduces data-fit low-fidelity models that are derived from data of the high-fidelity model. The construction of data-fit low-fidelity models is typically non-intrusive. Therefore, data-fit low-fidelity models can be derived from black-box high-fidelity models, where only inputs and the corresponding outputs (data) of the high-fidelity model are available and internals of the high-fidelity model are unknown. Interpolation and regression techniques from machine learning are commonly used to construct data-fit low-fidelity models.

Section 3.4 gives an outlook on more general information sources that go beyond computational models.

3.1 Simplified models

Simplified models are derived from the high-fidelity model by taking advantage of domain expertise and in-depth knowledge of the implementation details of the high-fidelity model. Domain expertise allows the derivation of several models with different computational costs and fidelities that all aim to estimate the same output of interest of

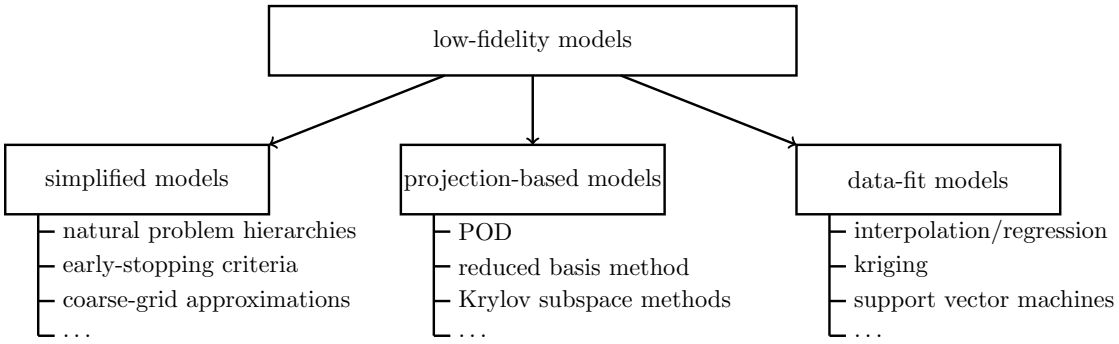


Figure 4: We distinguish between three types of low-fidelity models: simplified models, projection-based models, and data-fit models.

the system. For example, in computational fluid dynamics, there is a clear hierarchy of models for analyzing turbulent flow. From high to low fidelity, these are direct numerical simulations (DNS), large eddy simulations (LES), and Reynolds averaged Navier-Stokes (RANS). All these model turbulent flows, but DNS resolves the whole spatial and time domain to the scale of the turbulence, LES eliminates small scale behavior, and RANS applies the Reynolds decomposition to average over time. In aerodynamics, there is the hierarchy of RANS, Euler equations, and potential theory [100]. Similar hierarchies of models exist in other fields of engineering. Models for subsurface flows through karst aquifers reach from simple continuum pipe flow models to coupled Stokes and Darcy systems. In climate modeling, low-fidelity models consider only a limited number of atmospheric effects whereas high-fidelity models are fully-coupled atmospheric and oceanic simulation models [102, 131].

There are more general concepts to derive low-fidelity models by simplification, which also require domain expertise but which are applicable across disciplines. Coarse-grid discretizations are an important class of such approximations. As another example, in many settings a low-fidelity model can be derived by neglecting nonlinear terms (e.g., neglecting $\boldsymbol{g}$ in system (1)). For example, lower-fidelity linearized models are common in aerodynamic and structural analyses [166]. Neglecting nonlinear terms often results in significant reductions of the computational costs, which means that the resulting low-fidelity model is cheaper to evaluate than the high-fidelity model. High-fidelity models can be simplified if details of the implementation are known. If the high-fidelity model relies on an iterative solver (e.g., Krylov subspace solvers or Newton's method), a low-fidelity model can be derived by loosening the residual tolerances of the iterative method. Thus, to derive a low-fidelity approximation, the iterative solver is stopped earlier than if a high-fidelity output were computed. Another source of approximate information is from previously-computed cases. For example, intermediate results of an iterative optimization procedure are used as low-fidelity model outputs in a multifidelity optimization under uncertainty approach [149].

3.2 Projection-based low-fidelity models

Model reduction derives low-fidelity models from a high-fidelity model in a systematic way, by mathematically exploiting the problem structure rather than using domain knowledge of the problem at hand. In this section, we briefly overview projection-based model reduction for high-fidelity models that are based on systems of equations of the form (1) where the linear operator $\boldsymbol{A}(\boldsymbol{z})$ and the nonlinear function $\boldsymbol{g}$ are available.

Model reduction methods construct a reduced space from data of the high-fidelity model, and then derive an approximation of the state vector $\boldsymbol{u}(\boldsymbol{z}) \in \mathbb{R}^N$ of the high-fidelity model f_{hi} in the reduced space. Therefore, model reduction methods aim to construct a reduced system

$$\tilde{\boldsymbol{A}}(\boldsymbol{z})\tilde{\boldsymbol{u}}(\boldsymbol{z}) + \tilde{\boldsymbol{g}}(\tilde{\boldsymbol{u}}(\boldsymbol{z}), \boldsymbol{z}) = 0 \tag{2}$$

that has only $n \ll N$ degrees of freedom and where the low-fidelity state vector $\tilde{\boldsymbol{u}}(\boldsymbol{z}) \in \mathbb{R}^n$ of dimension $n \in \mathbb{N}$ leads to an acceptable approximation of the high-fidelity model output $f_{\text{hi}}(\boldsymbol{z})$. The matrix $\tilde{\boldsymbol{A}}(\boldsymbol{z}) \in \mathbb{R}^{n\times n}$ is the reduced linear operator and the function $\tilde{\boldsymbol{g}} : \mathbb{R}^n \times \mathcal{D} \to \mathbb{R}^n$ is the reduced nonlinear term. The costs of constructing the reduced system (2) may be high, but are compensated if, during the outer loop, a large number of evaluations of the high-fidelity model f_{hi} can be replaced by low-fidelity model evaluations f_{lo}.

Construction of the reduced space with proper orthogonal decomposition We discuss proper orthogonal decomposition (POD) to construct the basis of the reduced space [191, 23, 173, 120, 119]. The POD basis vectors are computed from so-called "snapshots"—state vectors of the high-fidelity model at selected inputs. Let

$$\boldsymbol{U} = [\boldsymbol{u}(\boldsymbol{z}_1), \dots, \boldsymbol{u}(\boldsymbol{z}_M)] \in \mathbb{R}^{N\times M} \tag{3}$$

denote the snapshot matrix. The columns of $\boldsymbol{U}$ are $M \in \mathbb{N}$ state vectors (i.e., the snapshots) of the high-fidelity model at inputs $\boldsymbol{z}_1, \dots, \boldsymbol{z}_M \in \mathcal{D}$. It is crucial that the inputs are selected such that the snapshots cover the relevant system characteristics of the high-fidelity model. Many sampling strategies for selecting the inputs are based on greedy approaches, see [203, 181, 33, 94] and [21] for detailed discussions.

The POD basis is the orthogonal basis that solves the minimization problem

$$\min_{\boldsymbol{v}_1,\dots,\boldsymbol{v}_n \in \mathbb{R}^N} \sum_{i=1}^{M} \|\boldsymbol{u}(\boldsymbol{z}_i) - \sum_{j=1}^{n} (\boldsymbol{v}_j^T \boldsymbol{u}(\boldsymbol{z}_i))\boldsymbol{v}_j\|_2^2 ,$$

and thus the POD basis spans the optimal n-dimensional space for the approximation of the snapshots with respect to the Euclidean norm $\|\cdot\|_2$. The POD basis vectors $\boldsymbol{v}_1, \dots, \boldsymbol{v}_n$ are the left singular vectors of the snapshot matrix (3) corresponding to the n largest singular values. Let $r \in \mathbb{N}$ be the rank of the snapshot matrix $\boldsymbol{U}$ and let further $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ be the singular values. We have

$$\sum_{i=1}^{M} \|\boldsymbol{u}(\boldsymbol{z}_i) - \sum_{j=1}^{n} (\boldsymbol{v}_j^T \boldsymbol{u}(\boldsymbol{z}_i))\boldsymbol{v}_j\|_2^2 = \sum_{i=n+1}^{r} \sigma_i^2 ,$$

and therefore the singular values of $\boldsymbol{U}$ provide guidance on how many basis vectors should be used. A common approach is to choose n such that

$$\frac{\sum_{i=1}^{n} \sigma_i^2}{\sum_{i=1}^{r} \sigma_i^2} > \epsilon \,,$$

where $\epsilon > 0$ is a user-specified tolerance, typically selected to be 0.99 or higher.

Other basis generation methods POD is a popular basis generation method because it is applicable to a wide range of problems, including time-dependent and nonlinear problems [40, 94, 119, 120, 186]. There are also methods based on Krylov subspaces to generate a reduced basis [78, 85], including multivariate Padé approximations and tangential interpolation for linear systems [15, 20, 84, 90]. Another basis generation approach is based on centroidal Voronoi tessellation (CVT) [70], where a special Voronoi clustering of the snapshots is constructed. The reduced basis is then derived from the generators of the Voronoi clustering. The work [35] discusses details on CVT-based basis construction. A combination of POD and CVT-based basis construction is introduced in [71]. Dynamic mode decomposition is another basis generation method that is popular in the context of computational fluid dynamics [187, 199, 168]. Balanced truncation [142, 143] is a common basis construction method used in the systems and control theory community. For stable linear time-invariant systems, balanced truncation provides a basis that guarantees asymptotically stable reduced systems and provides an error bound [10, 91]. Another basis generation approach is the reduced basis method [181, 89, 182, 180], where orthogonalized carefully selected snapshots are the basis vectors. Depending on the problem of interest, these reduced basis models can be equipped with cheap *a posteriori* error estimators for the reduced model outputs [107, 89, 95, 181, 200, 211]. Efficient error estimators can also sometimes be provided for other basis generation methods, such as the POD [104, 105].

Construction of the reduced system Once the basis vectors in $\boldsymbol{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n] \in \mathbb{R}^{N \times n}$ are generated, the reduced operator $\tilde{\boldsymbol{A}}(\boldsymbol{z}) \in \mathbb{R}^{n \times n}$ is constructed via, e.g., Galerkin projection as

$$\tilde{\boldsymbol{A}}(\boldsymbol{z}) = \boldsymbol{V}^T \boldsymbol{A}(\boldsymbol{z}) \boldsymbol{V} \,. \tag{4}$$

The operator $\boldsymbol{A}(\boldsymbol{z})$ depends on the input $\boldsymbol{z}$ and therefore the reduced operator $\tilde{\boldsymbol{A}}(\boldsymbol{z})$ cannot be pre-computed. Thus, to solve the reduced system for an input $\boldsymbol{z} \in \mathcal{D}$, the operator $\boldsymbol{A}(\boldsymbol{z})$ of the high-fidelity model f_{hi} is assembled and then the reduced operator $\tilde{\boldsymbol{A}}(\boldsymbol{z})$ is constructed via Galerkin projection as in (4). This is computationally expensive, which calls for techniques that approximate the reduced operator $\tilde{\boldsymbol{A}}(\boldsymbol{z})$ without requiring assembly of $\boldsymbol{A}(\boldsymbol{z})$ first [155, 7, 67]. Similarly, the reduced nonlinear function $\tilde{\boldsymbol{g}}$ cannot be pre-computed. Sparse sampling methods have been introduced in the context of model reduction, to derive computationally cheap interpolants $\tilde{\boldsymbol{g}}$ of a nonlinear function $\boldsymbol{g}$, see, e.g., [17, 12, 38, 40, 69, 73, 158, 44].

Proper generalized decomposition and incremental reduction methods Traditionally, a projection-based low-fidelity model is constructed with one-time high-computational costs and then stays fixed while it is used by the outer-loop application. The work [183, 184] introduces the *a priori* hyper-reduction (APHR) method. The APHR method leads to an incremental algorithm to build the bases as more snapshot data become available. To obtain the incremental updates, the APHR alternates between a reduction step and an enrichment step. We refer to [47, 48, 183, 184] for details on the approach and applications.

Proper generalized decomposition (PGD) [46, 49, 121] is another reduction technique, which is related to APHR and builds incrementally a separated representation of the high-fidelity model f_{hi}. In PGD, the multi-dimensional function $f_{\text{hi}}(\boldsymbol{z}) = f_{\text{hi}}(z_1, \dots, z_d)$ is approximated as a sum of products of low-dimensional functions. For example, if each of the low-dimensional functions is one-dimensional, then one obtains

$$f_{\text{hi}}(z_1, \dots, z_d) \approx f_{\text{lo}}(z_1, \dots, z_d) = \sum_{i=1}^{l} \prod_{j=1}^{d} f_j^{(i)}(z_j)\,,$$

where $f_j^{(i)}, i = 1, \dots l, j = 1, \dots, d$ are one-dimensional functions and $l \in \mathbb{N}$ is the number of terms. PGD has been successfully applied to a wide range of problems, including high-dimensional problems with $d \gg 1$. We refer to the review papers [46, 49], the literature [6, 45, 50, 77, 153, 154] and the references therein for details on PGD and its applications.

3.3 Data-fit low-fidelity models

This section considers the construction of low-fidelity models from high-fidelity models using data-fit techniques. These methods require only the inputs and corresponding outputs of the high-fidelity model, and thus they are applicable to black-box high-fidelity models. Let $\boldsymbol{z}_1, \dots, \boldsymbol{z}_M \in \mathcal{D}$ be $M \in \mathbb{N}$ inputs and let $f_{\text{hi}}(\boldsymbol{z}_1), \dots, f_{\text{hi}}(\boldsymbol{z}_M) \in \mathcal{Y}$ be the corresponding outputs computed with the high-fidelity model. Data-fit techniques use the data set

$$\{(\boldsymbol{z}_1, f_{\text{hi}}(\boldsymbol{z}_1)), \dots, (\boldsymbol{z}_M, f_{\text{hi}}(\boldsymbol{z}_M))\} \subset \mathcal{D} \times \mathcal{Y} \tag{5}$$

to derive a function $f_{\text{lo}} : \mathcal{D} \to \mathcal{Y}$ that is computationally cheap to evaluate and that approximates f_{hi} well for $\boldsymbol{z} \in \mathcal{D}$ in a certain metric. Note that we assume scalar outputs $f_{\text{hi}}(\boldsymbol{z}_1), \dots, f_{\text{hi}}(\boldsymbol{z}_M) \in \mathcal{Y} \subset \mathbb{R}$ for the sake of exposition here. In case of vector-valued outputs, a separate low-fidelity model is constructed for each component of the output.

3.3.1 Interpolation

We first consider interpolation approaches, i.e., where

$$f_{\text{lo}}(\boldsymbol{z}_i) = f_{\text{hi}}(\boldsymbol{z}_i)\,, \qquad i = 1, \dots, M\,, \tag{6}$$

for the inputs $\boldsymbol{z}_1, \dots, \boldsymbol{z}_M$ in the data set (5). We consider low-fidelity models that are a weighted sum of M basis functions $\phi_1, \dots, \phi_M : \mathcal{D} \to \mathbb{R}$ and weights $\boldsymbol{w} =$

$[w_1, \dots, w_M]^T \in \mathbb{R}^M$, i.e.,

$$f_{\text{lo}}(\boldsymbol{z}) = \sum_{i=1}^{M} w_i \phi_i(\boldsymbol{z}) \,. \tag{7}$$

The M weights $w_1, \dots, w_M$ are usually derived from the M equations given by the interpolation conditions (6). The choice of the basis functions $\phi_1, \dots, \phi_M$ is a more delicate matter. First, the basis functions greatly influence the approximation quality of the low-fidelity model. Second, the basis functions often depend on hyperparameters that allow for more accurate approximations but that also introduce additional complexity into the construction of the low-fidelity model f_{lo}. Third, numerical properties, such as the conditioning of the system of equations used to derive the weights, also depend on the choice of the basis functions. We discuss different choices of the basis functions and the construction of the corresponding low-fidelity models in more detail in the following paragraphs.

Polynomial interpolation Polynomials are classical basis functions that can be used in approximations of the form (7), e.g., Lagrange polynomials. Piecewise-polynomial interpolation approaches allow to use low-degree polynomials, which avoids problems with global polynomial interpolation of high degree, e.g., Runge's phenomenon [64, p. 99]. If the input $\boldsymbol{z}$ of f_{hi} is low dimensional, polynomial interpolants can be derived with tensor product approaches. In higher-dimensional settings, discretization methods based on sparse grids [34, 165] can be employed.

