

# SURVEY OF MULTIFIDELITY METHODS IN UNCERTAINTY PROPAGATION, INFERENCE, AND OPTIMIZATION

BENJAMIN PEHERSTORFER\*, KAREN WILLCOX†, AND MAX GUNZBURGER‡

**Abstract.** 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, where the low-fidelity evaluations arise from an explicit low-fidelity model (e.g., a simplified physics approximation, a reduced model, a data-fit surrogate, etc.) that approximates the same output quantity as the high-fidelity model. 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.

**Key words.** multifidelity; surrogate models; model reduction; multifidelity uncertainty quantification; multifidelity uncertainty propagation; multifidelity statistical inference; multifidelity optimization

**AMS subject classifications.** 65-02, 62-02, 49-02

**1. Introduction.** We begin by introducing the setting and concepts surveyed in this paper: Section 1.1 defines the setting of multifidelity models and Section 1.2 introduces the concepts of multifidelity methods. Section 1.3 discusses different types of low-fidelity models that may arise in the multifidelity setting. Section 1.4 defines the three outer-loop applications of interest: uncertainty propagation, statistical inference, and optimization. Section 1.5 outlines the remainder of the paper.

**1.1. Multifidelity models.** 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. For example, the numerical simulation might involve solving a partial differential equation (PDE), or solving a system of ordinary differential equations, or applying a particle method. Mathematically, we denote a model as a function  $f : \mathcal{Z} \rightarrow \mathcal{Y}$  that maps an input  $\mathbf{z} \in \mathcal{Z}$  to an output  $\mathbf{y} \in \mathcal{Y}$ , where  $\mathcal{Z} \subseteq \mathbb{R}^d$  is the domain of the inputs of the model, with dimension  $d \in \mathbb{N}$ , and  $\mathcal{Y} \subseteq \mathbb{R}^{d'}$  is the domain of the outputs of the model, with dimension  $d' \in \mathbb{N}$ . Model evaluations (i.e., evaluations of  $f$ ) incur computational costs  $c \in \mathbb{R}_+$

---

\*Department of Mechanical Engineering and Wisconsin Institute for Discovery, University of Wisconsin-Madison, Madison, WI 53706

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

‡Department of Scientific Computing, Florida State University, Tallahassee, FL 32306-4120

that typically increase with the accuracy of the approximation of the output, where  $\mathbb{R}_+ = \{x \in \mathbb{R} : x > 0\}$  is the set of positive real numbers.

In many situations, multiple models are available that estimate the same output quantity with varying approximation qualities and varying computational costs. We define a *high-fidelity model*  $f_{\text{hi}} : \mathcal{Z} \rightarrow \mathcal{Y}$  as a model that estimates the output with the accuracy that is necessary for the current task at hand. We define a *low-fidelity model*  $f_{\text{lo}} : \mathcal{Z} \rightarrow \mathcal{Y}$  as a model that estimates the same output with a lower accuracy than the high-fidelity model. The costs  $c_{\text{hi}} \in \mathbb{R}_+$  of the high-fidelity model  $f_{\text{hi}}$  are typically higher than the costs  $c_{\text{lo}} \in \mathbb{R}_+$  of a low-fidelity model  $f_{\text{lo}}$ . More generally, we consider  $k \in \mathbb{N}$  low-fidelity models,  $f_{\text{lo}}^{(1)}, \dots, f_{\text{lo}}^{(k)}$ , that each represent the relationship between the input and the output,  $f_{\text{lo}}^{(i)} : \mathcal{Z} \rightarrow \mathcal{Y}$ ,  $i = 1, \dots, k$ , and we denote the cost of evaluating model  $f_{\text{lo}}^{(i)}$  as  $c_{\text{lo}}^{(i)}$ .

**1.2. Multifidelity methods for the outer loop.** The use of principled approximations to accelerate computational tasks has long been a mainstay of scalable numerical algorithms. For example, quasi-Newton optimization methods [57, 69, 31] construct approximations of Hessians and apply low-rank updates to these approximations during the Newton iterations. Solvers based on Krylov subspace methods [121, 9, 122, 184] and on Anderson relaxation [5, 211, 202] perform intermediate computations in low-dimensional subspaces that are updated as the computation proceeds. Whereas these methods—and many others across the broad field of numerical algorithms—embed principled approximations within a numerical solver, we focus in this paper on the particular class of multifidelity methods that invoke explicit approximate models in solution of an outer-loop problem. We define this class of methods more precisely below; first we introduce the notion of an outer-loop application problem.

We use the term *outer-loop application* to define computational applications that form outer loops around a model—where in each iteration an input  $z \in \mathcal{Z}$  is received and the corresponding model output  $f(z)$  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 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.<sup>1</sup> 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 [182], a categorization that applies to most (if not all)

---

<sup>1</sup>We adopted the term “outer loop,” which is used by a number of people in the community, although it does not appear to have been formally defined. Ref. [116] gives specific examples of outer-loop applications in the context of petroleum production. Ref. [131, Chapter 10.1] discusses outer-loop applications in uncertainty quantification and optimization.

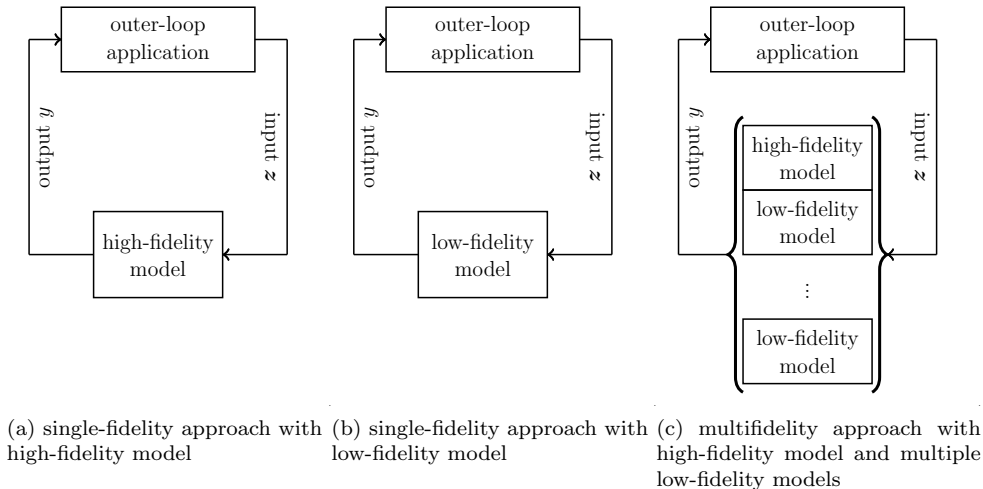


Figure 1: Multifidelity methods combine the high-fidelity model with low-fidelity models. The low-fidelity models are leveraged for speedup and the high-fidelity model is kept in the loop to establish accuracy and/or convergence guarantees on the outer-loop result.

outer-loop applications. We distinguish between many-query and outer-loop applications by considering 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  $f_{\text{hi}}$  in each iteration of the outer loop; however, evaluating the high-fidelity model in each iteration often leads to computational demands that exceed available resources. Simply replacing the high-fidelity model  $f_{\text{hi}}$  with a low-fidelity model  $f_{\text{lo}}$  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 a low-fidelity model  $f_{\text{lo}}$  (or in the general case multiple low-fidelity models  $f_{\text{lo}}^{(1)}, \dots, f_{\text{lo}}^{(k)}, k \in \mathbb{N}$ ), to obtain computational speedups, and (2) they use recourse to the high-fidelity model  $f_{\text{hi}}$  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  $f_{\text{lo}}^{(1)}, \dots, f_{\text{lo}}^{(k)}$ , that provide useful approximations of the high-fidelity model  $f_{\text{hi}}$ , and (2) a *model management* strategy that distributes work among

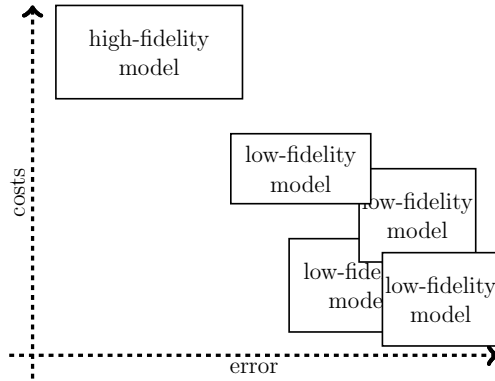


Figure 2: In many situations, different types of low-fidelity models are available, e.g., coarse-grid approximations, projection-based reduced models, data-fit interpolation and regression models, machine-learning-based models, and simplified models. The low-fidelity models vary with respect to error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

the models while providing theoretical guarantees that establish the accuracy and/or convergence of the outer-loop result.