Radial basis functions Radial basis functions based on the Gaussian density function are widely used because they lead to a numerically stable computation of the weights under certain conditions. Gaussian radial basis functions are of the form

$$\phi(\boldsymbol{z}_i, \boldsymbol{z}_j) = \exp\left(\frac{-\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2^2}{2\sigma^2}\right) \tag{8}$$

and depend on the hyperparameter $\sigma \in \mathbb{R}$. The basis functions $\phi_1, \dots, \phi_M$ used in (7) are derived by setting $\phi_i(\boldsymbol{z}) = \phi(\boldsymbol{z}_i, \boldsymbol{z})$ for all $i = 1, \dots, M$. Note that σ is shared among all basis functions, instead of being a separate hyperparameter for each basis function. The weights are determined by solving the system of linear equations

$$\boldsymbol{\Phi} \boldsymbol{w} = \boldsymbol{y} \,, \tag{9}$$

where $\Phi \in \mathbb{R}^{M \times M}$ is a matrix with $\Phi_{ij} = \phi_i(\boldsymbol{z}_j)$ and $\boldsymbol{y} = [f_{\text{hi}}(\boldsymbol{z}_1), \dots, f_{\text{hi}}(\boldsymbol{z}_M)]^T \in \mathbb{R}^M$. The system (9) guarantees the interpolation condition (6) for f_{lo}. Gaussian radial basis functions derived from (8) lead to a symmetric positive definite matrix $\boldsymbol{\Phi}$ under certain conditions and therefore to a system of linear equations that can be solved in a numerically stable way with a Cholesky factorization, see [80] for details.

The hyperparameter σ in (8) is often estimated via cross validation. Thus, the derivation of the low-fidelity model f_{lo} using the Gaussian radial basis function consists of two nested loops. In the outer loop the hyperparameter σ is estimated (e.g., using cross

validation) and in the inner loop the weights $\boldsymbol{w}$ are determined from the system of linear equations (9) for the current hyperparameter. A quick hyperparameter estimation is of particular importance in optimization, where the radial-basis model is updated in each iteration of the optimization process [1, 174, 175].

Kriging Kriging [114, 136, 140, 172, 185] represents the high-fidelity model f_{hi} as a stochastic process

$$b + \epsilon(\boldsymbol{z})\,, \tag{10}$$

where $b \in \mathbb{R}$ is the mean of this stochastic process and $\epsilon(\boldsymbol{z})$ is a stochastic process with zero-mean Gaussian distribution and standard deviation $\sigma \in \mathbb{R}$. The process ϵ represents the lack of knowledge due to the finite number of available samples of the high-fidelity model. Thus, the data in (5) are treated as if they were computed from a model of the form (10). The correlation between the inputs is modeled as

$$\operatorname{corr}(\epsilon(\boldsymbol{z}_i), \epsilon(\boldsymbol{z}_j)) = \exp\left(-\sum_{k=1}^{d} \theta_k |z_i^{(k)} - z^{(k)}|^{p_k}\right)\,, \tag{11}$$

where $\boldsymbol{z}_i = [z_i^{(1)}, \ldots, z_i^{(d)}]^T \in \mathcal{D}$ and $\boldsymbol{z} = [z^{(1)}, \ldots, z^{(d)}]^T \in \mathcal{D}$, and $\theta_1, \ldots, \theta_d \in \mathbb{R}$ and $p_1, \ldots, p_d \in \mathbb{R}$ are hyperparameters. We define the basis functions

$$\phi_i(\boldsymbol{z}) = \operatorname{corr}(\epsilon(\boldsymbol{z}_i), \epsilon(\boldsymbol{z}))\,, \tag{12}$$

for $i = 1, \ldots, M$. Note that the basis functions (12) are similar to the Gaussian radial basis functions (8) but have $2d$ hyperparameters.

The hyperparameters $\theta_1, \ldots, \theta_d \in \mathbb{R}$, $p_1, \ldots, p_d \in \mathbb{R}$, the mean b, and the standard deviation σ of the process ϵ are estimated using a maximum likelihood approach with respect to the given data (5). This hyperparameter estimation is often the computationally most expensive part of creating a kriging model [1]. We refer to, e.g., [82] for details on computational strategies for estimating the parameters. We also refer to [80, 109] for a detailed study of the effect of the hyperparameters on the properties of the kriging model.

Having estimated the hyperparameters, the weights $\boldsymbol{w} = [w_1, \ldots, w_M]^T \in \mathbb{R}^M$ are computed by solving the system of equations (9) with

$$\Phi_{ij} = \operatorname{corr}(\epsilon(\boldsymbol{z}_i), \epsilon(\boldsymbol{z}_j)) = \phi_i(\boldsymbol{z}_j)\,.$$

The kriging model is evaluated at an input $\boldsymbol{z} \in \mathcal{D}$ by

$$f_{\text{lo}}(\boldsymbol{z}) = \boldsymbol{\phi}^T \boldsymbol{w} + \hat{b} - \boldsymbol{\phi}^T \boldsymbol{\Phi}^{-1} \hat{b} \mathbf{1}\,, \tag{13}$$

where $\boldsymbol{\phi} = [\phi_1(\boldsymbol{z}), \ldots, \phi_M(\boldsymbol{z})]^T \in \mathbb{R}^M$ contains the basis functions evaluated at the input $\boldsymbol{z}$, and the mean is estimated as

$$\hat{b} = \frac{\mathbf{1}^T \boldsymbol{\Phi}^{-1} \boldsymbol{y}}{\mathbf{1}^T \boldsymbol{\Phi}^{-1} \mathbf{1}}.$$

The term $\boldsymbol{\phi}^T \boldsymbol{\Phi}^{-1} \hat{b} \mathbf{1}$ approximates the constant $\hat{b}$ as a linear combination of the basis $\phi_1, \ldots, \phi_M$. Note that (13) can be transformed into (7) by adding an additional basis function that is constant at all inputs in $\mathcal{D}$.

An error indicator of the kriging model is given by the mean-squared error (MSE) of (13) at $\boldsymbol{z} \in \mathcal{D}$. Such an error indicator is useful for adapting the kriging model, i.e., to select a new input $\boldsymbol{z}_{M+1} \in \mathcal{D}$ at which to evaluate the high-fidelity model and extend the kriging model with the obtained output. We refer to, e.g., [185] for a derivation of the MSE, and to [109] and Section 7.2 for an adaptivity strategy that is based on the MSE.

We described ordinary kriging. There is also universal kriging where the mean of the stochastic process model is a function depending on the input. Such a mean term allows fine-tuning of the kriging model by capturing a trend component, see the discussion in [80, 112]. In blind kriging [59, 110, 111], the trend of the mean term is learned from data.

3.3.2 Regression approaches

Support vector machines (SVMs) construct regression models that can be used to derive low-fidelity models. SVMs explicitly allow the low-fidelity model outputs to vary from the high-fidelity model outputs by a given threshold tolerance. We briefly survey classical regression approaches, discuss linear support vector regression, and then briefly generalize support vector regression to the nonlinear case.

Classical least-squares regression A classical regression approach is based orthogonal polynomial expansion. The lth-degree polynomial expansion expresses $f_{\text{hi}}(\boldsymbol{z})$ in a polynomial basis as

$$f_{\text{hi}}(\boldsymbol{z}) = \sum_{|\boldsymbol{i}|=0}^{l} w_{\boldsymbol{i}} \phi_{\boldsymbol{i}}(\boldsymbol{z}) \,, \tag{14}$$

where $l \in \mathbb{N}$ is an integer, $\boldsymbol{i} = [i_1, \ldots, i_d]^T \in \mathbb{N}^d$ is a multi-index, and $|\boldsymbol{i}| = i_1 + \ldots i_d$. The basis functions $\phi_{\boldsymbol{i}}$ are d-variate orthonormal polynomials of degree up to l satisfying

$$\int_{\mathcal{D}} \phi_{\boldsymbol{i}}(\boldsymbol{z}) \phi_{\boldsymbol{j}}(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} = \delta_{\boldsymbol{i},\boldsymbol{j}} \,, \qquad 0 \leq |\boldsymbol{i}|, |\boldsymbol{j}| \leq l \,,$$

where $\delta_{\boldsymbol{i},\boldsymbol{j}} = \prod_{l=1}^{d} \delta_{i_l, i_j}$. The expansion coefficients $w_{\boldsymbol{i}}$ are given by

$$w_{\boldsymbol{i}} = \int_{\mathcal{D}} f_{\text{hi}}(\boldsymbol{z}) \phi_{\boldsymbol{i}}(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} \,, \qquad 0 \leq |\boldsymbol{i}| \leq l \,.$$

The expansion (14) converges if $f_{\text{hi}}(\boldsymbol{z})$ is square integrable over $\mathcal{D}$.

Orthogonal polynomial expansion is referred to as polynomial chaos expansion if the input $\boldsymbol{z}$ is a random variable. The polynomials are then orthogonal with respect to the distribution of the random variable $\boldsymbol{z}$. The distribution of $\boldsymbol{z}$ therefore determines the

polynomials, see [209] for a detailed discussion. The expansion coefficients $w_{\boldsymbol{i}}$ are then given by

$$w_{\boldsymbol{i}} = \mathbb{E}[f_{\text{hi}}(\boldsymbol{z})\phi_{\boldsymbol{i}}(\boldsymbol{z})]\,, \qquad 0 \leq |\boldsymbol{i}| \leq l\,,$$

where the expectation is taken with respect to the random variable $\boldsymbol{z}$. Two popular approaches to numerically approximate the expansion coefficients are stochastic collocation and stochastic Galerkin projection [207]. We refer to [130, 208, 209, 139, 207] for an introduction to polynomial chaos expansion and further details.

Linear support vector regression Consider a linear regression model

$$f_{\text{lo}}(\boldsymbol{z}) = \boldsymbol{w}^T\boldsymbol{z} + b\,, \tag{15}$$

where $\boldsymbol{w} = [w_1, \ldots, w_d]^T \in \mathbb{R}^d$ are weights and $b \in \mathbb{R}$ is the bias term. For the sake of exposition, we first assume that for a tolerance $\epsilon > 0$, a low-fidelity model f_{lo} of the form (15) exists such that

$$|f_{\text{lo}}(\boldsymbol{z}_i) - f_{\text{hi}}(\boldsymbol{z}_i)| \leq \epsilon \tag{16}$$

for all $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_M$. Thus, the low-fidelity model outputs match the high-fidelity model outputs within a tolerance ϵ. Note that (16) holds only for the data (5) and not for all inputs in $\mathcal{D}$. The weights $\boldsymbol{w}$ for such a model f_{lo} are a solution of the constrained minimization problem

$$\begin{aligned}
\min \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\
\text{s.t.} \quad & \boldsymbol{w}^T\boldsymbol{z}_i + b - f_{\text{hi}}(\boldsymbol{z}_i) \leq \epsilon\,, && i = 1, \ldots, M\,, \\
& f_{\text{hi}}(\boldsymbol{z}_i) - \boldsymbol{w}^T\boldsymbol{z}_i - b \leq \epsilon\,, && i = 1, \ldots, M\,.
\end{aligned}$$

This is a quadratic optimization problem. The constraints are chosen such that condition (16) is enforced.

In most situations, however, a low-fidelity model of the form (15) does not exist for a given tolerance ϵ. Therefore, slack variables $\xi_i^+, \xi_i^- \in \mathbb{R}$ for $i = 1, \ldots, M$ are introduced that relax the constraints. The optimization problem becomes

$$\min \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \gamma \sum_{i=1}^{M} (\xi_i^+ + \xi_i^-) \tag{17}$$

$$\text{s.t.} \quad \boldsymbol{w}^T\boldsymbol{z}_i + b - f_{\text{hi}}(\boldsymbol{z}_i) \leq \epsilon + \xi_i^+\,, \qquad i = 1, \ldots, M \tag{18}$$

$$f_{\text{hi}}(\boldsymbol{z}_i) - \boldsymbol{w}^T\boldsymbol{z}_i - b \leq \epsilon + \xi_i^-\,, \qquad i = 1, \ldots, M \tag{19}$$

$$\xi_i^+, \xi_i^- \geq 0\,, \qquad i = 1, \ldots, M \tag{20}$$

where the cost parameter $\gamma \in \mathbb{R}$ controls the slack variable penalty. The optimization problem (17)-(20) is solved by introducing Lagrange multipliers [25, Section 7.1.4], leading to the weights

$$\boldsymbol{w} = \sum_{i=1}^{M} \hat{w}_i \boldsymbol{z}_i\,, \tag{21}$$

where $\hat{w}_1, \dots, \hat{w}_M \in \mathbb{R}$ are derived from the Lagrange multipliers of the reformulated optimization problem. The bias term b is derived from the weights [25, p. 343]. We refer to [25, 188] for details on solving the optimization problem numerically. Combining the low-fidelity model (15) with the weights (21) and the bias term b results in the low-fidelity model

$$f_{\text{lo}}(\boldsymbol{z}) = \boldsymbol{w}^T \boldsymbol{z} + b = \sum_{i=1}^{M} \hat{w}_i \boldsymbol{z}_i^T \boldsymbol{z} + b \,. \tag{22}$$

Nonlinear support vector regression We now consider the situation where the data (5) cannot be approximated well with a linear regression model such as the linear SVM model (22). Nonlinear SVMs introduce a map $\phi : \mathcal{D} \to \mathcal{V}$ into a potentially high-dimensional space $\mathcal{V}$, such that the data $\phi(\boldsymbol{z}_1), \dots, \phi(\boldsymbol{z}_M)$ mapped into $\mathcal{V}$ can be approximated well by a linear regression model. Thus, it is necessary to find the weights of the nonlinear regression model

$$f_{\text{lo}}(\boldsymbol{z}) = \boldsymbol{w}^T \phi(\boldsymbol{z}) + b \,. \tag{23}$$

However, the dimension of the space $\mathcal{V}$ is high and often grows exponentially in the number of components of $\boldsymbol{z}$, which makes a direct computation infeasible. The key observation, which renders nonlinear SVMs computationally tractable, is that the linear model in $\mathcal{V}$ is independent of the values $\phi(\boldsymbol{z})$ and that only dot products $\phi(\boldsymbol{z}_i)^T \phi(\boldsymbol{z})$ enter the evaluation of the model, cf. (22). Such a situation can be exploited with the kernel trick.

A kernel is a function $K : \mathcal{D} \times \mathcal{D} \to \mathbb{R}$ such that

$$K(\boldsymbol{z}_i, \boldsymbol{z}_j) = \phi(\boldsymbol{z}_i)^T \phi(\boldsymbol{z}_j)$$

corresponds to the dot product based on a map $\phi : \mathcal{D} \to \mathcal{V}$. In many situations, evaluating K is computationally more efficient than directly computing $\phi(\boldsymbol{z}_i)^T \phi(\boldsymbol{z}_j)$ in the space $\mathcal{V}$. For example, the polynomial kernel of order $q \in \mathbb{N}$ is given by

$$K(\boldsymbol{z}_i, \boldsymbol{z}_j) = (\boldsymbol{z}_i^T \boldsymbol{z}_j + \mathbf{1})^q \,,$$

with the vector $\mathbf{1} = [1, \dots, 1]^T \in \mathbb{N}^d$. The costs of evaluating the polynomial kernel are bounded by $\mathcal{O}(d)$ for $\boldsymbol{z}_i, \boldsymbol{z}_j \in \mathcal{D} \subset \mathbb{R}^d$. The corresponding feature map ϕ maps into a $\mathcal{O}(d^q)$-dimensional space, leading to costs of $\mathcal{O}(d^q)$ for a direct computation of the dot product $\phi(\boldsymbol{z}_i)^T \phi(\boldsymbol{z}_j)$. The kernel trick, therefore, replaces the direct computations of dot products in the optimization problem with computationally cheap kernel evaluations. We do not discuss further details of nonlinear SVMs here but refer to the introduction in [80] and the in-depth details in [25, 188].

Support vector regression depends on several parameters. There is the threshold tolerance ϵ, the cost parameter γ, and also the choice of kernel K, i.e., the choice of basis functions. Similarly to kriging, the parameters are usually estimated with cross validation, see, e.g., the implementation libsvm [39].

3.4 Information sources beyond models

We defined a model to be a function $f : \mathcal{D} \to \mathcal{Y}$ that maps from an input $\boldsymbol{z} \in \mathcal{D}$ to an output $\boldsymbol{y} \in \mathcal{Y}$. Each function evaluation incurs costs $c \in \mathbb{R}^+$ and provides an approximation of the output of interest of the system with a certain accuracy. This general definition extends to information sources that go beyond the classical computational model. For example, consider expert opinions. The expert is a function f that maps questions (inputs) to answers (outputs). Eliciting an expert opinion costs time and/or money (costs) and we have a certain trust in the answer of the expert (accuracy). Thus, an expert can be seen as an information source.

Another common information source is experimental data. Instead of performing a simulation, a physical experiment under certain conditions (input) is conducted and the result of the experiment is interpreted (output). Conducting the experiment incurs certain costs, again related to time and money. The accuracy of the result of the experiment depends on factors such as measurement noise and material perturbations. Note that often experimental data is available from the literature and can be reused without performing a new experiment. In such situations, it might be possible to incorporate the data without additional costs; except the costs to bring the data into a useful form.

Other examples of information sources include lookup tables, historical data, and operational data. We restrict the following discussion on multifidelity model management methods to models, because all of the multifidelity methods that we survey are developed in the context of models; however, we note that many of these multifidelity methods could potentially be extended to this broader class of information sources.

4 Multifidelity model management strategies

The model management in multifidelity methods defines how different models are employed during execution of the outer loop and how outputs from different models are combined. Models are managed such that low-fidelity models are leveraged for speedup, while judicious evaluations of the high-fidelity model establish accuracy and/or convergence of the outer-loop result. This section describes a categorization of model management methods into three types of strategies. The following sections then survey specific model management methods in the context of uncertainty propagation, statistical inference, and optimization.

As shown in Figure 3, we distinguish between three types of model management strategies: adaptation, fusion, and filtering.

4.1 Adaptation

The first model management strategy uses adaptation to enhance the low-fidelity model with information from the high-fidelity model while the computation proceeds. One example of model management based on adaptation is global optimization with EGO, where a kriging model is adapted in each iteration of the optimization process [109, 202]. Another example is the correction of low-fidelity model outputs via updates, which are

derived from the high-fidelity model. It is common to use additive updates, which define the correction based on the difference between sampled high-fidelity and low-fidelity outputs, and/or multiplicative updates, which define the correction based on the ratio between sampled high-fidelity and low-fidelity outputs [2, 4]. The correction model is then typically built using Taylor series expansion based on gradients, and possibly also on higher-order derivative information [74]. In [115], low-fidelity models are corrected (calibrated) with Gaussian process models to best predict the output of the high-fidelity model. Another multifidelity adaptation strategy is via adaptive model reduction, where projection-based reduced models are efficiently adapted as more data of the high-fidelity model become available during solution of the outer-loop application problem. Key to online adaptive model reduction is an efficient adaptation process. In [161, 162], the basis and operators of projection-based reduced models are adapted with low-rank updates. In [37], an h-adaptive refinement of the basis vectors uses clustering algorithms to learn and adapt a reduced basis from high-fidelity model residuals. The work [8] adapts localized reduced bases to smooth the transition from one localized reduced basis to another localized basis.

4.2 Fusion

The second model management strategy is based on information fusion. Approaches based on fusion evaluate low- and high-fidelity models and then combine information from all outputs. An example from uncertainty propagation is the control variate framework [31, 99, 146], where the variance of Monte Carlo estimators is reduced by exploiting the correlation between high- and low-fidelity models. The control variate framework leverages a small number of high-fidelity model evaluations to obtain unbiased estimators of the statistics of interest, together with a large number of low-fidelity model evaluations to obtain an estimator with a low variance. Another example from uncertainty propagation is the fusion framework introduced in [118], which is based on Bayesian regression.

Co-kriging is another example of a multifidelity method that uses model management based on fusion. Co-kriging derives a model from multiple information sources, e.g., a low- and a high-fidelity model [9, 144, 164]. Co-kriging is often used in the context of optimization if gradient information of the high-fidelity model is available, see [81]. The work [123] compares kriging and co-kriging models on aerodynamic test functions. In [52], gradients are computed cheaply with the adjoint method and then used to derive a co-kriging model for design optimization in large design spaces. The authors of [210] obtain gradient and Hessian information with the adjoint method and automatic differentiation, and combine gradient and Hessian information into a co-kriging model. In [100], co-kriging with gradients and further developments of co-kriging are compared for approximating aerodynamic models of airfoils.

4.3 Filtering

The third model management strategy is based on filtering, where the high-fidelity model is invoked following the evaluation of a low-fidelity filter. This might entail evaluating the high-fidelity model only if the low-fidelity model is deemed inaccurate, or it might entail evaluating the high-fidelity model only if the candidate point meets some criterion based on the low-fidelity evaluation. One example of a multifidelity filtering strategy is a multi-stage adaptive delayed acceptance MCMC algorithm. For example, in two-stage MCMC [51, 83], a candidate sample needs to be first accepted by the likelihood induced by the low-fidelity model before the high-fidelity model is evaluated at the candidate sample. As another example, in the multifidelity stochastic collocation approach in [145], the stochastic space is explored with the low-fidelity model to derive sampling points at which the high-fidelity model is then evaluated. A third example is multifidelity importance sampling, where the sampling of the high-fidelity model is guided by an importance sampling biasing distribution that is constructed with a low-fidelity model [159].

5 Multifidelity model management in uncertainty propagation

Inputs to systems are usually not known exactly due to measurement errors, noise and other perturbations, and therefore inputs are often modeled as random variables. With random inputs, the output of the system becomes a random variable as well. Uncertainty propagation aims to estimate statistics of the output random variable [138]. Sampling-based methods for uncertainty propagation evaluate the model of the system at a large number of inputs and then estimate statistics from the corresponding model outputs. Examples of sampling-based methods are Monte Carlo simulation and stochastic collocation approaches. In this section, we review multifidelity approaches for sampling-based methods in uncertainty propagation. These multifidelity approaches shift many of the model evaluations to low-fidelity models while evaluating the high-fidelity model a small number of times to establish unbiased estimators. Section 5.1 introduces the problem setup and briefly overviews the Monte Carlo simulation method. Sections 5.2 and 5.3 discuss multifidelity methods for Monte Carlo based on control variates and importance sampling, respectively. Multifidelity methods for stochastic collocation are discussed in Section 5.4.

5.1 Uncertainty propagation and Monte Carlo simulation

Consider the high-fidelity model $f_{\text{hi}} : \mathcal{D} \to \mathcal{Y}$ and let the uncertainties in the inputs be represented by a random variable Z with probability density function p. The goal of uncertainty propagation is to estimate statistics of the random variable $f_{\text{hi}}(Z)$, e.g., the expectation

$$\mathbb{E}[f_{\text{hi}}] = \int_{\mathcal{D}} f_{\text{hi}}(\boldsymbol{z}) p(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} \,, \tag{24}$$

and the variance

$$\mathbb{V}[f_{\text{hi}}] = \mathbb{E}[f_{\text{hi}}^2] - \mathbb{E}[f_{\text{hi}}]^2 \,. \tag{25}$$

The Monte Carlo method draws $m \in \mathbb{N}$ independent and identically distributed (i.i.d.) realizations $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_m \in \mathcal{D}$ of the random variable Z, and estimates the expectation $\mathbb{E}[f_{\text{hi}}]$ as

$$\bar{s}_m^{\text{hi}} = \frac{1}{m} \sum_{i=1}^{m} f_{\text{hi}}(\boldsymbol{z}_i) \,. \tag{26}$$

The Monte Carlo estimator is an unbiased estimator $\bar{s}_m^{\text{hi}}$ of $\mathbb{E}[f_{\text{hi}}]$, which means that $\mathbb{E}[\bar{s}_m^{\text{hi}}] = \mathbb{E}[f_{\text{hi}}]$. The root mean squared error (RMSE) of the Monte Carlo estimator $\bar{s}_m^{\text{hi}}$ therefore is

$$e(\bar{s}_m^{\text{hi}}) = \sqrt{\frac{\mathbb{V}[f_{\text{hi}}]}{m}} \,, \tag{27}$$

if the variance $\mathbb{V}[f_{\text{hi}}]$ is finite [177].

The convergence rate $\mathcal{O}(m^{-1/2})$ of the RMSE (27) is low if compared to deterministic quadrature rules, see Section 5.4; however, the rate is independent of the dimension d of the input $\boldsymbol{z}$, which means that the Monte Carlo method is well-suited for high dimensions d, and, in fact, is often the only choice available if d is large. Typically more important in practice, however, is the pre-asymptotic behavior of the RMSE of the Monte Carlo estimator. In the pre-asymptotic regime, the variance $\mathbb{V}[f_{\text{hi}}]$ dominates the RMSE. Variance reduction techniques reformulate the estimation problem such that a function with a lower variance is integrated instead of directly integrating $f_{\text{hi}}(Z)$. Examples of variance reduction techniques are antithetic variates, control variates, importance sampling, conditional Monte Carlo sampling, and stratified sampling [99, 177]. Variance reduction techniques often exploit the correlation between the random variable $f_{\text{hi}}(Z)$ of interest and an auxiliary random variable. Multifidelity methods construct the auxiliary random variable using low-fidelity models. We discuss multifidelity methods for variance reduction based on control variates in Section 5.2, and variance reduction based on importance sampling in Section 5.3.

5.2 Multifidelity uncertainty propagation based on control variates

The control variate framework [31, 99, 146] aims to reduce the estimator variance of a random variable by exploiting the correlation with an auxiliary random variable. In the classical control variate method, the auxiliary random variable is known [146]. Extensions relax this requirement by estimating the statistics of the auxiliary random variable from prior information [75, 157]. We now discuss multifidelity approaches that construct auxiliary random variables from low-fidelity models.

5.2.1 Control variates based on low-fidelity models

Consider a low-fidelity model f_{lo} of the high-fidelity model f_{hi}. In [148], the random variable $f_{\text{lo}}(Z)$ stemming from the low-fidelity model f_{lo} is used as a control variate for the random variable $f_{\text{hi}}(Z)$ of the high-fidelity model. Let $m \in \mathbb{N}$ be the number

of high-fidelity model evaluations, and let $m' \in \mathbb{N}$ be the number of low-fidelity model evaluations, with $m \ll m'$. The multifidelity approach presented in [148] draws m' realizations

$$\boldsymbol{z}_1, \ldots, \boldsymbol{z}_{m'} \tag{28}$$

from the random variable Z and computes the model outputs $f_{\text{hi}}(\boldsymbol{z}_1), \ldots, f_{\text{hi}}(\boldsymbol{z}_m)$ and $f_{\text{lo}}(\boldsymbol{z}_1), \ldots, f_{\text{lo}}(\boldsymbol{z}_{m'})$. These model outputs are used to derive the Monte Carlo estimates

$$\bar{s}_m^{\text{hi}} = \frac{1}{m} \sum_{i=1}^{m} f_{\text{hi}}(\boldsymbol{z}_i)\,, \qquad \bar{s}_m^{\text{lo}} = \frac{1}{m} \sum_{i=1}^{m} f_{\text{lo}}(\boldsymbol{z}_i)\,, \tag{29}$$

and

$$\bar{s}_{m'}^{\text{lo}} = \frac{1}{m'} \sum_{i=1}^{m'} f_{\text{lo}}(\boldsymbol{z}_i)\,. \tag{30}$$

Note that the estimates (29) use the first $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_m$ realizations of (28) only, whereas the estimate (30) uses all m' realizations. Following [148], the multifidelity estimator of $\mathbb{E}[f_{\text{hi}}]$ is

$$\bar{s}^{\text{MF}} = \bar{s}_m^{\text{hi}} + \alpha \left(\bar{s}_{m'}^{\text{lo}} - \bar{s}_m^{\text{lo}} \right)\,. \tag{31}$$

The control variate coefficient $\alpha \in \mathbb{R}$ balances the term $\bar{s}_m^{\text{hi}}$ stemming from the high-fidelity model and the term $\bar{s}_{m'}^{\text{lo}} - \bar{s}_m^{\text{lo}}$ from the low-fidelity model. The multifidelity estimator (31) based on the control variate framework evaluates the high- and the low-fidelity model and fuses both outputs into an estimate of the statistics of the high-fidelity model. The multifidelity estimator (31) therefore uses a model management based on fusion, see Section 4. We note that (31) could also be viewed as a correction, although the correction is to the estimator $\bar{s}$, not to the low-fidelity model outputs directly.

Properties of the multifidelity estimator The multifidelity estimator $\bar{s}^{\text{MF}}$ is an unbiased estimator of $\mathbb{E}[f_{\text{hi}}]$ because

$$\mathbb{E}[\bar{s}^{\text{MF}}] = \mathbb{E}[\bar{s}_m^{\text{hi}}] + \alpha \mathbb{E}[\bar{s}_{m'}^{\text{lo}} - \bar{s}_m^{\text{lo}}] = \mathbb{E}[f_{\text{hi}}]\,.$$

Therefore, the RMSE of the estimator $\bar{s}^{\text{MF}}$ is equal to the variance $\mathbb{V}[\bar{s}^{\text{MF}}]$ of the estimator, $e(\bar{s}^{\text{MF}}) = \mathbb{V}[\bar{s}^{\text{MF}}]$. The costs of the multifidelity estimator are

$$c(\bar{s}^{\text{MF}}) = m c_{\text{hi}} + m' c_{\text{lo}}\,,$$

because the high-fidelity model is evaluated at m realizations and the low-fidelity model at m' realizations of Z.

5.2.2 Multifidelity Monte Carlo

The multifidelity estimator (31) depends on the control variate coefficient α and on the number of realizations m and m'. In [148, 163], these parameters are chosen such that

the RMSE of the estimator (31) is minimized for a given computational budget $\gamma \in \mathbb{R}_+$. The solution to the optimization problem

$$\begin{aligned} \min_{\alpha,m,m' \in \mathbb{R}} \quad & e(\bar{s}^{\mathrm{MF}}) \\ \text{s.t.} \quad & m > 0 \\ & m' \geq m \\ & m c_{\mathrm{hi}} + m' c_{\mathrm{lo}} = \gamma \end{aligned} \tag{32}$$

gives the coefficient α^* and the number of model evaluations m^* and m'^* that minimize the RMSE of the multifidelity estimator $\bar{s}^{\mathrm{MF}}$ for a given computational budget γ. The constraints impose that $0 < m \leq m'$ and that the costs $c(\bar{s}^{\mathrm{MF}})$ of the estimator equal the computational budget γ.

Variance of the multifidelity estimator Since the multifidelity estimator $\bar{s}^{\mathrm{MF}}$ is unbiased, we have $e(\bar{s}^{\mathrm{MF}}) = \mathbb{V}[\bar{s}^{\mathrm{MF}}]$, and therefore the objective of minimizing the RMSE $e(\bar{s}^{\mathrm{MF}})$ can be replaced with the variance $\mathbb{V}[\bar{s}^{\mathrm{MF}}]$ in the optimization problem (32). The variance $\mathbb{V}[\bar{s}^{\mathrm{MF}}]$ of the multifidelity estimator $\bar{s}^{\mathrm{MF}}$ is

$$\mathbb{V}[\bar{s}^{\mathrm{MF}}] = \frac{\sigma_{\mathrm{hi}}^2}{m} + \left(\frac{1}{m} - \frac{1}{m'}\right)\left(\alpha^2 \sigma_{\mathrm{lo}}^2 - 2\alpha\rho\sigma_{\mathrm{hi}}\sigma_{\mathrm{lo}}\right), \tag{33}$$

where $-1 \leq \rho \leq 1$ is the Pearson correlation coefficient of the random variable $f_{\mathrm{hi}}(Z)$ and $f_{\mathrm{lo}}(Z)$, and

$$\sigma_{\mathrm{hi}}^2 = \mathbb{V}[f_{\mathrm{hi}}], \qquad \sigma_{\mathrm{lo}}^2 = \mathbb{V}[f_{\mathrm{lo}}],$$

are the variances of $f_{\mathrm{hi}}(Z)$ and $f_{\mathrm{lo}}(Z)$, respectively.

Optimal selection of the number of samples and control variate coefficients Under certain conditions on the low- and the high-fidelity model, the optimization problem (32) has a unique, analytic solution [163]

$$\alpha^* = \rho\frac{\sigma_{\mathrm{hi}}}{\sigma_{\mathrm{lo}}}, \quad m^* = \gamma\left(c_{\mathrm{hi}} + r^* c_{\mathrm{lo}}\right)^{-1}, \quad m'^* = m^* r^*, \tag{34}$$

where

$$r^* = \frac{m'^*}{m^*} = \sqrt{\frac{c_{\mathrm{hi}}\rho^2}{c_{\mathrm{lo}}(1-\rho^2)}}.$$

Interaction of models in multifidelity Monte Carlo To interpret the interaction between the high- and the low-fidelity model in the multifidelity estimator $\bar{s}^{\mathrm{MF}}$, we plug the solution α^*, m^* and m'^* into (33) and obtain

$$\mathbb{V}[\bar{s}^{\mathrm{MF}}] = \left(1 + r^*\frac{c_{\mathrm{lo}}}{c_{\mathrm{hi}}}\right)\left[1 - \left(1 - \frac{1}{r^*}\right)\rho^2\right]\frac{\sigma_{\mathrm{hi}}^2}{\gamma}. \tag{35}$$

If the low-fidelity model is cheap to evaluate, i.e., $c_{\text{lo}}/c_{\text{hi}} \to 0$, then the variance $\mathbb{V}[\bar{s}^{\text{MF}}] \to (1-\rho^2)\frac{\sigma_{\text{hi}}^2}{\gamma}$. If the low-fidelity model is an accurate approximation of the high-fidelity model, i.e., if the correlation $\rho \to 1$, then $\mathbb{V}[\bar{s}^{\text{MF}}] \to \frac{c_{\text{lo}}}{c_{\text{hi}}}\frac{\sigma_{\text{hi}}^2}{\gamma}$. This shows that a highly correlated low-fidelity model, i.e., where $\rho \approx 1$, is insufficient for variance reduction compared to the Monte Carlo estimator with the same computational budget γ; the low-fidelity model must also be cheaper to evaluate, i.e., $c_{\text{lo}} < c_{\text{hi}}$. Similar considerations with the ratio $r^* = m'^*/m^*$ show that the multifidelity method shifts model evaluations to the low-fidelity model if the low-fidelity model is cheap. i.e., if $c_{\text{lo}}/c_{\text{hi}}$ small. Similarly, if the correlation ρ is large, i.e., near 1, then r^* becomes large too, which in turn means that many low-fidelity model evaluations are performed. We refer to [163] for an in-depth discussion of the interaction between the models in multifidelity Monte Carlo estimation.