Note that a crucial component of this characterization of multifidelity methods for outer-loop problems is the use of explicit low-fidelity models that approximate the same output quantity as the high-fidelity model. This distinguishes the methods from those that embed approximations within the solver itself, such as the quasi-Newton and Krylov subspace methods discussed above.

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 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, 182, 87, 19], data-fit interpolation and regression models [72, 70], machine-learning-based models such as support vector machines (SVMs) [207, 49, 38], and other simplified models [132, 151], see Figure 2. We further discuss types of low-fidelity models in Section 1.3. 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 restrict the following discussion 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.

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

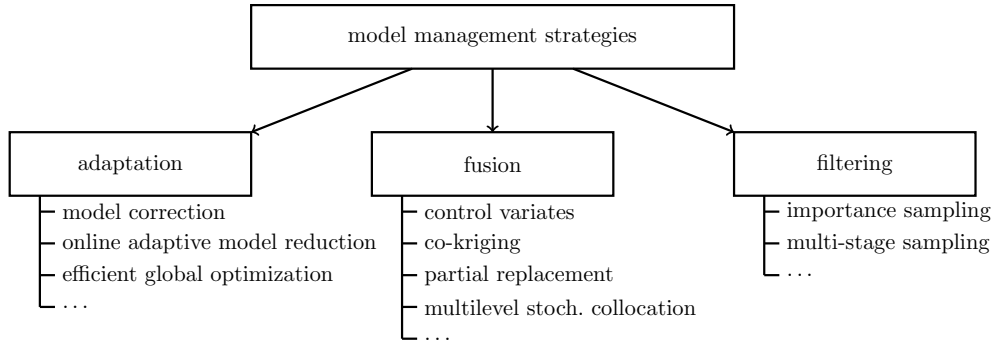


Figure 3: We distinguish between three model management strategies: adaptation, fusion, and filtering.

high-fidelity model only when indicated by a low-fidelity filter.<sup>2</sup> The appropriate model management strategy 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 2.

*Comparison to multilevel methods.* Multilevel methods have a long history in computational science and engineering, e.g., multigrid methods [28, 93, 30, 142, 204], multilevel preconditioners [27, 55], and multilevel function representations [216, 14, 56, 32]. 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 [7, 182, 19] 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 [182, 209, 86], 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

<sup>2</sup>Note that we use the term *filter* to denote selective evaluation based on the low-fidelity model. This differs from the predominant usage in signal processing and uncertainty quantification, where filtering describes the estimation of the state of a dynamical system from noisy and incomplete data (e.g., Kalman filter, particle filter).

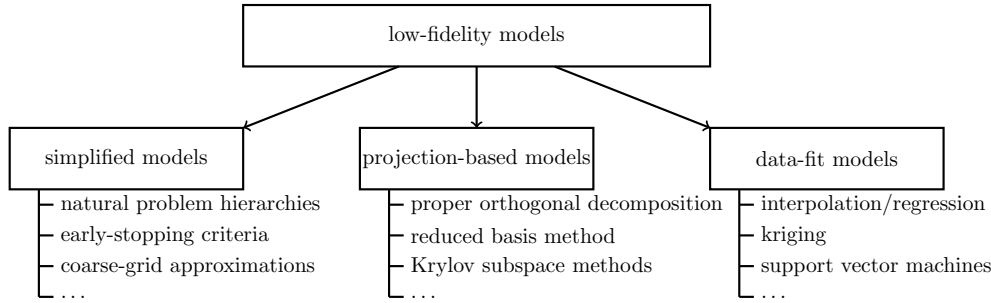


Figure 4: We categorize low-fidelity models as being of three types: simplified models, projection-based models, and data-fit models.

methods establish accuracy 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.

**1.3. Types of low-fidelity models.** We categorize low-fidelity as being of three types: simplified low-fidelity models, projection-based low-fidelity models, and data-fit low-fidelity models. Figure 4 depicts this categorization. For a given application, knowledge of and access to the high-fidelity model affects what kind of low-fidelity models can be created. In some cases, the high-fidelity system has a known structure that can be exploited to create low-fidelity models. In other cases, the high-fidelity models are considered to be “black box”: they can be evaluated at the inputs in  $\mathcal{Z}$  to obtain outputs in  $\mathcal{Y}$ , however no details are available on how the outputs are computed.

*Simplified low-fidelity 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 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 aerodynamic design, an often-employed hierarchy comprises RANS, Euler equations, and potential theory [97]. The supersonic aerodynamic design problem in [43, 42] employs the Euler equations, a vortice lattice model, and a classical empirical model. Similar hierarchies of models exist in other fields of engineering. Models for subsurface flows through karst aquifers reach from simple continuum pipe flow models [36] to coupled Stokes and Darcy systems [35]. 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 [99, 132]. 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. For example, lower-fidelity linearized models are common in aerodynamic and structural analyses [166]. Yet another example is when 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.

*Projection-based low-fidelity models.* Model reduction derives low-fidelity models from a high-fidelity model by mathematically exploiting the problem structure, rather than using domain knowledge of the problem at hand. These methods proceed by identifying a low-dimensional subspace that is constructed so as to retain the essential character of the system input-output map. Projecting the governing equations onto the low-dimensional subspace yields the reduced model. The projection step is generally (but not always) intrusive and requires knowledge of the high-fidelity model structure. There are a variety of ways to construct the low-dimensional subspace, see [19] for a detailed review. One common method is the proper orthogonal decomposition (POD) [191, 21, 175, 120, 119], which uses so-called “snapshots”—state vectors of the high-fidelity model at selected inputs—to construct a basis for the low-dimensional subspace. POD is a popular basis generation method because it is applicable to a wide range of problems, including time-dependent and nonlinear problems [39, 91, 119, 120, 186]. Another basis generation approach is based on centroidal Voronoi tessellation (CVT) [60], 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 [33] discusses details on CVT-based basis construction. A combination of POD and CVT-based basis construction is introduced in [61]. There are also methods based on Krylov subspaces to generate a reduced basis [67, 75], including multivariate Padé approximations and tangential interpolation for linear systems [13, 17, 74, 87]. Dynamic mode decomposition is another basis generation method that is popular in the context of computational fluid dynamics [187, 205, 171]. Balanced truncation [145, 146] 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 [7, 88]. Another basis generation approach is the reduced basis method [182, 86, 183, 181], 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, 86, 92, 182, 206, 215]. Efficient error estimators can also sometimes be provided for other basis generation methods, such as the POD [102, 103].

*Data-fit low-fidelity models.* Data-fit low-fidelity models are derived directly from inputs and the corresponding outputs of the high-fidelity model. Thus, data-fit models can be derived from black-box high-fidelity models because only the inputs and outputs of the high-fidelity model need to be available. In many cases, data-fit models are represented as linear combinations of basis functions. Data-fit models are constructed by fitting the coefficients of the linear combination via interpolation or regression to the inputs and the corresponding high-fidelity model outputs. The choice of the interpolation and regression bases is critical for the approximation quality of the data-fit models. Polynomials, e.g., Lagrange polynomials, are classical basis functions that can be used for deriving data-fit models. Piecewise-polynomial interpolation approaches allow use of low-degree polynomials, which avoids problems

with global polynomial interpolation of high degree, e.g., Runge’s phenomenon. If the inputs are low dimensional, a multi-variate data-fit model can be derived with tensor product approaches. In higher-dimensional settings, discretization methods based on sparse grids [32] can be employed. Radial basis functions are another type of basis functions that are widely used for constructing data-fit models [174, 70]. If based on the Gaussian density function, radial basis functions typically lead to numerically stable computations of coefficients of the linear combination of the data-fit model. Often the radial basis functions depend on hyper-parameters, e.g., the bandwidth of the Gaussian density function. Well-chosen hyper-parameters can greatly improve the approximation accuracy of the data-fit model but the optimization for these hyper-parameters is often computationally expensive [70]. A widely used approach to interpolation with radial basis functions is kriging, for which a sound theoretical understanding has been obtained and efficient approaches to optimize for the hyper-parameters have been developed [141, 185, 112, 138, 174, 72]. In particular, kriging models are equipped with error indicators, see, e.g., [185]. There are also support vector machines [207, 49, 23, 188], which have been developed by the machine-learning community for classification tasks but are now used as surrogates in science and engineering as well, see, e.g., [70, 16, 59, 164].

**1.4. 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 [76, 150, 148, 198, 160]. Note that we consider probabilistic approaches to uncertainty propagation only; other approaches to uncertainty propagation are, e.g., fuzzy set approaches [22] and worst-case scenario analysis [11].

*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. In statistical inference [195, 194, 110, 192], 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 [44, 62, 51, 54].

*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 [24, 112, 71, 70] or by using a low-fidelity model in conjunction with adaptive corrections and a trust-region model-management scheme [1, 2, 24, 135, 172]. 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].

**1.5. Outline of the paper.** The remainder of this paper focuses on model management strategies. Section 2 overviews the model management strategies of adaptation, fusion, and filtering. Sections 3–5 survey specific techniques in the context of uncertainty propagation, inference, and optimization, respectively. The outlook in Section 6 closes the survey.

**2. Multifidelity model management strategies.** 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.

**2.1. 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, 208]. 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 [1, 2]. The correction model is then typically built using Taylor series expansion based on gradients, and possibly also on higher-order derivative information [63]. In [113], 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 [162, 163], 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 [4] adapts localized reduced bases to smooth the transition from one localized reduced basis to another localized basis.

**2.2. 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 [29, 96, 149], 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 [6, 147, 165]. Co-kriging is often used in the context of optimization if gradient information of the high-fidelity model is available, see [71]. The work [123] compares kriging and co-kriging models on aerodynamic test functions. In [214], gradients are computed cheaply with the adjoint method and then used to derive a co-kriging model for design optimization in large design spaces. In [97], co-kriging with gradients and further developments of co-kriging are compared for approximating aerodynamic models of airfoils.

**2.3. 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 MCMC algorithm. For example, in two-stage MCMC [44, 73], 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 [148], 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 [160].

**3. Multifidelity model management in uncertainty propagation.** Inputs of models are often formulated as random variables to describe the stochasticity of the system of interest. With random inputs, the output of the model becomes a random variable as well. Uncertainty propagation aims to estimate statistics of the output random variable [140]. Sampling methods for uncertainty propagation evaluate the model 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 3.1 introduces the problem setup and briefly overviews the Monte Carlo simulation method. Sections 3.2–3.4 discuss multifidelity methods for Monte Carlo based on control variates, importance sampling, and other techniques, respectively. Multifidelity methods for stochastic collocation are discussed in Section 3.5.

**3.1. Uncertainty propagation and Monte Carlo simulation.** Consider the high-fidelity model  $f_{\text{hi}} : \mathcal{Z} \rightarrow \mathcal{Y}$  and let the uncertainties in the inputs be represented by a random variable  $Z$  with probability density function  $p$ . At this point, the only assumption we make on the random variable  $Z$  is that the distribution is absolutely

continuous such that a density function exists. In particular, the random variable  $Z$  can be a non-Gaussian random variable. The goal of uncertainty propagation is to estimate statistics of the random variable  $f_{\text{hi}}(Z)$ , e.g., the expectation

$$(3.1) \quad \mathbb{E}[f_{\text{hi}}] = \int_{\mathcal{Z}} f_{\text{hi}}(\mathbf{z})p(\mathbf{z})d\mathbf{z},$$

and the variance

$$(3.2) \quad \mathbb{V}[f_{\text{hi}}] = \mathbb{E}[f_{\text{hi}}^2] - \mathbb{E}[f_{\text{hi}}]^2,$$

which we assume to exist.

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

$$(3.3) \quad \bar{s}_m^{\text{hi}} = \frac{1}{m} \sum_{i=1}^m f_{\text{hi}}(\mathbf{z}_i).$$

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 mean squared error (MSE) of the Monte Carlo estimator  $\bar{s}_m^{\text{hi}}$  therefore is

$$(3.4) \quad e(\bar{s}_m^{\text{hi}}) = \frac{\mathbb{V}[f_{\text{hi}}]}{m}.$$

The convergence rate  $\mathcal{O}(m^{-1/2})$  of the root mean squared error (RMSE)  $\sqrt{e(\bar{s}_m^{\text{hi}})}$  is low if compared to deterministic quadrature rules, see Section 3.5; however, the rate is independent of the smoothness of the integrand and the dimension  $d$  of the input  $\mathbf{z}$ , 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 [96, 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 3.2, and variance reduction based on importance sampling in Section 3.3.

### 3.2. Multifidelity uncertainty propagation based on control variates.

The control variate framework [96, 29, 149] 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, as discussed in, e.g., [96], the statistics of the auxiliary random variable is known. Extensions relax this requirement by estimating the statistics of the auxiliary random variable from prior information [65, 159]. We now discuss multifidelity approaches that construct auxiliary random variables from low-fidelity models.

**3.2.1. Control variates based on low-fidelity models.** Consider the high-fidelity model  $f_{\text{hi}}$  and  $k \in \mathbb{N}$  low-fidelity models  $f_{\text{lo}}^{(1)}, \dots, f_{\text{lo}}^{(k)}$ . In [150, 164], a multifidelity method is introduced that uses the random variables  $f_{\text{lo}}^{(1)}(Z), \dots, f_{\text{lo}}^{(k)}(Z)$  stemming from the low-fidelity models as control variates for estimating statistics of the random variable  $f_{\text{hi}}(Z)$  of the high-fidelity model. An optimal model management is derived that minimizes the MSE of the multifidelity estimator for a given computational budget. 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 a wide range of low-fidelity model types.