Efficiency of the multifidelity estimator It is shown in [148] that the control variate estimator (35) with the optimal α^* and r^* is computationally cheaper than the Monte Carlo method using the high-fidelity model f_{hi} if

$$\rho^2 > \frac{c_{\text{lo}}}{c_{\text{lo}} + c_{\text{hi}}}\,. \tag{36}$$

The inequality (36) emphasizes that both correlation and costs of the models are critical for an efficient multifidelity estimator.

Extensions of multifidelity Monte Carlo to more than two models In [163], the multifidelity estimator (31) is extended to an arbitrary number of low-fidelity models of any type. An optimal model management is derived that leads to control variate coefficients and number of model evaluations that minimize the RMSE of the multifidelity estimator. In the numerical experiments, high-fidelity finite element models are combined with projection-based models, data-fit models, and support vector machines, which demonstrates that the multifidelity approach is applicable to any low-fidelity model type.

5.2.3 Other uses of control variates as a multifidelity technique

In [27, 28], the reduced basis method is used to construct control variates. The reduced basis models are built with greedy algorithms that use *a posteriori* error estimators to particularly target variance reduction. The work [204] uses error estimators to combine reduced basis models with control variates. The StackMC method presented in [197] successively constructs machine-learning-based low-fidelity models and combines them with the control variate framework. In [149], the multifidelity control variate method is used in the context of optimization, where information of previous iterations of the optimization problem are used as control variate. This means that data from previous iterations serve as a kind of low-fidelity "model".

Multilevel Monte Carlo method The multilevel Monte Carlo method [101, 86] uses the control variate framework to combine multiple low-fidelity models with a high-fidelity

model. Typically, in multilevel Monte Carlo, the low-fidelity models are coarse-grid approximations, where the accuracy and costs can be controlled by a discretization parameter. The properties of the low-fidelity models are therefore often described with rates. For example, the rate of the decay of the variance of the difference of two successive coarse-grid approximations and the rate of the increase of the costs with finer grids play a critical role in determining the efficiency of the multilevel Monte Carlo method. Additionally, rates are used to determine the number of evaluations of each low-fidelity model and the high-fidelity model, see, e.g., [53, Theorem 1]. In the setting of stochastic differential equations and coarse-grid approximations, multilevel Monte Carlo has been very successful, see, e.g., [53, 195], the recent advances on multi-index Monte Carlo [98], and the nesting of multilevel Monte Carlo and control variates [151] for detailed studies and further references.

5.3 Multifidelity uncertainty propagation based on importance sampling

Importance sampling [79, 177, 192] uses a problem-dependent sampling strategy. The goal is an estimator with a lower variance than a Monte Carlo estimator such as (26). The problem-dependent sampling means that samples are drawn from a biasing distribution, instead of directly from the distribution of the random variable Z of interest, and then the change of the distribution is compensated with a re-weighting. Importance sampling is particularly useful in the case of rare event simulation, where the probability of the event of interest is small and therefore many realizations of the random variable Z are necessary to obtain a Monte Carlo estimate of reasonable accuracy. Importance sampling with a suitable biasing distribution can explicitly target the rare event and reduce the number of realizations required to achieve an acceptable accuracy. The challenge of importance sampling is the construction of a biasing distribution, which usually is problem-dependent and typically requires model evaluations. We discuss multifidelity methods that use low-fidelity models for the construction of biasing distributions and methods that combine high- and low-fidelity model evaluations during the importance sampling step.

5.3.1 Importance sampling

Consider the indicator function $I_{\text{hi}} : \mathcal{D} \to \{0, 1\}$ defined as

$$I_{\text{hi}}(\boldsymbol{z}) = \begin{cases} 1\,, & f_{\text{hi}}(\boldsymbol{z}) < 0\,, \\ 0\,, & f_{\text{hi}}(\boldsymbol{z}) \geq 0 \end{cases}.$$

We define the set $\mathcal{I} = \{\boldsymbol{z} \in \mathcal{D} \,|\, I_{\text{hi}}(\boldsymbol{z}) = 1\}$. The goal is to estimate the probability of the event $Z^{-1}(\mathcal{I})$, which is $\mathbb{E}_p[I_{\text{hi}}]$, with importance sampling. Note that we now explicitly denote in the subscript of $\mathbb{E}$ with respect to which distribution the expectation is taken.

Step 1: Construction of biasing distribution Traditionally, importance sampling consists of two steps. In the first step, the biasing distribution with density q is constructed.

Let Z' be the biasing random variable with the biasing density q. Recall that the input random variable Z with the nominal distribution has the nominal density p. Let

$$\mathrm{supp}(p) = \{\boldsymbol{z} \in \mathcal{D} : p(\boldsymbol{z}) > 0\}$$

be the support of the density p. If the support of the nominal density p is a subset of the support of the biasing density q, i.e., $\mathrm{supp}(p) \subset \mathrm{supp}(q)$, then the expectation $\mathbb{E}_p[I_{\mathrm{hi}}]$ can be rewritten in terms of the biasing density q as

$$\mathbb{E}_p[I_{\mathrm{hi}}] = \int_{\mathcal{D}} I_{\mathrm{high}}(\boldsymbol{z}) p(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} = \int_{\mathcal{D}} I_{\mathrm{hi}}(\boldsymbol{z}') q(\boldsymbol{z}') \frac{p(\boldsymbol{z}')}{q(\boldsymbol{z}')} \mathrm{d}\boldsymbol{z}' = \mathbb{E}_q\left[I_{\mathrm{hi}} \frac{p}{q} \right] ,$$

where the ratio p/q serves as a weight.

Step 2: Deriving an importance sampling estimate In the second step, the importance sampling estimator

$$\bar{s}_m^{\mathrm{IS}} = \frac{1}{m} \sum_{i=1}^{m} I_{\mathrm{hi}}(\boldsymbol{z}_i') \frac{p(\boldsymbol{z}_i')}{q(\boldsymbol{z}_i')} \tag{37}$$

is evaluated for realizations $\boldsymbol{z}_1', \ldots, \boldsymbol{z}_m' \in \mathcal{D}$ of the random variable Z'. The RMSE of the estimator (37) is

$$e(\bar{s}_m^{\mathrm{IS}}) = \sqrt{\frac{\mathbb{V}_q[I_{\mathrm{hi}} \frac{p}{q}]}{m}} . \tag{38}$$

Variance of importance sampling estimator The variance in (38) is with respect to the biasing density q, cf. Section 5.1 and the RMSE of the Monte Carlo estimator (27). Therefore, the goal is to construct a biasing distribution with

$$\mathbb{V}_q\left[I_{\mathrm{hi}} \frac{p}{q} \right] < \mathbb{V}_p[I_{\mathrm{hi}}] ,$$

to obtain an importance sampling estimator $\bar{s}_m^{\mathrm{IS}}$ that has a lower RMSE than the Monte Carlo estimator $\bar{s}_m^{\mathrm{hi}}$ for the same number of realizations m.

5.3.2 Construction of the biasing distribution with low-fidelity models

The multifidelity importance sampling approach introduced in [159] uses a low-fidelity model to construct the biasing distribution in the first step of importance sampling, and derives the statistics using high-fidelity model evaluations in step two. In that sense, multifidelity importance sampling uses a model management strategy based on filtering, see Section 4.

In step one, the low-fidelity model f_{lo} is evaluated at a large number $n \in \mathbb{N}$ of realizations $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$ of the input random variable Z. This is computationally feasible because the low-fidelity model is cheap to evaluate. A mixture model q of Gaussian distributions is fitted with the expectation-maximization algorithm to the set of realizations

$$\{\boldsymbol{z}_i \,|\, I_{\mathrm{lo}}(\boldsymbol{z}_i) = 1 \,, i = 1, \ldots, n\}$$

for which the low-fidelity model predicts the event of interest with the indicator function $I_{\text{lo}} : \mathcal{D} \to \{0, 1\}$

$$I_{\text{lo}}(\boldsymbol{z}) = \begin{cases} 1\,, & f_{\text{lo}}(\boldsymbol{z}) < 0\,, \\ 0\,, & f_{\text{lo}}(\boldsymbol{z}) \geq 0 \end{cases}\,.$$

The mixture model q serves as a biasing distribution. Note that other density estimation methods can be used instead of fitting a mixture model of Gaussian distributions with the expectation-maximization algorithm [190, 141, 160].

In step two, the high-fidelity model is evaluated at realizations $\boldsymbol{z}'_1, \ldots, \boldsymbol{z}'_m$ from the biasing random variable Z' with biasing density q. From the high-fidelity model evaluations $f_{\text{hi}}(\boldsymbol{z}'_1), \ldots, f_{\text{hi}}(\boldsymbol{z}'_m)$ an estimate of the event probability $\mathbb{E}_p[I_{\text{hi}}]$ is obtained. Under the condition that the support of the biasing density q includes the support of the nominal density p, the multifidelity importance sampling approach leads to an unbiased estimate of the probability of the event. If the low-fidelity model is sufficiently accurate, then significant runtime savings can be obtained during the construction of the biasing distribution. Note that using I_{lo} in the second step of the multifidelity approach would lead to a biased estimator of $\mathbb{E}_p[I_{\text{hi}}]$ because f_{lo} is only an approximation of f_{hi} and thus $\mathbb{E}_p[I_{\text{hi}}] \neq \mathbb{E}_p[I_{\text{lo}}]$ in general.

5.3.3 Other model management strategies for probability estimates and limit state function evaluation

In [125, 126], a multifidelity approach is presented that replaces high-fidelity model evaluations with low-fidelity model evaluations during the second step of importance sampling. Low-fidelity evaluations are used only if the low-fidelity model is sufficiently accurate for the given input, which means that a model management strategy based on fusion is employed. The switching between the models as proposed in [125, 126] requires knowledge of the error of the low-fidelity model. The authors also propose a heuristic strategy to switch between the high- and the low-fidelity model based on a threshold parameter. Similarly, the work [43] uses *a posteriori* error estimators for reduced basis models to decide if either the reduced model or the high-fidelity model should be used.

The multifidelity methods in [19, 162] all use a model management strategy based on adaptation to estimate a failure boundary. In [19], an SVM low-fidelity model is adaptively refined along a failure boundary. The work [162] adapts a nonlinear reduced model as the computation of the failure boundary proceeds.

5.4 Stochastic collocation and multifidelity

Stochastic collocation methods [14, 92] compute statistics such as the expectation (24) and the variance (25) by using a deterministic quadrature rule instead of the Monte Carlo method. The quadrature rules are often based on sparse grids [34, 150] to perform the quadrature efficiently for high-dimensional inputs.

In [147], statistics are computed using stochastic collocation, where the outputs of a low-fidelity model are corrected with a discrepancy model that accounts for the difference between the high- and the low-fidelity model. The discrepancy model is then

used to derive either an additive correction, a multiplicative correction, or a weighted combination of additive and multiplicative corrections to the low-fidelity model outputs. Thus, this is another example of model management based on adaptation, see Section 4. The authors of [147] point out that an adaptive refinement of the discrepancy model is necessary because the complexity of the discrepancy between the high- and the low-fidelity model varies distinctly in the stochastic domain. This is because low-fidelity models tend to approximate the high-fidelity model well only in certain regions of the stochastic domain, whereas in other regions they hardly match the high-fidelity model at all.

Another multifidelity stochastic collocation method is presented in [145]. This method is based on a filtering model management strategy. The low-fidelity model is first evaluated at a large number of collocation points to sample the stochastic domain. From these samples, a small number of points is selected via a greedy procedure, and the high-fidelity model is evaluated at these points. The state solutions of the high-fidelity model at the selected collocation points span a space in which approximations of the high-fidelity model states at all other sampling points are derived.

In [194], a multilevel stochastic collocation method uses a hierarchy of models to accelerate convergence. Low-fidelity models are coarse-grid approximations of the high-fidelity model. Similarly to the multilevel Monte Carlo method, a reduction of the computational complexity can be shown if the errors of the models in the hierarchy decay with a higher rate than the rate of the increase of the costs. We categorize this multilevel stochastic collocation method as model management based on fusion, because low- and high-fidelity model outputs are fused to derive an estimate of the statistics of the high-fidelity model.

6 Multifidelity model management in statistical inference

In a Bayesian setting, inverse problems are cast in a statistical formulation where the unknown input is modeled as a random variable and is described by its posterior distribution [193]. MCMC is a popular way to sample from the posterior distribution. Statistical inference raises several computational challenges, including the design of MCMC sampling schemes, the construction of approximate models that can reduce the costs of MCMC sampling, and the development of alternatives to MCMC sampling such as variational approaches [138]. Detailed discussions of these and many other important aspects of statistical inference can be found in the literature, e.g., [177, 113, 193, 128]. We focus here on a few specific aspects of statistical inference in which multifidelity methods have been used. In particular, we survey multifidelity methods that use a two-stage formulation of MCMC, where a candidate sample has to be first accepted by a low-fidelity model before it is passed on to be either accepted or rejected by the high-fidelity model. Section 6.1 describes the problem setup. Section 6.2 describes a two-stage MCMC framework and Section 6.3 discusses a framework where a low-fidelity model is adapted while it is evaluated in an MCMC algorithm.

Algorithm 1 Metropolis-Hastings

procedure MetropolisHastings(L, p_0, q, m)
  Choose a starting point $\boldsymbol{z}_0$
  for $i = 1, \dots, m$ **do**
    Draw candidate $\boldsymbol{z}^*$ from proposal $q(\cdot|\boldsymbol{z}_{i-1})$
    Compute acceptance probability

$$\alpha(\boldsymbol{z}_{i-1}, \boldsymbol{z}^*) = \min\left\{1, \frac{q(\boldsymbol{z}_{i-1}|\boldsymbol{z}^*)L(\boldsymbol{y}_{\text{obs}}|\boldsymbol{z}^*)p_0(\boldsymbol{z}^*)}{q(\boldsymbol{z}^*|\boldsymbol{z}_{i-1})L(\boldsymbol{y}_{\text{obs}}|\boldsymbol{z}_{i-1})p_0(\boldsymbol{z}_{i-1})}\right\}$$

    Set the sample $\boldsymbol{z}_i$ to

$$\boldsymbol{z}_i = \begin{cases} \boldsymbol{z}^*\,, & \text{with probability } \alpha(\boldsymbol{z}_{i-1}, \boldsymbol{z}^*)\,, \\ \boldsymbol{z}_{i-1}\,, & \text{with probability } 1 - \alpha(\boldsymbol{z}_{i-1}, \boldsymbol{z}^*) \end{cases}$$

  end for
  return $\boldsymbol{z}_1, \dots, \boldsymbol{z}_m$
end procedure

6.1 Bayesian framework for inference

Consider the high-fidelity model f_{hi} that maps inputs $\boldsymbol{z} \in \mathcal{D}$ onto outputs $\boldsymbol{y} \in \mathcal{Y}$. Let p_0 be a prior density that describes the input $\boldsymbol{z}$ before any measurements. Let further $\boldsymbol{y}_{\text{obs}}$ be noisy observational data with the stochastic relationship

$$\boldsymbol{y}_{\text{obs}} = f_{\text{hi}}(\boldsymbol{z}) + \boldsymbol{\epsilon}\,,$$

where $\boldsymbol{\epsilon}$ is a random vector that captures the measurement error, noise, and other uncertainties of the observation $\boldsymbol{y}_{\text{obs}}$. In the following, the random vector $\boldsymbol{\epsilon}$ is modeled as a zero-mean Gaussian with covariance $\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}} \in \mathbb{R}^{d\times d}$. Define the data-misfit function as

$$\Phi(\boldsymbol{z}) = \frac{1}{2}\left\|\boldsymbol{\Sigma}_{\boldsymbol{\epsilon}}^{-\frac{1}{2}}\left(f_{\text{hi}}(\boldsymbol{z}) - \boldsymbol{y}_{\text{obs}}\right)\right\|^2\,,$$

with the norm $\|\cdot\|$. The likelihood function $L : \mathcal{D} \to \mathbb{R}$ is then proportional to

$$L(\boldsymbol{y}_{\text{obs}}|\boldsymbol{z}) \propto \exp(-\Phi(\boldsymbol{z}))\,. \tag{39}$$

Note that an evaluation of the likelihood L entails an evaluation of the high-fidelity model f_{hi}. The posterior probability density is

$$p(\boldsymbol{z}|\boldsymbol{y}_{\text{obs}}) \propto L(\boldsymbol{y}_{\text{obs}}|\boldsymbol{z})p_0(\boldsymbol{z})\,.$$

6.1.1 Exploring the posterior

The solution of the inference problem is explored by drawing samples from the posterior distribution. The posterior samples can then be used to estimate the input with the maximum posterior density and to estimate expectations of functions of interest $h : \mathcal{D} \to \mathbb{R}$ with respect to the posterior distribution. For example, one could be interested in the expected value of h over the posterior distribution

$$\mathbb{E}[h] = \int_{\mathcal{D}} h(\boldsymbol{z}) p(\boldsymbol{z}|\boldsymbol{y}_{\text{obs}}) \mathrm{d}\boldsymbol{z} \,. \tag{40}$$

MCMC methods are a popular way to sample from the posterior distribution, which have been studied extensively [178, 93, 137, 87, 58, 135, 62, 61]. One example of an MCMC method is the Metropolis-Hastings algorithm, which is an iterative scheme that draws candidate samples from a proposal distribution and then accepts the candidate sample with a probability that depends on the ratio of the posterior at the current candidate sample and the posterior at the sample of the previous iteration.