Let  $m_0 \in \mathbb{N}$  be the number of high-fidelity model evaluations. Let  $m_i \in \mathbb{N}$  be the number of evaluations of the low-fidelity model  $f_{\text{lo}}^{(i)}$  for  $i = 1, \dots, k$ , where  $0 < m_0 \leq m_1 \leq \dots \leq m_k$ . The multifidelity approach presented in [164] draws  $m_k$  realizations

$$(3.5) \quad \mathbf{z}_1, \dots, \mathbf{z}_{m_k}$$

from the random variable  $Z$  and computes the model outputs  $f_{\text{hi}}(\mathbf{z}_1), \dots, f_{\text{hi}}(\mathbf{z}_{m_0})$  and

$$(3.6) \quad f_{\text{lo}}^{(i)}(\mathbf{z}_1), \dots, f_{\text{lo}}^{(i)}(\mathbf{z}_{m_i})$$

for  $i = 1, \dots, k$ . These model outputs are used to derive Monte Carlo estimates

$$(3.7) \quad \bar{s}_{m_0}^{\text{hi}} = \frac{1}{m_0} \sum_{j=1}^{m_0} f_{\text{hi}}(\mathbf{z}_j), \quad \bar{s}_{m_i}^{(i)} = \frac{1}{m_i} \sum_{j=1}^{m_i} f_{\text{lo}}^{(i)}(\mathbf{z}_j), \quad i = 1, \dots, k,$$

and

$$(3.8) \quad \bar{s}_{m_{i-1}}^{(i)} = \frac{1}{m_{i-1}} \sum_{j=1}^{m_{i-1}} f_{\text{lo}}^{(i)}(\mathbf{z}_j), \quad i = 1, \dots, k.$$

Note that the estimates (3.8) use the first  $f_{\text{lo}}^{(i)}(\mathbf{z}_1), \dots, f_{\text{lo}}^{(i)}(\mathbf{z}_{m_{i-1}})$  model outputs of (3.6) only, whereas the estimate (3.7) uses all  $m_i$  model outputs (3.6) for  $i = 1, \dots, k$ . Following [164], the multifidelity estimator of  $\mathbb{E}[f_{\text{hi}}]$  is

$$(3.9) \quad \bar{s}^{\text{MF}} = \bar{s}_{m_0}^{\text{hi}} + \sum_{i=1}^k \alpha_i \left( \bar{s}_{m_i}^{(i)} - \bar{s}_{m_{i-1}}^{(i)} \right).$$

The control variate coefficients  $\alpha_1, \dots, \alpha_k \in \mathbb{R}$  balance the term  $\bar{s}_{m_0}^{\text{hi}}$  stemming from the high-fidelity model and the terms  $\bar{s}_{m_i}^{(i)} - \bar{s}_{m_{i-1}}^{(i)}$  from the low-fidelity models. The multifidelity estimator (3.9) 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 (3.9) therefore uses a model management based on fusion, see Section 2. We note that (3.9) could also be viewed as a correction, although the correction is to the estimators stemming from the low-fidelity models, 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

$$\mathbb{E}[\bar{s}^{\text{MF}}] = \mathbb{E}[\bar{s}_{m_0}^{\text{hi}}] + \sum_{i=1}^k \alpha_i \mathbb{E}[\bar{s}_{m_i}^{(i)} - \bar{s}_{m_{i-1}}^{(i)}] = \mathbb{E}[f_{\text{hi}}].$$

Therefore, the MSE 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

$$c(\bar{s}^{\text{MF}}) = m_0 c_{\text{hi}} + \sum_{i=1}^k m_i c_{\text{lo}}^{(i)} = \mathbf{m}^T \mathbf{c},$$

where  $\mathbf{m} = [m_0, m_1, \dots, m_k]^T$  and  $\mathbf{c} = [c_{\text{hi}}, c_{\text{lo}}^{(1)}, \dots, c_{\text{lo}}^{(k)}]^T$ . The high-fidelity model is evaluated at  $m_0$  realizations and the low-fidelity model  $f_{\text{lo}}^{(i)}$  at  $m_i$  realizations of  $Z$ , for  $i = 1, \dots, k$ .

**3.2.2. Multifidelity Monte Carlo.** The multifidelity estimator (3.9) depends on the control variate coefficients  $\alpha_1, \dots, \alpha_k$  and on the number of model evaluations  $m_0, m_1, \dots, m_k$ . In [150, 164], these parameters are chosen such that the MSE of the estimator (3.9) is minimized for a given computational budget  $\gamma \in \mathbb{R}_+$ . The solution to the optimization problem

$$(3.10) \quad \begin{aligned} \min_{\substack{m_0, m_1, \dots, m_k \\ \alpha_1, \dots, \alpha_k}} \quad & e(\bar{s}^{\text{MF}}) \\ \text{s.t.} \quad & m_0 > 0 \\ & m_i \geq m_{i-1}, \quad i = 1, \dots, k \\ & \mathbf{m}^T \mathbf{c} = \gamma \end{aligned}$$

gives the coefficients  $\alpha_1^*, \dots, \alpha_k^*$  and the number of model evaluations  $m_0^*, \dots, m_k^*$  that minimize the MSE of the multifidelity estimator  $\bar{s}^{\text{MF}}$  for the given computational budget  $\gamma$ . The constraints impose that  $0 < m_0^* \leq m_1^* \leq \dots \leq m_k^*$  and that the costs  $c(\bar{s}^{\text{MF}})$  of the estimator equal the computational budget  $\gamma$ .

*Variance of the multifidelity estimator.* Since the multifidelity estimator  $\bar{s}^{\text{MF}}$  is unbiased, we have  $e(\bar{s}^{\text{MF}}) = \mathbb{V}[\bar{s}^{\text{MF}}]$ , and therefore the objective of minimizing the MSE  $e(\bar{s}^{\text{MF}})$  can be replaced with the variance  $\mathbb{V}[\bar{s}^{\text{MF}}]$  in the optimization problem (3.10). The variance  $\mathbb{V}[\bar{s}^{\text{MF}}]$  of the multifidelity estimator  $\bar{s}^{\text{MF}}$  is

$$(3.11) \quad \mathbb{V}[\bar{s}^{\text{MF}}] = \frac{\sigma_{\text{hi}}^2}{m_0} + \sum_{i=1}^k \left( \frac{1}{m_{i-1}} - \frac{1}{m_i} \right) (\alpha_i^2 \sigma_i^2 - 2\alpha_i \rho_i \sigma_{\text{hi}} \sigma_i),$$

where  $-1 \leq \rho_i \leq 1$  is the Pearson correlation coefficient of the random variable  $f_{\text{hi}}(Z)$  and  $f_{\text{lo}}^{(i)}(Z)$  for  $i = 1, \dots, k$ . The quantities

$$\sigma_{\text{hi}}^2 = \mathbb{V}[f_{\text{hi}}], \quad \sigma_i^2 = \mathbb{V}[f_{\text{lo}}^{(i)}], \quad i = 1, \dots, k,$$

are the variances of  $f_{\text{hi}}(Z)$  and  $f_{\text{lo}}^{(i)}(Z)$ , 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 (3.10) has a unique, analytic solution [164]. The optimal control variate coefficients are

$$(3.12) \quad \alpha_i^* = \rho_i \frac{\sigma_{\text{hi}}}{\sigma_i}, \quad i = 1, \dots, k.$$

The optimal number of evaluations  $m_0^*, m_1^*, \dots, m_k^*$  are

$$(3.13) \quad m_0^* = \frac{\gamma}{\mathbf{c}^T \mathbf{r}}, \quad m_i^* = r_i m_0, \quad i = 1, \dots, k,$$

where the components of the vector  $\mathbf{r} = [1, r_1, \dots, r_k]^T \in \mathbb{R}^{k+1}$  are given as

$$(3.14) \quad r_i = \sqrt{\frac{c_{\text{hi}}(\rho_i^2 - \rho_{i+1}^2)}{c_{\text{lo}}^{(i)}(1 - \rho_1^2)}}, \quad i = 1, \dots, k.$$

Note that the convention  $\rho_{k+1} = 0$  is used in (3.14). We refer to [164] for details.

*Interaction of models in multifidelity Monte Carlo.* We compare the multifidelity estimator  $\bar{s}^{\text{MF}}$  to a benchmark Monte Carlo estimator  $\bar{s}^{\text{MC}}$  that uses the high-fidelity model alone. The multifidelity estimator  $\bar{s}^{\text{MF}}$  and the benchmark estimator  $\bar{s}^{\text{MC}}$  have the same costs  $\gamma$ . With the MSE  $e(\bar{s}^{\text{MF}})$  of the multifidelity estimator and the MSE  $e(\bar{s}^{\text{MC}})$  of the benchmark Monte Carlo estimator, the variance reduction ratio is

$$(3.15) \quad \frac{e(\bar{s}^{\text{MF}})}{e(\bar{s}^{\text{MC}})} = \left( \sqrt{1 - \rho_1^2} + \sum_{i=1}^k \sqrt{\frac{c_{\text{lo}}^{(i)}}{c_{\text{hi}}}(\rho_i^2 - \rho_{i+1}^2)} \right)^2.$$

The ratio (3.15) quantifies the variance reduction achieved by the multifidelity estimator compared to the benchmark Monte Carlo estimator. The variance reduction ratio is a sum over the costs  $c_{\text{hi}}, c_{\text{lo}}^{(1)}, \dots, c_{\text{lo}}^{(k)}$  and the correlation coefficients  $\rho_1, \dots, \rho_k$  of all models in the multifidelity estimator. This shows that the contribution of a low-fidelity model to the variance reduction of the multifidelity estimator cannot be determined by the properties of that low-fidelity model alone but only by taking into account all other models that are used in the multifidelity estimator. Thus, the interaction between the models is what drives the efficiency of the multifidelity estimator  $\bar{s}^{\text{MF}}$ . We refer to [164] for an in-depth discussion of the interaction between the models and a more detailed analysis.

*Efficiency of the multifidelity estimator.* It is shown in [164] that the MFMC estimator  $\bar{s}^{\text{MF}}$  is computationally cheaper than the benchmark Monte Carlo estimator that uses the high-fidelity model  $f_{\text{hi}}$  alone if

$$(3.16) \quad \sqrt{1 - \rho_1^2} + \sum_{i=1}^k \sqrt{\frac{c_{\text{lo}}^{(i)}}{c_{\text{hi}}}(\rho_i^2 - \rho_{i+1}^2)} < 1$$

The inequality (3.16) emphasizes that both correlation and costs of the models are critical for an efficient multifidelity estimator.

*Algorithm.* Algorithm 1 summarizes the multifidelity Monte Carlo method as presented in [164]. Inputs are the models  $f_{\text{hi}}, f_{\text{lo}}^{(1)}, \dots, f_{\text{lo}}^{(k)}$  and the variances  $\sigma_{\text{hi}}$  and  $\sigma_1, \dots, \sigma_k$ . The inputs  $\rho_i$  are the correlation coefficients of the random variable

**Algorithm 1** Multifidelity Monte Carlo

- 
- 1: **procedure** MFMC( $f_{\text{hi}}, f_{\text{lo}}^{(1)}, \dots, f_{\text{lo}}^{(k)}, \sigma_{\text{hi}}, \sigma_1, \dots, \sigma_k, \rho_1, \dots, \rho_k, c_{\text{hi}}, c_{\text{lo}}^{(1)}, \dots, c_{\text{lo}}^{(k)}, \gamma$ )
  - 2:   Ensure  $f_{\text{hi}}, f_{\text{lo}}^{(1)}, \dots, f_{\text{lo}}^{(k)}$  are ordered as described in [164, Section 3.5]
  - 3:   Set  $\rho_{k+1} = 0$  and define vector  $\mathbf{r} = [1, r_1, \dots, r_k]^T \in \mathbb{R}_+^{k+1}$  as

$$r_i = \sqrt{\frac{c_{\text{hi}}(\rho_i^2 - \rho_{i+1}^2)}{c_{\text{lo}}^{(i)}(1 - \rho_1^2)}}, \quad i = 1, \dots, k$$

- 4:   Select number of model evaluations  $\mathbf{m}^* \in \mathbb{R}_+^{k+1}$  as

$$\mathbf{m}^* = \left[ \frac{\gamma}{\mathbf{c}^T \mathbf{r}}, r_1 m_0^*, \dots, r_k m_0^* \right]^T \in \mathbb{R}_+^{k+1}$$

- 5:   Set coefficients  $\boldsymbol{\alpha}^* = [\alpha_1^*, \dots, \alpha_k^*]^T \in \mathbb{R}^k$  to

$$\alpha_i^* = \frac{\rho_i \sigma_{\text{hi}}}{\sigma_i}, \quad i = 1, \dots, k$$

- 6:   Draw  $\mathbf{z}_1, \dots, \mathbf{z}_{m_k^*} \in \mathcal{Z}$  realizations of  $Z$
  - 7:   Evaluate high-fidelity model  $f_{\text{hi}}$  at realizations  $\mathbf{z}_1, \dots, \mathbf{z}_{m_0^*}$
  - 8:   Evaluate model  $f_{\text{lo}}^{(i)}$  at realizations  $\mathbf{z}_1, \dots, \mathbf{z}_{m_i^*}$  for  $i = 1, \dots, k$
  - 9:   Compute the multifidelity estimate  $\bar{s}^{\text{MF}}$  as in (3.9)
  - 10: **return** multifidelity estimate  $\bar{s}^{\text{MF}}$
  - 11: **end procedure**
- 

$f_{\text{hi}}(Z)$  stemming from the high-fidelity model and the random variables  $f_{\text{lo}}^{(i)}(Z)$  for  $i = 1, \dots, k$ . The costs of the models are  $c_{\text{hi}}, c_{\text{lo}}^{(1)}, \dots, c_{\text{lo}}^{(k)}$  and the computational budget is  $\gamma$ . Line 2 of Algorithm 1 ensures that the ordering of the models is used that minimizes the MSE of the multifidelity estimator, see [164, Section 3.5] for details. Line 3 defines the vector of ratios  $\mathbf{r} = [r_0, r_1, \dots, r_k]^T$ , cf. (3.14). The number of model evaluations  $m_0^*, m_1^*, \dots, m_k^*$  of the models  $f_{\text{hi}}, f_{\text{lo}}^{(1)}, \dots, f_{\text{lo}}^{(k)}$  are derived from  $\mathbf{r}$  as in (3.13). The control variate coefficients  $\alpha_1^*, \dots, \alpha_k^*$  are obtained as in (3.12). In line 6,  $m_k^*$  realizations  $\mathbf{z}_1, \dots, \mathbf{z}_{m_k^*}$  are drawn from the random variable  $Z$ . The high-fidelity model  $f_{\text{hi}}$  is evaluated at the realizations  $\mathbf{z}_1, \dots, \mathbf{z}_{m_0^*}$  and models  $f_{\text{lo}}^{(i)}$  are evaluated at  $\mathbf{z}_1, \dots, \mathbf{z}_{m_i^*}$  for  $i = 1, \dots, k$ . The multifidelity estimate is obtained as in (3.9) and returned.

**3.2.3. Other uses of control variates as a multifidelity technique.** In [25, 26], 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 [210] uses error estimators to combine reduced basis models with control variates. The StackMC method presented in [203] successively constructs machine-learning-based low-fidelity models and combines them with the control variate framework. In [151], 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”.

The multilevel Monte Carlo method [98, 76] 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., [45, Theorem 1]. In the setting of stochastic differential equations and coarse-grid approximations, multilevel Monte Carlo has been very successful, see, e.g., [45, 199], the recent advances on multi-index Monte Carlo [95], and the nesting of multilevel Monte Carlo and control variates [155] for detailed studies and further references.

**3.3. Multifidelity uncertainty propagation based on importance sampling.** Importance sampling [96] uses a problem-dependent sampling strategy. The goal is an estimator with a lower variance than a Monte Carlo estimator such as (3.3). 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.

**3.3.1. Importance sampling.** Consider the indicator function  $I_{\text{hi}} : \mathcal{Z} \rightarrow \{0, 1\}$  defined as

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

We define the set  $\mathcal{I} = \{\mathbf{z} \in \mathcal{Z} \mid I_{\text{hi}}(\mathbf{z}) = 1\}$ . 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

$$\text{supp}(p) = \{\mathbf{z} \in \mathcal{Z} : p(\mathbf{z}) > 0\}$$

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.,  $\text{supp}(p) \subset \text{supp}(q)$ , then the expectation



$\mathbb{E}_p[I_{\text{hi}}]$  can be rewritten in terms of the biasing density  $q$  as

$$\mathbb{E}_p[I_{\text{hi}}] = \int_{\mathcal{Z}} I_{\text{hi}}(\mathbf{z})p(\mathbf{z})d\mathbf{z} = \int_{\mathcal{Z}} I_{\text{hi}}(\mathbf{z}')q(\mathbf{z}')\frac{p(\mathbf{z}')}{q(\mathbf{z}')}d\mathbf{z}' = \mathbb{E}_q \left[ I_{\text{hi}}\frac{p}{q} \right],$$

where the ratio  $p/q$  serves as a weight.