Metropolis-Hastings algorithm Algorithm 1 summarizes the Metropolis-Hastings algorithm. Inputs are the likelihood L, the prior density p_0, a proposal density q, and the number of samples m. The proposal density q is used to draw the next candidate sample. A typical choice for the proposal density is a Gaussian distribution that is centered at the sample of the previous iteration. In each iteration $i = 1, \ldots, m$, a candidate sample $\boldsymbol{z}^*$ is drawn from the proposal $q(\cdot|\boldsymbol{z}_{i-1})$ that depends on the sample $\boldsymbol{z}_{i-1}$ of the previous iteration. The candidate sample $\boldsymbol{z}^*$ is accepted $\boldsymbol{z}_i = \boldsymbol{z}^*$ with the probability $\alpha(\boldsymbol{z}^*, \boldsymbol{z}_{i-1})$, which depends on the ratio of the likelihood at the candidate sample $\boldsymbol{z}^*$ and the sample $\boldsymbol{z}_{i-1}$ of the previous iteration. If the candidate sample is rejected, then $\boldsymbol{z}_i = \boldsymbol{z}_{i-1}$. Algorithm 1 returns the samples $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_m$.

Metropolis-Hastings algorithm in practice Several techniques are necessary to make the Metropolis-Hastings algorithm practical. For example, the samples generated in the first iterations are usually discarded (burn-in) because they are strongly influenced by the initial starting point $\boldsymbol{z}_0$. We refer to the literature [177] for more details and further practical considerations.

6.1.2 Efficiency of MCMC sampling

Once samples $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_m$ are drawn with an MCMC algorithm, they can be used to, e.g., estimate the expectation (40) as

$$\bar{h} = \frac{1}{m} \sum_{i=1}^{m} h(\boldsymbol{z}_i) \,.$$

MCMC generates correlated samples $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_m$. The efficiency of MCMC sampling can therefore be measured by the effective sample size for a given computational budget

with respect to the estimator $\bar{h}$. To define the effective sample size, consider the samples $\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3, \ldots$ drawn with an MCMC algorithm and define the integrated autocorrelation time as

$$\tau_{\text{int}}(h) = \frac{1}{2} + \sum_{j=1}^{\infty} \rho_j \,,$$

where $\rho_j = \text{corr}(h(\boldsymbol{z}_1), h(\boldsymbol{z}_{j+1}))$ is the correlation between $h(\boldsymbol{z}_1)$ and $h(\boldsymbol{z}_{j+1})$. The effective sample size $m_{\text{eff}}(h)$ is

$$m_{\text{eff}}(h) = \frac{m}{2\tau_{\text{int}}(h)} \,,$$

such that

$$\mathbb{V}[\bar{h}] \approx \frac{\mathbb{V}[h]}{m_{\text{eff}}(h)} \,,$$

see [128, p. 125f] for a detailed derivation and further references. This means that there are two ways to improve the efficiency of sampling with MCMC [63]: (1) Increase the effective sample size for a given number of MCMC iterations with, e.g., adaptive MCMC [93, 179]. (2) Increase the number of MCMC iterations for a given computational budget, so that more samples can be generated for a given budget with, e.g., two-stage MCMC [51, 83].

6.2 Two-stage Markov chain Monte Carlo

Two-stage MCMC methods aim to increase the number of iterations for a given computational budget. In many applications, the Metropolis-Hastings algorithm, and MCMC in general, requires many iterations to produce an acceptable effective sample size. Each iteration means a likelihood evaluation, which means a high-fidelity model f_{hi} evaluation in the case of the likelihood L as defined in (39). The work [51, 83] proposes a two-stage delayed acceptance MCMC sampling. The candidate sample $\boldsymbol{z}^*$ has to be accepted with the likelihood induced by a low-fidelity model first, before $\boldsymbol{z}^*$ is passed on to be either accepted or rejected with the likelihood induced by the high-fidelity model.

Two-stage delayed acceptance MCMC algorithm The two-stage MCMC method is summarized in Algorithm 2. Inputs are the likelihood L^{hi} and L^{lo}, corresponding to the high- and low-fidelity model respectively, the prior density p_0, the proposal q, and the number of samples m. Consider one of the iterations $i = 1, \ldots, m$. The first stage of Algorithm 2 (lines 4–6) proceeds as the Metropolis-Hastings algorithm with a candidate sample $\boldsymbol{z}^{\text{lo}}$ drawn from the proposal distribution, except that the likelihood of the low-fidelity model is used, instead of the likelihood of the high-fidelity model. The result of the first stage is a sample $\boldsymbol{z}^{\text{hi}}$, which either is $\boldsymbol{z}^{\text{hi}} = \boldsymbol{z}^{\text{lo}}$ (accept) or $\boldsymbol{z}^{\text{hi}} = \boldsymbol{z}_{i-1}$ (reject). In the second stage of the algorithm (lines 7–9), the high-fidelity model is used to either accept or reject the candidate sample $\boldsymbol{z}^{\text{hi}}$ of the first stage. If the sample $\boldsymbol{z}^{\text{hi}}$ is accepted in the second stage, then the algorithm sets $\boldsymbol{z}_i = \boldsymbol{z}^{\text{hi}}$. If the sample $\boldsymbol{z}^{\text{hi}}$ is rejected in the second stage, then $\boldsymbol{z}_i = \boldsymbol{z}_{i-1}$. Note that in case the first stage rejected the sample

$\boldsymbol{z}^{\text{lo}}$, i.e., $\boldsymbol{z}^{\text{hi}} = \boldsymbol{z}_{i-1}$, no high-fidelity model evaluation is necessary in the second stage because the high-fidelity model output at $\boldsymbol{z}_{i-1}$ is available from the previous iteration. The proposal distribution Q in the second stage depends on the likelihood L^{lo} of the low-fidelity model. Note that $\delta_{\boldsymbol{z}_{i-1}}$ is the Dirac mass at $\boldsymbol{z}_{i-1}$. Algorithm 2 returns the samples $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_m$.

Efficiency of two-stage MCMC The key to the efficiency of Algorithm 2 is that no high-fidelity model evaluation is necessary in the second stage if the candidate sample $\boldsymbol{z}^{\text{lo}}$ was rejected at the first stage. Since the rejection rate of MCMC is typically high, many high-fidelity model evaluations are saved by the two-stage MCMC. We refer to [51] for an asymptotic analysis and the convergence properties of the two-stage MCMC. The two-stage MCMC uses a model management based on filtering because only candidate samples accepted with the low-fidelity model are passed on to the high-fidelity model, see Section 4.

Other multifidelity MCMC approaches In [72], two-stage MCMC is seen as a preconditioned Metropolis-Hastings algorithm. The low-fidelity model in [72] is a coarse-scale model of a high-fidelity multiscale finite volume model. We also mention [103], where multiple MCMC chains from coarse- (low-fidelity) and fine-scale (high-fidelity) models are coupled using a product chain. The work [68] couples multilevel Monte Carlo and MCMC to accelerate the estimation of expected values with respect to a posterior distribution. The low-fidelity models form a hierarchy of coarse-grid approximations of the high-fidelity model.

6.3 Markov chain Monte Carlo with adaptive low-fidelity models

In [63] an algorithm for combining high- and low-fidelity models in MCMC sampling is presented. The low-fidelity model is used in the first step of a two-stage MCMC approach to increase the acceptance rate of candidates in the second step, where the high-fidelity model is used to either accept or reject the sample. Additionally, the high-fidelity model outputs computed in the second step are used to adapt the low-fidelity model. In that sense, the approach uses a model management based on a combination of adaptation and filtering.

The low-fidelity model is a projection-based model in [63], see Section 3.2. The low-fidelity model is constructed in an offline phase from an initial reduced basis. At each MCMC iteration, the low-fidelity model is used in a first stage to generate a certain number of samples with the Metropolis-Hastings algorithm. The goal is to generate so many samples with the low-fidelity model that the initial sample and the last sample are uncorrelated. Then, in the second stage of the MCMC iteration, the last sample generated with the low-fidelity model is used as a candidate sample and the acceptance probability is computed using the high-fidelity model. Similarly to the two-stage MCMC methods discussed above, this algorithm aims to increase the acceptance probability at the second stage; however, the high-fidelity model output is used to improve the low-fidelity model after a sample has been accepted, and is not discarded. In that way,

the authors of [63] improve the low-fidelity model during the MCMC iterations, and consequently increase the acceptance probability in the second stage of the MCMC iterations. The resulting low-fidelity model is data-driven because it uses information provided by the observation $\boldsymbol{y}_{\text{obs}}$, rather than only prior information. Using samples from this adaptive two-stage MCMC approach yields unbiased Monte Carlo estimators, see the analysis in [63].

7 Multifidelity model management in optimization

The goal of optimization is to find an input that leads to an optimal model output with respect to a given objective function. Optimization is typically formulated as an iterative process that requires evaluating a model many times. Using only an expensive high-fidelity model is often computationally prohibitive. We discuss multifidelity methods that leverage low-fidelity models for speeding up the optimization while still resulting in a solution that satisfies the optimality conditions associated with the high-fidelity model. Section 7.1 formalizes the optimization task. Section 7.2 discusses global multifidelity optimization, which searches over the entire feasible domain, and Section 7.3 reviews local multifidelity optimization, which searches for a locally optimal solution, typically in a neighborhood of an initial input.

7.1 Optimization with high-fidelity models

We consider the setting of unconstrained optimization. Given is the high-fidelity model $f_{\text{hi}} : \mathcal{D} \to \mathcal{Y}$, with the d-dimensional input $\boldsymbol{z} \in \mathcal{D}$. The goal is to find an input $\boldsymbol{z}^* \in \mathcal{D}$ that solves

$$\min_{\boldsymbol{z} \in \mathcal{D}} f_{\text{hi}}(\boldsymbol{z}) \,. \tag{41}$$

Typically, the optimal input $\boldsymbol{z}^*$ is obtained in an iterative way, where a sequence of inputs $\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots$ is constructed such that this sequence $(\boldsymbol{z}_i)$ converges to the optimal input $\boldsymbol{z}^*$ in a certain sense [152].

Local and global optimization We distinguish between local and global optimization approaches. Global optimization searches over the entire feasible domain $\mathcal{D}$ for a minimizer $\boldsymbol{z}^*$, whereas local optimization terminates when a local optimum is found. Local optimization thus searches in a neighborhood of an initial point $\boldsymbol{z} \in \mathcal{D}$. Global methods typically do not require the gradient of f_{hi}, which may be advantageous in situations where the model is given as a black box and where approximations of the gradient contain significant noise. The use of gradient information in local methods leads to a more efficient search process that typically uses fewer model evaluations and that is more scalable to problems with high-dimensional input. There are also local methods that do not require gradient information.

We note that the multifidelity solution of constrained optimization problems can be achieved using penalty formulations that convert the constrained problem into an unconstrained one, although some multifidelity methods admit more sophisticated ways

to handle approximations of the constraints. Local methods can more easily deal with constraints, whereas global methods often fall back to heuristics (see, e.g., the discussion in [132]).

7.2 Global multifidelity optimization

A large class of global multifidelity optimization methods use adaptation as a model management strategy. These methods search for a minimizer with respect to an adaptively refined low-fidelity model f_{lo}. They guarantee that the resulting optimal solution is also a minimizer of the high-fidelity model f_{hi} by the way in which the low-fidelity model is adapted throughout the optimization search, using information from high-fidelity model evaluations. A critical task in this class of global multifidelity optimization methods therefore is to balance between exploitation (i.e., minimizing the current low-fidelity model) and exploration (i.e., adapting the low-fidelity model with information from the high-fidelity model).

7.2.1 Efficient global optimization (EGO)

EGO with expected improvement is frequently used to balance exploitation and exploration in cases where the low-fidelity model is a kriging model, see [109] and Section 3.3.1. Let $\boldsymbol{z}$ be the input that minimizes the low-fidelity model f_{lo} at the current iteration. Sampling at a new point $\boldsymbol{z}' \in \mathcal{D}$ means evaluating the high-fidelity model at $\boldsymbol{z}'$ and then adapting the low-fidelity model with information obtained from $f_{\text{hi}}(\boldsymbol{z}')$.

EGO provides a formulation for choosing the new sample point $\boldsymbol{z}'$. It does this by considering the value $f_{\text{hi}}(\boldsymbol{z}')$ of the high-fidelity model at the new sampling point $\boldsymbol{z}'$ to be uncertain before the high-fidelity model is evaluated at $\boldsymbol{z}'$. The uncertainty in $f_{\text{hi}}(\boldsymbol{z}')$ is modeled as a Gaussian random variable Y with mean and standard deviation given by the kriging (low-fidelity) model, see Section 3.3.1. The improvement at point $\boldsymbol{z}'$ is then given by $I(\boldsymbol{z}') = \max\{f_{\text{hi}}(\boldsymbol{z}) - Y, 0\}$, and the expected improvement is

$$\mathbb{E}[I(\boldsymbol{z}')] = \mathbb{E}[\max\{f_{\text{hi}}(\boldsymbol{z}) - Y, 0\}]\,.$$

The expected improvement at a point $\boldsymbol{z}'$ can be efficiently computed by exploiting the cheap computation of the MSE (and therefore standard deviation) of kriging models [82, 109]. The optimization process is then to start with an initial kriging model, find a new input that maximizes the expected improvement, evaluate the high-fidelity model at this new input, update the kriging model, and iterate. The optimization loop is typically stopped when the expected improvement is less than some small positive number [108]. We refer to [42] for a discussion on other stopping criteria.

7.2.2 Discussion of EGO

EGO can be made globally convergent and does not require high-fidelity model derivatives [108]. However, as discussed in [108, 132], EGO is sensitive to the initial points used for building the kriging model, which means that first a fairly exhaustive search

around the initial points might be necessary before a more global search begins. The reason for this behavior is that EGO proceeds in two steps. In the first step, a kriging model is learned, and in the second step, the kriging model is used as if it were correct to determine the next point. We refer to [108, 82] for details and possible improvements. The work in [171, 170], builds on EGO and develops a concept of expected improvement for multiobjective optimization that considers improvement and risk when selecting a new point. We also mention [41], where EGO is equipped with an adaptive target method that learns from previous iterations how much improvement to expect in the next iteration.

7.2.3 Other approaches to combine multifidelity models in the context of global optimization

In the work [5, 122], information from multiple kriging models is fused. Each model in the multifidelity hierarchy is approximated by a kriging model, and the random variables representing the kriging models are fused following the technique introduced in [206]. The fused model is a multifidelity model that is used in [122] for optimization. The authors of [88] propose to use a weighted average of an ensemble of low-fidelity models in an optimization framework. The weights are derived from the errors of the low-fidelity models.

There are multifidelity optimization methods based on pattern search [106, 196] that use filtering as model management strategy. For example, the multifidelity pattern search algorithm presented in [26] uses low-fidelity models to provide additional search directions, while preserving the convergence to a minimizer of the high-fidelity model. We also refer to [169] for a discussion of pattern search algorithms with low-fidelity models.

7.3 Local multifidelity optimization

Local multifidelity optimization methods in the literature typically use adaptation as a model management strategy. One class of approaches uses direct adaptation of a low-fidelity model, using high-fidelity evaluations that are computed as the optimization proceeds. Another class of approaches performs optimization on a corrected low-fidelity model, where the corrections are computed from high-fidelity model evaluations as the optimization proceeds. These latter approaches typically use a trust-region framework to manage the corrections.

7.3.1 Surrogate-based optimization

Surrogate-based optimization creates a low-fidelity model from data of the high-fidelity model and searches for a minimizer of the objective function induced by the low-fidelity model. Popular types of low-fidelity models in the context of surrogate-based optimization are kriging models, radial basis models, and polynomial regression models. Once a minimizer of the low-fidelity model has been found, the high-fidelity model is evaluated at the minimizer and the high-fidelity model output is then used to adapt the

low-fidelity model. This process is repeated until convergence to a minimizer that approximates well the minimizer of the high-fidelity model. Convergence to the minimizer of the high-fidelity model can be guaranteed in limited situations, see [18, 169, 82] for details.