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

$$(3.17) \quad \bar{s}_m^{\text{IS}} = \frac{1}{m} \sum_{i=1}^m I_{\text{hi}}(\mathbf{z}'_i) \frac{p(\mathbf{z}'_i)}{q(\mathbf{z}'_i)}$$

is evaluated for realizations  $\mathbf{z}'_1, \dots, \mathbf{z}'_m \in \mathcal{Z}$  of the random variable  $Z'$ . The MSE of the estimator (3.17) is

$$(3.18) \quad e(\bar{s}_m^{\text{IS}}) = \frac{\mathbb{V}_q[I_{\text{hi}}\frac{p}{q}]}{m}.$$

*Variance of importance sampling estimator.* The variance in (3.18) is with respect to the biasing density  $q$ , cf. Section 3.1 and the MSE of the Monte Carlo estimator (3.4). Therefore, the goal is to construct a biasing distribution with

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

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

### 3.3.2. Construction of the biasing distribution with low-fidelity models.

The multifidelity importance sampling approach introduced in [160] 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 2.

In step one, the low-fidelity model  $f_{\text{lo}}$  is evaluated at a large number  $n \in \mathbb{N}$  of realizations  $\mathbf{z}_1, \dots, \mathbf{z}_n$  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

$$\{\mathbf{z}_i \mid I_{\text{lo}}(\mathbf{z}_i) = 1, i = 1, \dots, n\}$$

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

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

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, 143, 161].

In step two, the high-fidelity model is evaluated at realizations  $\mathbf{z}'_1, \dots, \mathbf{z}'_m$  from the biasing random variable  $Z'$  with biasing density  $q$ . From the high-fidelity model evaluations  $f_{\text{hi}}(\mathbf{z}'_1), \dots, f_{\text{hi}}(\mathbf{z}'_m)$  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 estimator 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  $I_{\text{lo}}$  in the second step of the multifidelity approach would lead to a biased estimator of  $\mathbb{E}_p[I_{\text{hi}}]$  because  $f_{\text{lo}}$  is only an approximation of  $f_{\text{hi}}$  and thus  $\mathbb{E}_p[I_{\text{hi}}] \neq \mathbb{E}_p[I_{\text{lo}}]$  in general.

**3.4. Other model management strategies for probability estimation and limit state function evaluation.** The multifidelity approaches for estimating  $\mathbb{E}_p[I_{\text{hi}}]$  that are discussed in Section 3.3 first use the low-fidelity model to construct a biasing density  $q$  and then the high-fidelity model to estimate the failure probability  $\mathbb{E}_p[I_{\text{hi}}]$  with importance sampling. Under mild conditions on the biasing density  $q$  derived from the low-fidelity model, the importance sampling estimator using the high-fidelity model is an unbiased estimator of the failure probability  $\mathbb{E}_p[I_{\text{hi}}]$ . In this section, we review multifidelity methods that combine low- and high-fidelity model evaluations to obtain an indicator function  $\tilde{I}$  that approximates  $I_{\text{hi}}$  and that is computationally cheaper to evaluate than  $I_{\text{hi}}$ . Thus, instead of exploiting the two-step importance sampling procedure as in Section 3.3, the techniques in this section leverage the low- and high-fidelity model to approximate  $I_{\text{hi}}$  with  $\tilde{I}$  with the aim that the error  $|\mathbb{E}_p[\tilde{I}] - \mathbb{E}_p[I_{\text{hi}}]|$  is small.

The multifidelity approach introduced in [128] is based on filtering to combine low- and high-fidelity model outputs to obtain an approximation  $\tilde{I}$  of  $I_{\text{hi}}$ . Let  $f_{\text{lo}}$  be a low-fidelity model and  $f_{\text{hi}}$  the high-fidelity model. Let further  $\gamma > 0$  be a positive threshold value. The approach in [128] considers the indicator function

$$\tilde{I}(\mathbf{z}) = \begin{cases} 1, & f_{\text{lo}}(\mathbf{z}) < -\gamma \text{ or } (|f_{\text{lo}}(\mathbf{z})| \leq \gamma \text{ and } f_{\text{hi}}(\mathbf{z}) < 0) \\ 0, & \text{else} \end{cases},$$

which evaluates to 1 at an input  $\mathbf{z}$  if either  $f_{\text{lo}}(\mathbf{z}) < -\gamma$  or  $|f_{\text{lo}}(\mathbf{z})| \leq \gamma$  and  $f_{\text{hi}}(\mathbf{z}) < 0$ . Evaluating the indicator function  $\tilde{I}$  at  $\mathbf{z}$  means that first the low-fidelity model  $f_{\text{lo}}$  is evaluated at  $\mathbf{z}$ . If  $f_{\text{lo}}(\mathbf{z}) < -\gamma$ , then 1 is returned, and no high-fidelity model evaluation is necessary. Similarly, if  $f_{\text{lo}}(\mathbf{z}) > \gamma$ , then the indicator function  $\tilde{I}$  evaluates to 0 without requiring a high-fidelity model evaluation. However, if  $|f_{\text{lo}}(\mathbf{z})| \leq \gamma$ , then the input  $\mathbf{z}$  lies near the failure boundary, and the high-fidelity model is evaluated to decide whether the indicator function returns 0 or 1. How often the high-fidelity model is evaluated is determined by the positive threshold value  $\gamma$ . The choice of  $\gamma$  directly depends on the error of the low-fidelity model  $f_{\text{lo}}$  in a certain norm. If the error of  $f_{\text{lo}}$  is known, the work [128] establishes a convergence theory under mild conditions on the error of  $f_{\text{lo}}$ . In particular, the authors of [128] show that the error  $|\mathbb{E}_p[\tilde{I}] - \mathbb{E}_p[I_{\text{hi}}]|$  can be reduced below any  $\epsilon > 0$  by a choice of  $\gamma$  that depends on the error of  $f_{\text{lo}}$ . If the error of the low-fidelity model is unknown, the work [128] proposes an iterative heuristic approach that avoids the choice of  $\gamma$ . In [127], the multifidelity approach of [128] is extended to importance sampling with the cross-entropy method. Similarly to the approaches in [128, 127], the work [41] switches between low- and high-fidelity model evaluations by relying on *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 [16, 59, 163] all use a model management strategy based on adaptation to estimate a failure boundary or failure probability. In [16], a support vector machine is used to derive a low-fidelity model of the limit state function in failure probability estimation and design. The authors decompose the input space using decision boundaries obtained via the support vector machine, and so handle the discontinuities that arise when approximating the limit state function. An adaptive sampling scheme is introduced that refines the low-fidelity model along the failure boundary. In [59], another approach is introduced that uses a support vector machine to approximate the failure boundary. It is proposed to train the support vector machine with data obtained from a low- and a high-fidelity model. With the low-fidelity model, an initial approximation of the failure boundary is obtained by extensively sampling the input domain, which is computationally tractable because the low-fidelity model is cheap to evaluate. The training of the support vector machine for approximating the failure boundary corresponding to the high-fidelity model is then initialized with the approximate boundary obtained with the low-fidelity model. Additional samples are drawn in regions where the low- and the high-fidelity failure boundary differ, to refine the approximation of the high-fidelity failure boundary. The work [163] presents an online adaptive reduced modeling approach, which is demonstrated in the context of failure probability estimation. To increase the accuracy of the reduced model, it is adapted to the failure boundary as the estimation of the failure probability proceeds. The adaptation is performed via low-rank updates to the basis matrix of the reduced model. The low-rank updates are derived from sparse samples of the high-fidelity model.