7.3.2 Multifidelity trust-region methods

A classical way of exploiting a low-fidelity model in an optimization framework is to optimize over the low-fidelity model in a trust region—that is, to solve an optimization subproblem using the low-fidelity model, but to restrict the size of the step to lie within the trust region. A classical example is to derive a quadratic approximation of the high-fidelity model with its gradient and Hessian at the center of the trust region. The size of the trust region is then determined depending on the approximation quality of the quadratic approximation. We refer to [54] for an introduction, theory, and implementation of these classical trust region methods. The work [2, 3] established a multifidelity trust region framework for more general low-fidelity models. In particular, [2, 3] formulates a first-order consistency requirement on the low-fidelity model. The first-order consistency requirement is that the low- and the high-fidelity model have equal value and gradient at the center of the trust region. This consistency requirement ensures that the resulting multifidelity optimization algorithm converges to an optimal solution of the high-fidelity model.

Multifidelity trust-region algorithm The multifidelity trust-region approach of [2] is summarized in Algorithm 3. Inputs are the high-fidelity model f_{hi}, the parameters $\eta_1, \gamma_1, \eta_2, \gamma_2 > 0$ that control the expansion and contraction of the trust region, and an upper bound $\Delta^* > 0$ on the step size. In each iteration of the loop, an update to the current point is proposed and the trust-region is either expanded or contracted. Note that we have omitted stopping criteria in Algorithm 3 for the ease of exposition. In each iteration $i = 1, 2, 3, \ldots$, a low-fidelity model $f_{\text{lo}}^{(i)}$ is constructed that satisfies the first-order consistency requirement. Thus, the low-fidelity model output $f_{\text{lo}}^{(i)}(\boldsymbol{z}_i) = f_{\text{hi}}(\boldsymbol{z}_i)$ equals the high-fidelity model output at the current point $\boldsymbol{z}_i$, and the gradient of the low-fidelity model $\nabla f_{\text{lo}}^{(i)}(\boldsymbol{z}_i) = \nabla f_{\text{hi}}(\boldsymbol{z}_i)$ equals the gradient of the high-fidelity model. It is shown in [2] that with an additive or multiplicative correction an arbitrary low-fidelity model can be adjusted to satisfy the first-order consistency requirement. The first-order consistency requirement can also be established with the scaling approach introduced in [97]. Then, the step $\boldsymbol{s}_i$ is computed and accepted if the point $\boldsymbol{z}_i + \boldsymbol{s}_i$ decreases the high-fidelity model f_{hi} with respect to the current point $\boldsymbol{z}_i$, i.e., if $f_{\text{hi}}(\boldsymbol{z}_i + \boldsymbol{s}_i) < f_{\text{hi}}(\boldsymbol{z}_i)$. The size of the trust region is updated depending on the ratio of the actual decrease $f_{\text{hi}}(\boldsymbol{z}_i) - f_{\text{hi}}(\boldsymbol{z}_i + \boldsymbol{s}_i)$ to the estimated decrease $f_{\text{lo}}^{(i)}(\boldsymbol{z}_i) - f_{\text{lo}}^{(i)}(\boldsymbol{z}_i + \boldsymbol{s}_i)$. The parameters γ_1 and γ_2 control when to shrink and when to expand the trust region, respectively. The parameters η_1 and η_2 define the size of the contracted and expanded trust region, respectively. The algorithm returns the points $\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3, \ldots$.

Multifidelity trust-region with adaptive models In Algorithm 3, the low-fidelity model is corrected in each iteration. In [11, 76], the trust-region POD (TRPOD) is proposed, which uses a projection-based POD reduced model in conjunction with a trust-region optimization framework and adapts the POD reduced model in each optimization iteration. Similarly, in [22, 214], projection-based models are adapted within similar trust region frameworks. The work [213] uses error bounds for interpolatory reduced models to define the trust region and refinement of the reduced models to guarantee convergence to the optimality conditions associated with the high-fidelity model.

Multifidelity trust-region without gradients If gradients are unavailable, or too costly to approximate, then the framework developed by [2, 3] relying on the first-order consistency of the low-fidelity model cannot be applied. In [133], a gradient-free multifidelity trust-region framework with radial basis function interpolants as low-fidelity models is introduced, building on the gradient-free trust-region methods of [55, 56] and the tailored radial basis function modeling of [205]. This leads to an error interpolation that makes the low-fidelity model satisfy the sufficient condition to prove convergence to a minimizer of the high-fidelity model.

7.4 Multifidelity optimization under uncertainty

In their simplest forms, optimization under uncertainty formulations consider a similar problem to (41), but where now the objective function f_{hi} incorporates one or more statistics that in turn depend on underlying uncertainty in the problem. For example, a common objective function is a weighted sum of expected performance and performance standard deviation. Thus, each optimization iteration embeds an uncertainty quantification loop (e.g., Monte Carlo sampling or stochastic collocation) over the uncertain parameters. [148] uses the control variate-based multifidelity Monte Carlo method of Section 5.2 to derive a multifidelity optimization under uncertainty method that provides substantial reductions in computational cost using a variety of low-fidelity model types. In [149], it is shown that evaluations computed at previous optimization iterations form an effective and readily-available low-fidelity model that can be exploited by this control-variate formulation in the optimization under uncertainty setting. The work in [189] uses a combination of polynomial chaos stochastic expansions and corrections based on coarse-grid approximations to formulate a multifidelity robust optimization approach.

We highlight optimization under uncertainty as an important target area for future multifidelity methods. It is a computationally demanding process, but one with critical importance to many areas, such as engineering design.

8 Conclusions and outlook

As can be seen from the vast literature surveyed in this paper, multifidelity methods have begun to have impact across diverse outer-loop applications in computational science and engineering. Multifidelity methods have been used for more than two decades

to accelerate solution of optimization problems. Their application in uncertainty quantification is more recent, but appears to offer even more opportunity for computational speedups, due to the heavy computational burden typically associated with uncertainty quantification tasks such as Monte Carlo and MCMC sampling.

This paper highlights the broad range of multifidelity approaches that exist in the literature. These approaches span many types of low-fidelity models as well as many specific strategies for achieving the multifidelity model management. We attempt to bring some perspective on the similarities and differences across methods, as well as their relative advantages, by categorizing methods that share a common theoretical foundation. We discuss methods to create low-fidelity models according to the three areas of simplified models, projection-based models, and data-fit models. We categorize multifidelity model management methods as being based on adaptation, fusion, and filtering. In most settings, one can flexibly choose a combination of model management strategy and low-fidelity model type, although—as is always the case in computational modeling—bringing to bear knowledge of the problem structure helps to make effective decisions on the low-fidelity modeling and multifidelity model management strategies that are best suited to the problem at hand. We note that this paper focused on outer-loop applications and therefore mentioned only briefly multifidelity methods that do not directly exploit the outer-loop setting.

Multifidelity methods have advanced considerably, especially in the past decade. Yet a number of important challenges remain. In almost all existing multifidelity methods (and in the presentation of material in this paper), the assumption is that the high-fidelity model represents some "truth." This ignores the important fact that the output of the high-fidelity model is itself an approximation of reality. Even in the absence of uncertainties in the inputs, all models, including the high-fidelity model, are inadequate [116]. Furthermore, the relationships among different models may be much richer than the linear hierarchy assumed by existing multifidelity methods. One way multifidelity methods can account for model inadequacy is by fusing outputs from multiple models. Model inadequacy is often quantified by probabilities, which describe the belief that a model yields the true output. Techniques for assigning model probabilities reach from expert opinion [215, 176] to statistical information criteria [127, 36] to quantifying the difference between experimental data and model outputs [156]. The probabilities are then used to fuse the model outputs with, e.g., Bayesian model averaging [124] or adjustment factors [176]. The approach presented in [5] assigns a probability distribution to the output of each model and employs Bayesian estimation to fuse the outputs together. Incorporating such approaches into multifidelity model management strategies for outer loop applications remains an important open research challenge.

Another important challenge, already discussed briefly in this paper, is to move beyond methods that focus exclusively on models, so that decision-makers can draw on a broader range of available information sources. Again, some of the foundations exist in the statistical literature, such as [115] which derives models by fusing multiple information sources such as experiments, expert opinions, lookup tables, and computational models. In drawing on various information sources, multifidelity model management strategies must be expanded to address not just which information source to evaluate

when, but also *where* (i.e., at what inputs) to evaluate the information source. Relevant foundations to address this challenge include the experimental design literature and value of information analysis [167].

Acknowledgements

The first two authors acknowledge support of the AFOSR MURI on multi-information sources of multi-physics systems under Award Number FA9550-15-1-0038, the United States Department of Energy Applied Mathematics Program, Awards DE-FG02-08ER2585 and DE-SC0009297, as part of the DiaMonD Multifaceted Mathematics Integrated Capability Center, DARPA EQUiPS Award UTA15-001067, and the MIT-SUTD International Design Center. The third author was supported by the US Department of Energy Office of Science grant DE-SC0009324.

References

[1] T. Akhtar and C. A. Shoemaker. Multi objective optimization of computationally expensive multi-modal functions with RBF surrogates and multi-rule selection. *Journal of Global Optimization*, 64(1):17–32, 2016.

[2] N. M. Alexandrov, J. E. D. Jr, R. M. Lewis, and V. Torczon. A trust-region framework for managing the use of approximation models in optimization. *Structural optimization*, 15(1):16–23, 1998.

[3] N. M. Alexandrov, R. Lewis, C. R. Gumbert, L. L. Green, and P. A. Newman. Optimization with variable-fidelity models applied to wing design. Technical Report CR-1999-209826, NASA, 1999.

[4] N. M. Alexandrov, R. M. Lewis, C. R. Gumbert, L. L. Green, and P. A. Newman. Approximation and model management in aerodynamic optimization with variable-fidelity models. *Journal of Aircraft*, 38(6):1093–1101, 2001.

[5] D. Allaire and K. Willcox. A mathematical and computational framework for multifidelity design and analysis with computer models. *International Journal for Uncertainty Quantification*, 4(1):1–20, 2014.

[6] A. Ammar, F. Chinesta, P. Diez, and A. Huerta. An error estimator for separated representations of highly multidimensional models. *Computer Methods in Applied Mechanics and Engineering*, 199(25–28):1872 – 1880, 2010.

[7] D. Amsallem and C. Farhat. An online method for interpolating linear parametric reduced-order models. *SIAM Journal on Scientific Computing*, 33(5):2169–2198, 2011.

[8] D. Amsallem, M. J. Zahr, and K. Washabaugh. Fast local reduced basis updates for the efficient reduction of nonlinear systems with hyper-reduction. *Advances in Computational Mathematics*, 41(5):1187–1230, 2015.

[9] A. E. Annels. Geostatistical Ore-reserve Estimation. In *Mineral Deposit Evaluation*, pages 175–245. Springer Netherlands, 1991.

[10] A. C. Antoulas. *Approximation of Large-scale Dynamical Systems*. SIAM, 2005.

[11] E. Arian, M. Fahl, and E. Sachs. Trust-region proper orthogonal decomposition models by optimization methods. In *Proceedings of the 41st IEEE Conference on Decision and Control*, pages 3300–3305, Las Vegas, NV, 2002.

[12] P. Astrid, S. Weiland, K. Willcox, and T. Backx. Missing point estimation in models described by proper orthogonal decomposition. *IEEE Transactions on Automatic Control*, 53(10):2237–2251, 2008.

[13] I. Babuška, F. Nobile, and R. Tempone. Worst case scenario analysis for elliptic problems with uncertainty. *Numerische Mathematik*, 101(2):185–219, 2005.

[14] I. Babuška, F. Nobile, and R. Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 45(3):1005–1034, 2007.

[15] Z. Bai. Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems. *Applied Numerical Mathematics*, 43(1–2):9–44, 2002.

[16] R. E. Bank, T. F. Dupont, and H. Yserentant. The hierarchical basis multigrid method. *Numerische Mathematik*, 52(4):427–458, 1988.

[17] M. Barrault, Y. Maday, N.-C. Nguyen, and A. Patera. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique*, 339(9):667–672, 2004.

[18] J. F. M. Barthelemy and R. T. Haftka. Approximation concepts for optimum structural design — a review. *Structural optimization*, 5(3):129–144, 1993.

[19] A. Basudhar and S. Missoum. An improved adaptive sampling scheme for the construction of explicit boundaries. *Structural and Multidisciplinary Optimization*, 42(4):517–529, 2010.

[20] U. Baur, C. Beattie, P. Benner, and S. Gugercin. Interpolatory projection methods for parameterized model reduction. *SIAM Journal on Scientific Computing*, 33(5):2489–2518, 2011.

[21] P. Benner, S. Gugercin, and K. Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, 57(4):483–531, 2015.

[22] M. Bergmann and L. Cordier. Optimal control of the cylinder wake in the laminar regime by trust-region methods and POD reduced-order models. *Journal of Computational Physics*, 227(16):7813 – 7840, 2008.

[23] G. Berkooz, P. Holmes, and J. L. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review of Fluid Mechanics*, 25(1):539–575, 1993.

[24] A. Bernardini. What are the Random and Fuzzy Sets and How to Use Them for Uncertainty Modelling in Engineering Systems? In I. Elishakoff, editor, *Whys and Hows in Uncertainty Modelling*, number 388 in CISM Courses and Lectures, pages 63–125. Springer Vienna, 1999.

[25] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[26] A. J. Booker, J. E. Dennis, P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset. A rigorous framework for optimization of expensive functions by surrogates. *Structural optimization*, 17(1):1–13, 1999.

[27] S. Boyaval. A fast Monte–Carlo method with a reduced basis of control variates applied to uncertainty propagation and Bayesian estimation. *Computer Methods in Applied Mechanics and Engineering*, 241–244:190–205, 2012.

[28] S. Boyaval and T. Lelièvre. A variance reduction method for parametrized stochastic differential equations using the reduced basis paradigm. *Communications in Mathematical Sciences*, 8(3):735–762, 2010.

[29] J. H. Bramble, J. E. Pasciak, and J. Xu. Parallel multilevel preconditioners. *Mathematics of Computation*, 55(191):1–22, 1990.

[30] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31:333–390, 1977.

[31] P. Bratley, B. L. Fox, and L. E. Schrage. *A Guide to Simulation*. Springer, 1987.

[32] W. Briggs, V. E. Henson, and S. F. McCormick. *A Multigrid Tutorial*. SIAM, 2000.

[33] T. Bui-Thanh, K. Willcox, and O. Ghattas. Model reduction for large-scale systems with high-dimensional parametric input space. *SIAM Journal on Scientific Computing*, 30(6):3270–3288, 2008.

[34] H.-J. Bungartz and M. Griebel. Sparse grids. *Acta Numerica*, 13:147–269, 2004.

[35] J. Burkardt, M. Gunzburger, and H.-C. Lee. POD and CVT-based reduced-order modeling of Navier-Stokes flows. *Computer Methods in Applied Mechanics and Engineering*, 196(1–3):337 – 355, 2006.

[36] K. Burnham and D. Anderson. *Model Selection and and Multi-model Inference: A Practical Guide Information-theoretic Approach*. Springer, 2002.

[37] K. Carlberg. Adaptive h-refinement for reduced-order models. *International Journal for Numerical Methods in Engineering*, 102(5):1192–1210, 2015.

[38] K. Carlberg, C. Bou-Mosleh, and C. Farhat. Efficient non-linear model reduction via a least-squares Petrov-Galerkin projection and compressive tensor approximations. *International Journal for Numerical Methods in Engineering*, 86(2):155–181, 2011.

[39] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[40] S. Chaturantabut and D. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.

[41] A. Chaudhuri and R. T. Haftka. Efficient global optimization with adaptive target setting. *AIAA Journal*, 52(7):1573–1578, 2014.

[42] A. Chaudhuri and R. T. Haftka. Effectiveness indicators for stopping criteria based on minimum required improvement. In *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA SciTech. American Institute of Aeronautics and Astronautics, 2015.

[43] P. Chen and A. Quarteroni. Accurate and efficient evaluation of failure probability for partial differential equations with random input data. *Computer Methods in Applied Mechanics and Engineering*, 267:233–260, 2013.

[44] P. Chen, A. Quarteroni, and G. Rozza. A weighted empirical interpolation method: a priori convergence analysis and applications. *ESAIM: Mathematical Modelling and Numerical Analysis*, 48:943–953, 2014.

[45] F. Chinesta, A. Ammar, and E. Cueto. Proper generalized decomposition of multiscale models. *International Journal for Numerical Methods in Engineering*, 83(8-9):1114–1132, 2010.

[46] F. Chinesta, A. Ammar, and E. Cueto. Recent advances and new challenges in the use of the proper generalized decomposition for solving multidimensional models. *Archives of Computational Methods in Engineering*, 17(4):327–350, 2010.

[47] F. Chinesta, A. Ammar, A. Falcó, and M. Laso. On the reduction of stochastic kinetic theory models of complex fluids. *Modelling and Simulation in Materials Science and Engineering*, 15(6):639, 2007.