**3.5. Stochastic collocation and multifidelity.** Stochastic collocation methods [12, 154, 89] compute statistics such as the expectation (3.1) and the variance (3.2) by using a deterministic quadrature rule instead of the Monte Carlo method. The quadrature rules are often based on sparse grids [32, 154] to perform the quadrature efficiently for high-dimensional inputs.

In [64], 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 2. The authors of [64] 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 [148]. 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 [198], 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.

**4. 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 [195, 194, 110, 192]. 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 [140]. Detailed discussions of these and many other important aspects of statistical inference can be found in the literature, e.g., [177, 110, 192, 130]. 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 4.1 describes the problem setup. Section 4.2 describes a two-stage MCMC framework and Section 4.3 discusses a framework where a low-fidelity model is adapted while it is evaluated in an MCMC algorithm. A Bayesian approach to model and correct the error of low-fidelity models is discussed in Section 4.4.

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

$$(4.1) \quad \mathbf{y}_{\text{obs}} = f_{\text{hi}}(\mathbf{z}) + \boldsymbol{\epsilon},$$

where  $\boldsymbol{\epsilon}$  is a random vector that captures the measurement error, noise, and other uncertainties of the observation  $\mathbf{y}_{\text{obs}}$ . 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

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

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

$$(4.2) \quad L(\mathbf{y}_{\text{obs}}|\mathbf{z}) \propto \exp(-\Phi(\mathbf{z})).$$

An evaluation of the likelihood  $L$  entails an evaluation of the high-fidelity model  $f_{\text{hi}}$ . The posterior probability density is

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

We note that there are many challenging and important questions regarding the formulation of an inverse problem in the Bayesian setting. For example, the selection of the prior is a delicate and problem-dependent question, see for example [110, Section 3.3] for a discussion of prior models. We do not address those issues here, but

**Algorithm 2** Metropolis-Hastings

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

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

- 6:     Set the sample  $\mathbf{z}_i$  to

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

- 7:   **end for**
  - 8:   **return**  $\mathbf{z}_1, \dots, \mathbf{z}_m$
  - 9: **end procedure**
- 

note that our stated goal of a multifidelity formulation is to recover a solution of the outer-loop problem (here the inverse problem) that retains the accuracy of the high-fidelity formulation. Thus, the multifidelity approaches described below will inherit the choices made in the high-fidelity formulation of the Bayesian inverse problem.

*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  $h : \mathcal{Z} \rightarrow \mathbb{R}$  with respect to the posterior distribution. For example, one could be interested in the expected value of  $h$  over the posterior distribution

$$(4.3) \quad \mathbb{E}[h] = \int_{\mathcal{Z}} h(\mathbf{z})p(\mathbf{z}|\mathbf{y}_{\text{obs}})d\mathbf{z}.$$

MCMC methods are a popular way to sample from the posterior distribution, which have been studied extensively [178, 90, 139, 80, 50, 137, 53, 52]; see also the books [77, 130]. 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 2 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  $i = 1, \dots, m$ , a candidate sample  $\mathbf{z}^*$  is drawn from the proposal  $q(\cdot|\mathbf{z}_{i-1})$  that depends on the sample  $\mathbf{z}_{i-1}$  of the previous iteration. The candidate sample  $\mathbf{z}^*$  is accepted  $\mathbf{z}_i = \mathbf{z}^*$  with the probability  $\alpha(\mathbf{z}^*, \mathbf{z}_{i-1})$ , which depends on the ratio of the likelihood at the candidate sample  $\mathbf{z}^*$  and the sample  $\mathbf{z}_{i-1}$  of the previous iteration. If the candidate sample is rejected, then  $\mathbf{z}_i = \mathbf{z}_{i-1}$ . Algorithm 2 returns the samples  $\mathbf{z}_1, \dots, \mathbf{z}_m$ .

*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  $\mathbf{z}_0$ . We refer to the literature [177] for more details and further practical considerations.

*Efficiency of MCMC sampling.* Once samples  $\mathbf{z}_1, \dots, \mathbf{z}_m$  are drawn with an MCMC algorithm, they can be used to, e.g., estimate the expectation (4.3) as

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

MCMC generates correlated samples  $\mathbf{z}_1, \dots, \mathbf{z}_m$ . 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  $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \dots$  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  $\rho_j = \text{corr}(h(\mathbf{z}_1), h(\mathbf{z}_{j+1}))$  is the correlation between  $h(\mathbf{z}_1)$  and  $h(\mathbf{z}_{j+1})$ . 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 [130, p. 125f] for a detailed derivation and further references. This means that there are two ways to improve the efficiency of sampling with MCMC [54]: (1) Increase the effective sample size for a given number of MCMC iterations with, e.g., adaptive MCMC [78, 79, 179, 90]. (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 [44, 73].

**4.2. Two-stage Markov chain Monte Carlo.** Two-stage MCMC methods aim to increase the number of MCMC 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  $f_{\text{hi}}$  evaluation in the case of the likelihood  $L$  as defined in (4.2). Two-stage MCMC methods employ delayed acceptance or delayed rejection strategies that use multifidelity models to reduce the number of samples evaluated using the expensive high-fidelity model.

The work [44, 73] proposes a two-stage delayed acceptance MCMC sampling. The candidate sample  $\mathbf{z}^*$  has to be accepted with the likelihood induced by a low-fidelity model first, before  $\mathbf{z}^*$  is passed on to be either accepted or rejected with the likelihood induced by the high-fidelity model.

Multi-stage MCMC methods based on delayed rejection, in contrast to delayed acceptance, have been proposed in [200, 144, 85]. A candidate sample that is rejected

in one stage is retried with a different proposal distribution at a subsequent stage. Ref. [200] suggests using an independence sampler based on a density that is thought to be a good approximation of the posterior density on the first stage. In case the density corresponding to the independence sampler is a poor approximation, a random walk is used on the second stage. Another approach discussed in [200] is to use a proposal based on a local quadratic approximation of the log posterior density, which is restricted to smaller and smaller neighborhoods of the current point in a trust-region fashion. In [85], the proposal at the first stage is a normal distribution. The proposal at the second stage is a normal distribution with the same mean but with a higher variance. In principle, the proposals in the context of delayed rejection MCMC, just as in delayed acceptance MCMC, can also be derived from models with different fidelities and costs (although we are not aware of such implementations in the literature that pre-date the work by [44, 73]).

*Two-stage delayed acceptance MCMC algorithm.* The two-stage MCMC method introduced in [44] is summarized in Algorithm 3. Inputs are the likelihood  $L^{\text{hi}}$  and  $L^{\text{lo}}$ , 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  $i = 1, \dots, m$ . The first stage of Algorithm 3 (lines 4–6) proceeds as the Metropolis-Hastings algorithm with a candidate sample  $\mathbf{z}^{\text{lo}}$  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  $\mathbf{z}^{\text{hi}}$ , which either is  $\mathbf{z}^{\text{hi}} = \mathbf{z}^{\text{lo}}$  (accept) or  $\mathbf{z}^{\text{hi}} = \mathbf{z}_{i-1}$  (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  $\mathbf{z}^{\text{hi}}$  of the first stage. If the sample  $\mathbf{z}^{\text{hi}}$  is accepted in the second stage, then the algorithm sets  $\mathbf{z}_i = \mathbf{z}^{\text{hi}}$ . If the sample  $\mathbf{z}^{\text{hi}}$  is rejected in the second stage, then  $\mathbf{z}_i = \mathbf{z}_{i-1}$ . Note that in case the first stage rejected the sample  $\mathbf{z}^{\text{lo}}$ , i.e.,  $\mathbf{z}^{\text{hi}} = \mathbf{z}_{i-1}$ , no high-fidelity model evaluation is necessary in the second stage because the high-fidelity model output at  $\mathbf{z}_{i-1}$  is available from the previous iteration. The proposal distribution  $Q$  in the second stage depends on the likelihood  $L^{\text{lo}}$  of the low-fidelity model. Note that  $\delta_{\mathbf{z}_{i-1}}$  is the Dirac mass at  $\mathbf{z}_{i-1}$ . Algorithm 3 returns the samples  $\mathbf{z}_1, \dots, \mathbf{z}_m$ .