[48] F. Chinesta, A. Ammar, F. Lemarchand, P. Beauchene, and F. Boust. Alleviating mesh constraints: Model reduction, parallel time integration and high resolution homogenization. *Computer Methods in Applied Mechanics and Engineering*, 197(5):400 – 413, 2008.

[49] F. Chinesta, P. Ladevèze, and E. Cueto. A short review on model order reduction based on proper generalized decomposition. *Archives of Computational Methods in Engineering*, 18(4):395–404, 2011.

[50] F. Chinesta, A. Leygue, F. Bordeu, J. V. Aguado, E. Cueto, D. Gonzalez, I. Alfaro, A. Ammar, and A. Huerta. PGD-based computational vademecum for efficient design, optimization and control. *Archives of Computational Methods in Engineering*, 20(1):31–59, 2013.

[51] J. A. Christen and C. Fox. Markov chain Monte Carlo using an approximation. *Journal of Computational and Graphical Statistics*, 14(4):795–810, 2005.

[52] H. S. Chung and J. Alonso. Design of a low-boom supersonic business jet using cokriging approximation models. In *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. American Institute of Aeronautics and Astronautics, 2002.

[53] K. A. Cliffe, M. B. Giles, R. Scheichl, and A. L. Teckentrup. Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients. *Computing and Visualization in Science*, 14(1):3–15, 2011.

[54] A. Conn, N. Gould, and P. Toint. *Trust Region Methods*. MOS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, 2000.

[55] A. Conn, K. Scheinberg, and L. Vicente. Global convergence of general derivative-free trust-region algorithms to first- and second-order critical points. *SIAM Journal of Optimization*, 20(1):387–415, 2009.

[56] A. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to Derivative-Free Optimization*. MPS/SIAM Series on Optimization. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2009.

[57] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[58] S. L. Cotter, G. O. Roberts, A. M. Stuart, and D. White. MCMC methods for functions: Modifying old algorithms to make them faster. *Statist. Sci.*, 28(3):424–446, 08 2013.

[59] I. Couckuyt, A. Forrester, D. Gorissen, F. De Turck, and T. Dhaene. Blind Kriging: Implementation and performance analysis. *Advances in Engineering Software*, 49:1–13, 2012.

[60] T. Cui, C. Fox, and M. J. O'Sullivan. Bayesian calibration of a large-scale geothermal reservoir model by a new adaptive delayed acceptance Metropolis Hastings algorithm. *Water Resources Research*, 47(10):1–26, 2011.

[61] T. Cui, K. J. Law, and Y. M. Marzouk. Dimension-independent likelihood-informed MCMC. *Journal of Computational Physics*, 304:109 – 137, 2016.

[62] T. Cui, J. Martin, Y. M. Marzouk, A. Solonen, and A. Spantini. Likelihood-informed dimension reduction for nonlinear inverse problems. *Inverse Problems*, 30(11):114015, 2014.

[63] T. Cui, Y. M. Marzouk, and K. Willcox. Data-driven model reduction for the Bayesian solution of inverse problems. *International Journal for Numerical Methods in Engineering*, 102(5):966–990, 2015.

[64] G. Dahlquist, Å. Björck, and N. Anderson. *Numerical Methods*. Dover, 1974.

[65] W. Dahmen and A. Kunoth. Multilevel preconditioning. *Numerische Mathematik*, 63(1):315–344, 1992.

[66] I. Daubechies. *Ten Lectures on Wavelets*. SIAM, 1992.

[67] J. Degroote, J. Vierendeels, and K. Willcox. Interpolation among reduced-order matrices to obtain parameterized models for design, optimization and probabilistic analysis. *International Journal for Numerical Methods in Fluids*, 63(2):207–230, 2010.

[68] T. J. Dodwell, C. Ketelsen, R. Scheichl, and A. L. Teckentrup. A hierarchical multilevel Markov chain Monte Carlo algorithm with applications to uncertainty quantification in subsurface flow. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):1075–1108, 2015.

[69] M. Drohmann, B. Haasdonk, and M. Ohlberger. Reduced basis approximation for nonlinear parametrized evolution equations based on empirical operator interpolation. *SIAM Journal on Scientific Computing*, 34(2):A937–A969, 2012.

[70] Q. Du, V. Faber, and M. Gunzburger. Centroidal voronoi tessellations: Applications and algorithms. *SIAM Review*, 41(4):637–676, 1999.

[71] Q. Du and M. D. Gunzburger. Centroidal voronoi tessellation based proper orthogonal decomposition analysis. In W. Desch, F. Kappel, and K. Kunisch, editors, *Control and Estimation of Distributed Parameter Systems: International Conference in Maria Trost (Austria), July 15–21, 2001*, pages 137–150, Basel, 2003. Birkhäuser Basel.

[72] Y. Efendiev, T. Hou, and W. Luo. Preconditioning Markov chain Monte Carlo simulations using coarse-scale models. *SIAM Journal on Scientific Computing*, 28(2):776–803, 2006.

[73] J. L. Eftang and B. Stamm. Parameter multi-domain 'hp' empirical interpolation. *International Journal for Numerical Methods in Engineering*, 90(4):412–428, 2012.

[74] M. Eldred, A. Giunta, and S. Collis. Second-order corrections for surrogate-based optimization with model hierarchies. In *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Multidisciplinary Analysis Optimization Conferences. American Institute of Aeronautics and Astronautics, 2004.

[75] M. Emsermann and B. Simon. Improving simulation efficiency with quasi control variates. *Stochastic Models*, 18(3):425–448, 2002.

[76] M. Fahl and E. Sachs. Reduced order modelling approaches to PDE-constrained optimization based on proper orthogonal decomposition. In L. Biegler et al., editor, *Large-Scale PDE-Constrained Optimization*, pages 268–280. Springer, 2003.

[77] A. Falcó and A. Nouy. Proper generalized decomposition for nonlinear convex problems in tensor banach spaces. *Numerische Mathematik*, 121(3):503–530, 2012.

[78] P. Feldmann and R. Freund. Efficient linear circuit analysis by Padé approximation via the Lanczos process. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 14(5):639–649, 1995.

[79] G. Fishman. *Monte Carlo*. Springer, 1996.

[80] A. I. J. Forrester and A. J. Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1–3):50–79, 2009.

[81] A. I. J. Forrester, A. Sobester, and A. J. Keane. Multi-fidelity optimization via surrogate modelling. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 463(2088):3251–3269, 2007.

[82] A. I. J. Forrester, A. Sobester, and A. J. Keane. *Engineering design via surrogate modelling: a practical guide*. Wiley, 2008.

[83] C. Fox and G. Nicholls. Sampling conductivity images via MCMC. In *The Art and Science of Bayesian Image Analysis*, pages 91–100. University of Leeds, 1997.

[84] R. W. Freund. Model reduction methods based on Krylov subspaces. *Acta Numerica*, 12:267–319, 2003.

[85] K. Gallivan, E. Grimme, and P. Van Dooren. Padé Approximation of Large-Scale Dynamic Systems with Lanczos Methods. Proceedings of the 33rd IEEE Conference on Decision and Control, 1994.

[86] M. Giles. Multi-level Monte Carlo path simulation. *Operations Research*, 56(3):607–617, 2008.

[87] M. Girolami and B. Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.

[88] T. Goel, R. T. Haftka, W. Shyy, and N. V. Queipo. Ensemble of surrogates. *Structural and Multidisciplinary Optimization*, 33(3):199–216, 2007.

[89] M. A. Grepl, Y. Maday, N. C. Nguyen, and A. T. Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 41(03):575–605, 2007.

[90] S. Gugercin, A. Antoulas, and C. Beattie. $\mathcal{H}_2$ model reduction for large-scale linear dynamical systems. *SIAM Journal on Matrix Analysis and Applications*, 30(2):609–638, 2008.

[91] S. Gugercin and A. C. Antoulas. A survey of model reduction by balanced truncation and some new results. *International Journal of Control*, 77(8):748–766, 2004.

[92] M. D. Gunzburger, C. G. Webster, and G. Zhang. Stochastic finite element methods for partial differential equations with random input data. *Acta Numerica*, 23:521–650, 2014.

[93] H. Haario, E. Saksman, and J. Tamminen. An adaptive Metropolis algorithm. *Bernoulli*, 7(2):223–242, 2001.

[94] B. Haasdonk. Convergence rates of the POD-Greedy method. *ESAIM: Mathematical Modelling and Numerical Analysis*, 47:859–873, 2013.

[95] B. Haasdonk and M. Ohlberger. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM: M2AN*, 42(2):277–302, 2008.

[96] W. Hackbusch. *Multi-Grid Methods and Applications*. Springer, 1985.

[97] R. T. Haftka. Combining global and local approximations. *AIAA Journal*, 29(9):1523–1525, 1991.

[98] A.-L. Haji-Ali, F. Nobile, and R. Tempone. Multi-index Monte Carlo: when sparsity meets sampling. *Numerische Mathematik*, 132(4):767–806, 2015.

[99] J. M. Hammersley and D. C. Handscomb. *Monte Carlo Methods*. Methuen London, 1964.

[100] Z.-H. Han, S. Görtz, and R. Zimmermann. Improving variable-fidelity surrogate modeling via gradient-enhanced kriging and a generalized hybrid bridge function. *Aerospace Science and Technology*, 25(1):177 – 189, 2013.

[101] S. Heinrich. Multilevel Monte Carlo methods. In S. Margenov, J. Waśniewski, and P. Yalamov, editors, *Large-Scale Scientific Computing*, number 2179 in Lecture Notes in Computer Science, pages 58–67. Springer Berlin Heidelberg, 2001.

[102] I. M. Held. The gap between simulation and understanding in climate modeling. *Bulletin of the American Meteorological Society*, 86(11):1609–1614, 2005.

[103] D. Higdon, H. Lee, and Z. Bi. A Bayesian approach to characterizing uncertainty in inverse problems using coarse and fine-scale information. *IEEE Transactions on Signal Processing*, 50(2):389–399, 2002.

[104] M. Hinze and S. Volkwein. Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: Error estimates and suboptimal control. In P. Benner, V. Mehrmann, and D. Sorensen, editors, *Dimension Reduction of Large-Scale Systems*, volume 45 of *Lecture Notes in Computational and Applied Mathematics*, pages 261–306, 2005.

[105] M. Hinze and S. Volkwein. Error estimates for abstract linear-quadratic optimal control problems using proper orthogonal decomposition. *Computational Optimization and Applications*, 39(3):319–345, 2008.

[106] R. Hooke and T. A. Jeeves. "Direct search" solution of numerical and statistical problems. *Journal of the ACM*, 8(2):212–229, 1961.

[107] D. Huynh, G. Rozza, S. Sen, and A. Patera. A successive constraint linear optimization method for lower bounds of parametric coercivity and inf–sup stability constants. *Comptes Rendus Mathematique*, 345(8):473 – 478, 2007.

[108] D. R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383, 2001.

[109] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

[110] V. R. Joseph. A Bayesian approach to the design and analysis of fractionated experiments. *Technometrics*, 48(2):219–229, 2006.

[111] V. R. Joseph, Y. Hung, and A. Sudjianto. Blind Kriging: A new method for developing metamodels. *Journal of Mechanical Design*, 130(3):031102–031102, 2008.

[112] A. G. Journel and M. E. Rossi. When do we need a trend model in kriging? *Mathematical Geology*, 21(7):715–739, 1989.

[113] J. Kaipo and E. Somersalo. *Statistical and computational inverse problems*. Springer, 2005.

[114] A. J. Keane. Wing optimization using design of experiment, response surface, and data fusion methods. *Journal of Aircraft*, 40(4):741–750, 2003.

[115] M. C. Kennedy and A. O'Hagan. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 2000.

[116] M. C. Kennedy and A. O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001.

[117] D. E. Keyes. Exaflop/s: The why and the how. *Comptes Rendus Mécanique*, 339(2–3):70–77, 2011.

[118] P. Koutsourelakis. Accurate uncertainty quantification using inaccurate computational models. *SIAM Journal on Scientific Computing*, 31(5):3274–3300, 2009.

[119] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for parabolic problems. *Numerische Mathematik*, 90(1):117–148, 2001.

[120] K. Kunisch and S. Volkwein. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM Journal on Numerical Analysis*, 40(2):492–515, 2002.

[121] P. Ladevèze. *Nonlinear Computational Structural Mechanics*. Springer, 1999.

[122] R. Lam, D. Allaire, and K. Willcox. Multifidelity optimization using statistical surrogate modeling for non-hierarchical information sources. In *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. American Institute of Aeronautics and Astronautics, 2015.

[123] J. Laurenceau and P. Sagaut. Building efficient response surfaces of aerodynamic functions with kriging and cokriging. *AIAA Journal*, 46(2):498–507, 2008.

[124] E. Leamer. *Specification Searches: Ad Hoc Inference with Nonexperimental Data*. John Wiley & Sons, 1978.

[125] J. Li, J. Li, and D. Xiu. An efficient surrogate-based method for computing rare failure probability. *Journal of Computational Physics*, 230(24):8683–8697, 2011.

[126] J. Li and D. Xiu. Evaluation of failure probability via surrogate models. *Journal of Computational Physics*, 229(23):8966–8980, 2010.

[127] W. A. Link and R. J. Barker. Model weights and the foundations of multimodel inference. *Ecology*, 87(10):2626–2635, 2006.

[128] J. S. Liu. *Monte Carlo strategies in scientific computing*. Springer, 2008.

[129] R. Lucas, J. Ang, K. Bergman, S. Borkar, W. Carlson, L. Carrington, G. Chiu, R. Colwell, W. Dally, J. Dongarra, A. Geist, G. Grider, R. Haring, J. Hittinger, A. Hoisie, D. Klein, P. Kogge, R. Lethin, V. Sarkar, R. Schreiber, J. Shalf, T. Sterling, and R. Stevens. Top ten exascale research challenges. Technical report, U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, 2014.

[130] O. P. L. Maître and O. M. Knio. *Spectral Methods for Uncertainty Quantification*. Springer, 2010.

[131] A. J. Majda and B. Gershgorin. Quantifying uncertainty in climate change science through empirical information theory. *Proceedings of the National Academy of Sciences of the United States of America*, 107(34):14958–14963, 2010.

[132] A. March. *Multifidelity methods for multidisciplinary system design*. Phd thesis, Massachusetts Institute of Technology, June 2012.

[133] A. March and K. Willcox. Constrained multifidelity optimization using model calibration. *Structural and Multidisciplinary Optimization*, 46(1):93–109, 2012.

[134] A. March and K. Willcox. Provably convergent multifidelity optimization algorithm not requiring high-fidelity derivatives. *AIAA Journal*, 50(5):1079–1089, 2012.

[135] J. Martin, L. C. Wilcox, C. Burstedde, and O. Ghattas. A stochastic Newton MCMC method for large-scale statistical inverse problems with application to seismic inversion. *SIAM Journal on Scientific Computing*, 34(3):A1460–A1487, 2012.

[136] J. D. Martin and T. W. Simpson. Use of Kriging models to approximate deterministic computer models. *AIAA Journal*, 43(4):853–863, 2005.

[137] Y. Marzouk and H. N. Najm. Dimensionality reduction and polynomial chaos acceleration of Bayesian inference in inverse problems. *Journal of Computational Physics*, 228(6):1862 – 1902, 2009.

[138] Y. Marzouk and K. Willcox. Uncertainty quantification. In N. Higham, editor, *The Princeton Companion to Applied Mathematics*. Princeton University Press, 2015.

[139] Y. Marzouk and D. Xiu. A stochastic collocation approach to Bayesian inference in inverse problems. *Communications in Computational Physics*, 6:826–847, 2009.

[140] G. Matheron. Principles of geostatistics. *Economic Geology*, 58(8):1246–1266, 1963.

[141] P. Mills. Efficient statistical classification of satellite measurements. *International Journal of Remote Sensing*, 32(21):6109–6132, 2011.

[142] B. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, 26(1):17–32, 1981.

[143] C. Mullis and R. Roberts. Synthesis of minimum roundoff noise fixed point digital filters. *IEEE Transactions on Circuits and Systems*, 23(9):551–562, 1976.

[144] D. E. Myers. Matrix formulation of co-kriging. *Journal of the International Association for Mathematical Geology*, 14(3):249–257, 1982.

[145] A. Narayan, C. Gittelson, and D. Xiu. A stochastic collocation algorithm with multifidelity models. *SIAM Journal on Scientific Computing*, 36(2):A495–A521, 2014.

[146] B. L. Nelson. On control variate estimators. *Computers & Operations Research*, 14(3):219–225, 1987.

[147] L. W. Ng and M. Eldred. Multifidelity uncertainty quantification using non-intrusive polynomial chaos and stochastic collocation. In *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Structures, Structural Dynamics, and Materials and Co-located Conferences. American Institute of Aeronautics and Astronautics, 2012.

[148] L. W. Ng and K. Willcox. Multifidelity approaches for optimization under uncertainty. *International Journal for Numerical Methods in Engineering*, 100(10):746–772, 2014.

[149] L. W. Ng and K. Willcox. Monte-Carlo information-reuse approach to aircraft conceptual design optimization under uncertainty. *Journal of Aircraft*, 53(2):427–438, 2016.

[150] F. Nobile, R. Tempone, and C. G. Webster. A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 46(5):2309–2345, 2008.

[151] F. Nobile and F. Tesei. A multi level Monte Carlo method with control variate for elliptic PDEs with log-normal coefficients. *Stochastic Partial Differential Equations: Analysis and Computations*, 3(3):398–444, 2015.

[152] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, 2006.

[153] A. Nouy. A generalized spectral decomposition technique to solve a class of linear stochastic partial differential equations. *Computer Methods in Applied Mechanics and Engineering*, 196(45–48):4521 – 4537, 2007.

[154] A. Nouy. Proper generalized decompositions and separated representations for the numerical solution of high dimensional stochastic problems. *Archives of Computational Methods in Engineering*, 17(4):403–434, 2010.

[155] H. Panzer, J. Mohring, R. Eid, and B. Lohmann. Parametric model order reduction by matrix interpolation. *at – Automatisierungstechnik*, 58(8):475–484, 2010.

[156] I. Park, H. K. Amarchinta, and R. V. Grandhi. A Bayesian approach for quantification of model uncertainty. *Reliability Engineering & System Safety*, 95(7):777–785, 2010.

[157] R. Pasupathy, B. W. Schmeiser, M. R. Taaffe, and J. Wang. Control-variate estimation using estimated control means. *IIE Transactions*, 44(5):381–385, 2012.

[158] B. Peherstorfer, D. Butnaru, K. Willcox, and H.-J. Bungartz. Localized discrete empirical interpolation method. *SIAM Journal on Scientific Computing*, 36(1):A168–A192, 2014.

[159] B. Peherstorfer, T. Cui, Y. Marzouk, and K. Willcox. Multifidelity importance sampling. *Computer Methods in Applied Mechanics and Engineering*, 300:490 – 509, 2016.

[160] B. Peherstorfer, D. Pflüger, and H.-J. Bungartz. Density estimation with adaptive sparse grids for large data sets. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 443–451. SIAM, 2014.

[161] B. Peherstorfer and K. Willcox. Dynamic data-driven reduced-order models. *Computer Methods in Applied Mechanics and Engineering*, 291:21–41, 2015.

[162] B. Peherstorfer and K. Willcox. Online adaptive model reduction for nonlinear systems via low-rank updates. *SIAM Journal on Scientific Computing*, 37(4):A2123–A2150, 2015.

[163] B. Peherstorfer, K. Willcox, and M. Gunzburger. Optimal model management for multifidelity Monte Carlo estimation. Technical Report 15-2, Aerospace Computational Design Laboratory, MIT, 2015.

[164] P. Perdikaris, D. Venturi, J. O. Royset, and G. E. Karniadakis. Multi-fidelity modelling via recursive co-kriging and Gaussian–Markov random fields. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 471(2179), 2015.

[165] D. Pflüger, B. Peherstorfer, and H.-J. Bungartz. Spatially adaptive sparse grids for high-dimensional data-driven problems. *Journal of Complexity*, 26(5):508 – 522, 2010.

[166] P. Piperni, A. DeBlois, and R. Henderson. Development of a multilevel multidisciplinary-optimization capability for an industrial environment. *AIAA Journal*, 51(10):2335–2352, 2013.

[167] M. Poloczek, J. Wang, and P. I. Frazier. Multi-information source optimization with general model discrepancies. *arXiv preprint arXiv:1603.00389*, 2016.

[168] J. L. Proctor, S. L. Brunton, and J. N. Kutz. Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, 15(1):142–161, 2016.

[169] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. K. Tucker. Surrogate-based analysis and optimization. *Progress in Aerospace Sciences*, 41(1):1 – 28, 2005.

[170] D. Rajnarayan. *Trading Risk and Performance for Engineering Design Optimization using Multifidelity Analyses*. Phd thesis, Stanford University, 2009.

[171] D. Rajnarayan, A. Haas, and I. Kroo. A multifidelity gradient-free optimization method and application to aerodynamic design. In *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Multidisciplinary Analysis Optimization Conferences. American Institute of Aeronautics and Astronautics, 2008.

[172] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[173] M. Rathinam and L. Petzold. A new look at proper orthogonal decomposition. *SIAM Journal on Numerical Analysis*, 41(5):1893–1925, 2003.

[174] R. G. Regis and C. A. Shoemaker. Constrained global optimization of expensive black box functions using radial basis functions. *Journal of Global Optimization*, 31(1):153–171, 2005.

[175] R. G. Regis and C. A. Shoemaker. A stochastic radial basis function method for the global optimization of expensive functions. *INFORMS Journal on Computing*, 19(4):497–509, 2007.

[176] J. M. Reinert and G. E. Apostolakis. Including model uncertainty in risk-informed decision making. *Annals of Nuclear Energy*, 33(4):354–369, 2006.

[177] C. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2004.

[178] G. O. Roberts, A. Gelman, and W. R. Gilks. Weak convergence and optimal scaling of random walk Metropolis algorithms. *Ann. Appl. Probab.*, 7(1):110–120, 02 1997.

[179] G. O. Roberts and J. S. Rosenthal. Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms. *Journal of Applied Probability*, 44(2):458–475, 2007.

[180] G. Rozza, D. B. P. Huynh, and A. Manzoni. Reduced basis approximation and a posteriori error estimation for stokes flows in parametrized geometries: roles of the inf-sup stability constants. *Numerische Mathematik*, 125(1):115–152, 2013.

[181] G. Rozza, D. B. P. Huynh, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Archives of Computational Methods in Engineering*, 15(3):229–275, 2008.

[182] G. Rozza and K. Veroy. On the stability of the reduced basis method for stokes equations in parametrized domains. *Computer Methods in Applied Mechanics and Engineering*, 196(7):1244 – 1260, 2007.

[183] D. Ryckelynck. A priori hyperreduction method: an adaptive approach. *Journal of Computational Physics*, 202(1):346 – 366, 2005.

[184] D. Ryckelynck, F. Chinesta, E. Cueto, and A. Ammar. On the a priori model reduction: Overview and recent developments. *Archives of Computational Methods in Engineering*, 13(1):91–128, 2006.

[185] J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–423, 1989.

[186] S. Sargsyan, S. L. Brunton, and J. N. Kutz. Nonlinear model reduction for dynamical systems using sparse sensor locations from learned libraries. *Physical Review E*, 92:033304, 2015.

[187] P. Schmid and S. J. Dynamic mode decomposition of numerical and experimental data. In *Bull. Amer. Phys. Soc., 61st APS meeting*, page 208. American Physical Society, 2008.

[188] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, 2001.

[189] H. Shah, S. Hosder, S. Koziel, Y. A. Tesfahunegn, and L. Leifsson. Multi-fidelity robust aerodynamic design optimization under mixed uncertainty. *Aerospace Science and Technology*, 45:17–29, 2015.

[190] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.

[191] L. Sirovich. Turbulence and the dynamics of coherent structures. *Quarterly of Applied Mathematics*, pages 561–571, 1987.

[192] R. Srinivasan. *Importance Sampling*. Springer, 2002.

[193] A. M. Stuart. Inverse problems: A Bayesian perspective. *Acta Numerica*, 19:451–559, 2010.

[194] A. L. Teckentrup, P. Jantsch, C. G. Webster, and M. Gunzburger. A multilevel stochastic collocation method for partial differential equations with random input data. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):1046–1074, 2015.

[195] A. L. Teckentrup, R. Scheichl, M. B. Giles, and E. Ullmann. Further analysis of multilevel Monte Carlo methods for elliptic PDEs with random coefficients. *Numerische Mathematik*, 125(3):569–600, 2013.

[196] V. Torczon. On the convergence of pattern search algorithms. *SIAM Journal on Optimization*, 7(1):1–25, 1997.

[197] B. Tracey, D. Wolpert, and J. J. Alonso. Using supervised learning to improve Monte-Carlo integral estimation. *AIAA Journal*, 51(8):2015–2023, 2013.

[198] U. Trottenberg, C. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, 2001.

[199] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz. On dynamic mode decomposition: Theory and applications. *Journal of Computational Dynamics*, 1(2):391–421, 2014.

[200] K. Urban and A. T. Patera. A new error bound for reduced basis approximation of parabolic partial differential equations. *Comptes Rendus Mathematique*, 350(3–4):203 – 207, 2012.

[201] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[202] G. Venter, R. T. Haftka, and J. H. Starnes. Construction of response surface approximations for design optimization. *AIAA Journal*, 36(12):2242–2249, 1998.

[203] K. Veroy and A. Patera. Certified real-time solution of the parametrized steady incompressible Navier-Stokes equations: rigorous reduced-basis a posteriori error bounds. *International Journal for Numerical Methods in Fluids*, 47(8-9):773–788, 2005.

[204] F. Vidal-Codina, N. Nguyen, M. Giles, and J. Peraire. A model and variance reduction method for computing statistical outputs of stochastic elliptic partial differential equations. *Journal of Computational Physics*, 297:700 – 720, 2015.

[205] S. Wild and C. A. Shoemaker. Global convergence of radial basis function trust region derivative-free algorithms. *SIAM Journal of Optimization*, 21(3):761–781, 2011.

[206] R. L. Winkler. Combining probability distributions from dependent information sources. *Management Science*, 27(4):479–488, 1981.

[207] D. Xiu. Fast numerical methods for stochastic computations: A review. *Communications in Computational Physics*, 5:242–272, 2009.

[208] D. Xiu. *Numerical Methods for Stochastic Computations: A Spectral Method Approach*. Princeton University Press, 2010.

[209] D. Xiu and G. E. Karniadakis. The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2):619–644, 2002.

[210] W. Yamazaki, M. Rumpfkeil, and D. Mavriplis. Design optimization utilizing gradient/hessian enhanced surrogate model. In *28th AIAA Applied Aerodynamics Conference*, Fluid Dynamics and Co-located Conferences. American Institute of Aeronautics and Astronautics, 2010.

[211] M. Yano. A space-time Petrov–Galerkin certified reduced basis method: Application to the Boussinesq equations. *SIAM Journal on Scientific Computing*, 36(1):A232–A266, 2014.

[212] H. Yserentant. Hierarchical bases give conjugate gradient type methods a multigrid speed of convergence. *Applied Mathematics and Computation*, 19(1-4):347–358, 1986.

[213] Y. Yue and K. Meerbergen. Accelerating optimization of parametric linear systems by model order reduction. *SIAM Journal on Optimization*, 23(2):1344–1370, 2013.

[214] M. J. Zahr and C. Farhat. Progressive construction of a parametric reduced-order model for PDE-constrained optimization. *International Journal for Numerical Methods in Engineering*, 102(5):1111–1135, 2015.

[215] E. Zio and G. E. Apostolakis. Two methods for the structured assessment of model uncertainty by experts in performance assessments of radioactive waste repositories. *Reliability Engineering & System Safety*, 54(2–3):225–241, 1996.

Algorithm 2 Two-stage MCMC

procedure 2StageMCMC($L^{\text{hi}}, L^{\text{lo}}, p_0, q, m$)
  Choose a starting point $\boldsymbol{z}_0$
  for $i = 1, \ldots, m$ **do**
    Draw candidate $\boldsymbol{z}^{\text{lo}}$ from proposal $q(\cdot|\boldsymbol{z}_{i-1})$
    Compute acceptance probability

$$\alpha^{\text{lo}}(\boldsymbol{z}_{i-1}, \boldsymbol{z}^{\text{lo}}) = \min\left\{1, \frac{q(\boldsymbol{z}_{i-1}|\boldsymbol{z}^{\text{lo}})L^{\text{lo}}(\boldsymbol{y}_{\text{obs}}|\boldsymbol{z}^{\text{lo}})p_0(\boldsymbol{z}^{\text{lo}})}{q(\boldsymbol{z}^{\text{lo}}|\boldsymbol{z}_{i-1})L^{\text{lo}}(\boldsymbol{y}_{\text{obs}}|\boldsymbol{z}_{i-1})p_0(\boldsymbol{z}_{i-1})}\right\}$$

    Set the candidate sample $\boldsymbol{z}^{\text{hi}}$ to

$$\boldsymbol{z}^{\text{hi}} = \begin{cases} \boldsymbol{z}^{\text{lo}}, & \text{with probability } \alpha^{\text{lo}}(\boldsymbol{z}_{i-1}, \boldsymbol{z}^{\text{lo}}), \\ \boldsymbol{z}_{i-1}, & \text{with probability } 1 - \alpha^{\text{lo}}(\boldsymbol{z}_{i-1}, \boldsymbol{z}^{\text{lo}}) \end{cases}$$

    Set the distribution Q to

$$Q(\boldsymbol{z}|\boldsymbol{z}_{i-1}) = \alpha^{\text{lo}}(\boldsymbol{z}_{i-1}, \boldsymbol{z})q(\boldsymbol{z}|\boldsymbol{z}_{i-1}) + \left(1 - \int_{\mathcal{D}} \alpha^{\text{lo}}(\boldsymbol{z}_{i-1}, \boldsymbol{z})q(\boldsymbol{z}|\boldsymbol{z}_{i-1})\mathrm{d}\boldsymbol{z}\right)\delta_{\boldsymbol{z}_{i-1}}(\boldsymbol{z})$$

    Compute acceptance probability

$$\alpha^{\text{hi}}(z_{i-1}, z^{\text{hi}}) = \min\left\{1, \frac{Q(\boldsymbol{z}_{i-1}|\boldsymbol{z}^{\text{hi}})L^{\text{hi}}(\boldsymbol{y}_{\text{obs}}|\boldsymbol{z}^{\text{hi}})p_0(\boldsymbol{z}^{\text{hi}})}{Q(\boldsymbol{z}^{\text{hi}}|\boldsymbol{z}_{i-1})L^{\text{hi}}(\boldsymbol{y}_{\text{obs}}|\boldsymbol{z}_{i-1})p_0(\boldsymbol{z}_{i-1})}\right\}$$

    Set sample $\boldsymbol{z}_i$ to

$$\boldsymbol{z}_i = \begin{cases} \boldsymbol{z}^{\text{hi}}, & \text{with probability } \alpha^{\text{hi}}(\boldsymbol{z}_{i-1}, \boldsymbol{z}^{\text{hi}}), \\ \boldsymbol{z}_{i-1}, & \text{with probability } 1 - \alpha^{\text{hi}}(\boldsymbol{z}_{i-1}, \boldsymbol{z}^{\text{hi}}) \end{cases}$$

  end for
  return $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_m$
end procedure

Algorithm 3 A multifidelity trust-region algorithm

procedure TRUSTREGION($f_{\text{hi}}, \eta_1, \eta_2, \gamma_1, \gamma_2, \Delta^*$)
  Choose a starting point $\boldsymbol{z}_0$ and initial step size $\Delta_0 > 0$
  for $i = 1, 2, 3, \ldots$ until convergence **do**
    Construct low-fidelity model $f_{\text{lo}}^{(i)}$ with

$$f_{\text{lo}}^{(i)}(\boldsymbol{z}_i) = f_{\text{hi}}(\boldsymbol{z}_i)\,, \text{ and } \nabla f_{\text{lo}}^{(i)}(\boldsymbol{z}_i) = \nabla f_{\text{hi}}(\boldsymbol{z}_i)$$

    Find $\boldsymbol{s}_i \in \mathcal{D}$ that solves

$$\begin{aligned} \min \quad & f_{\text{lo}}(\boldsymbol{z}_i + \boldsymbol{s}_i) \\ \text{s.t.} \quad & \|\boldsymbol{s}_i\| \leq \Delta_i \end{aligned}$$

    Update point

$$\boldsymbol{z}_{i+1} = \begin{cases} \boldsymbol{z}_i + \boldsymbol{s}_i\,, & \text{if } f_{\text{hi}}(\boldsymbol{z}_i + \boldsymbol{s}_i) < f_{\text{hi}}(\boldsymbol{z}_i)\,, \\ \boldsymbol{z}_i\,, & \text{otherwise} \end{cases}$$

    Compare actual and estimated decrease in f_{hi}

$$\gamma = \frac{f_{\text{hi}}(\boldsymbol{z}_i) - f_{\text{hi}}(\boldsymbol{z}_i + \boldsymbol{s}_i)}{f_{\text{lo}}^{(i)}(\boldsymbol{z}_i) - f_{\text{lo}}^{(i)}(\boldsymbol{z}_i + \boldsymbol{s}_i)}$$

    Update step size

$$\Delta_{i+1} = \begin{cases} \eta_1 \|\boldsymbol{s}_i\|\,, & \gamma < \gamma_1\,, \\ \min\{\eta_2 \Delta_i, \Delta^*\}\,, & \gamma > \gamma_2\,, \\ \|\boldsymbol{s}_i\|\,, & \text{otherwise} \end{cases}$$

  end for
  return $\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3, \ldots$
end procedure