*Efficiency of two-stage delayed acceptance MCMC.* The key to the efficiency of Algorithm 3 is that no high-fidelity model evaluation is necessary in the second stage if the candidate sample  $\mathbf{z}^{\text{lo}}$  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 [44] 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 2.

*Other multifidelity MCMC approaches.* In [62], two-stage MCMC is seen as a preconditioned Metropolis-Hastings algorithm. The low-fidelity model in [62] is a coarse-scale model of a high-fidelity multiscale finite volume model. We also mention [101], where multiple MCMC chains from coarse- (low-fidelity) and fine-scale (high-fidelity) models are coupled using a product chain. The work [58] 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.

**4.3. Markov chain Monte Carlo with adaptive low-fidelity models.** In [54] an algorithm for combining high- and low-fidelity models in MCMC sampling

**Algorithm 3** Two-stage MCMC

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

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

- 6:     Set the candidate sample  $\mathbf{z}^{\text{hi}}$  to

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

- 7:     Set the distribution  $Q$  to

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

- 8:     Compute acceptance probability

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

- 9:     Set sample  $\mathbf{z}_i$  to

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

- 10:   **end for**
  - 11:   **return**  $\mathbf{z}_1, \dots, \mathbf{z}_m$
  - 12: **end procedure**
- 

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 reduced model in [54], see Section 1.3. 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 almost 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 [54] improve the low-fidelity model during the MCMC iterations, and consequently increase the acceptance probability in the second stage of the MCMC iterations. The authors show that under certain technical conditions, the Hellinger distance  $d_{\text{Hell}}(p_{\text{hi}}, p_{\text{lo}})$  between the posterior distribution  $p_{\text{hi}}$  corresponding to the high-fidelity model  $f_{\text{hi}}$  and the approximate posterior distribution  $p_{\text{lo}}$  induced by the reduced model  $f_{\text{lo}}$  decreases as the reduced model  $f_{\text{lo}}$  is improved during the MCMC iterations. Thus, for any  $\epsilon > 0$ , the Hellinger distance between the high-fidelity and low-fidelity posterior distribution can be bounded

$$d_{\text{Hell}}(p_{\text{hi}}, p_{\text{lo}}) \leq \epsilon$$

after a sufficiently large number of MCMC iterations. The resulting low-fidelity model is data-driven because it uses information provided by the observation  $\mathbf{y}_{\text{obs}}$ , rather than only prior information. Using samples from this adaptive two-stage MCMC approach yields unbiased Monte Carlo estimators, see the analysis in [54].

**4.4. Bayesian estimation of low-fidelity model error.** In [110, 111], a multifidelity approach for Bayesian inference is presented that relies on a Bayesian approximate error model of the low-fidelity model derived from the high-fidelity model. The inference is performed with the low-fidelity model but an additional term is introduced that quantifies the error between the low-fidelity and the high-fidelity model. The error term relies on the high-fidelity model and is used to correct for the error introduced by the low-fidelity model. The approach follows the Bayesian paradigm and considers the error term as a random variable, which becomes another noise term in the Bayesian formulation of the inverse problem. Within our categorization, this approach uses a model management based on model adaptation.

In [110, 111], the relationship  $\mathbf{y}_{\text{obs}} = f_{\text{hi}}(\mathbf{z}) + \epsilon$  is formulated as

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

which can be rewritten as

$$\mathbf{y}_{\text{obs}} = f_{\text{lo}}(\mathbf{z}) + \mathbf{e}(\mathbf{z}) + \epsilon,$$

with the error term  $\mathbf{e}(\mathbf{z})$ . If  $f_{\text{hi}}$  and  $f_{\text{lo}}$  are linear in  $\mathbf{z}$ , the sum of the covariance of the error  $\mathbf{e}(\mathbf{z})$  and noise term  $\epsilon$  can be derived explicitly under certain assumptions [110, 111]. If  $f_{\text{hi}}$  and  $f_{\text{lo}}$  are nonlinear, a Gaussian approximation of the covariance is proposed in [10].

The Bayesian approximate error model is used in [10, 117] for a diffuse optical tomography problem, where the low-fidelity model corresponds to a coarse-grid finite-element approximation and the high-fidelity model is a fine-grid approximation. In [197, 196], diffuse optical tomography is considered with a simplified-physics low-fidelity model. The error introduced by the low-fidelity model is corrected with the Bayesian approximate error modeling approach. Correcting for errors due to truncation of the computational domain and unknown domain shape are investigated in [125, 153]. The Bayesian approximate error model approach was verified on real data in [152]. Extensions to time-dependent problems are presented in [106, 105].

We note that estimating the error term  $\mathbf{e}$  is related to estimating the model inadequacy of the low-fidelity model with respect to the high-fidelity model. Often, Gaussian-process models are used to attempt to correct model inadequacy [114, 115, 100, 157, 18]. The corrected low-fidelity model is typically cheap to evaluate, because the Gaussian-process model is cheap, and so the corrected low-fidelity model can be used in MCMC sampling with the aim of decreasing the costs per MCMC iteration compared to using the high-fidelity model. The work [110, 111] can be seen as dealing with model inadequacy in the specific context of inverse problems.

**5. 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 5.1 formalizes the optimization task. Section 5.2 discusses global multifidelity optimization, which searches over the entire feasible domain, and Section 5.3 reviews local multifidelity optimization, which searches for a locally optimal solution, typically in a neighborhood of an initial input.

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

$$(5.1) \quad \min_{\mathbf{z} \in \mathcal{Z}} f_{\text{hi}}(\mathbf{z}).$$

Typically, the optimal input  $\mathbf{z}^*$  is obtained in an iterative way, where a sequence of inputs  $\mathbf{z}_1, \mathbf{z}_2, \dots$  is constructed such that this sequence  $(\mathbf{z}_i)$  converges to the optimal input  $\mathbf{z}^*$  in a certain sense [156].

*Local and global optimization.* We distinguish between local and global optimization approaches. Global optimization searches over the entire feasible domain  $\mathcal{Z}$  for a minimizer  $\mathbf{z}^*$ , whereas local optimization terminates when a local optimum is found. Local optimization thus searches in a neighborhood of an initial point  $\mathbf{z} \in \mathcal{Z}$ . Global methods typically do not require the gradient of  $f_{\text{hi}}$ , 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 [133]).

**5.2. Global multifidelity optimization.** A large class of global multifidelity optimization methods uses adaptation as a model management strategy. These methods search for a minimizer with respect to an adaptively refined low-fidelity model  $f_{\text{lo}}$ . They guarantee that the resulting optimal solution is also a minimizer of the high-fidelity model  $f_{\text{hi}}$  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).

**5.2.1. 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 1.3. Let  $\mathbf{z}$  be the input that minimizes the low-fidelity model  $f_{\text{lo}}$  at the current iteration. Sampling at a new point  $\mathbf{z}' \in \mathcal{Z}$  means evaluating the high-fidelity model at  $\mathbf{z}'$  and then adapting the low-fidelity model with information obtained from  $f_{\text{hi}}(\mathbf{z}')$ .

EGO provides a formulation for choosing the new sample point  $\mathbf{z}'$ . It does this by considering the value  $f_{\text{hi}}(\mathbf{z}')$  of the high-fidelity model at the new sampling point  $\mathbf{z}'$  to be uncertain before the high-fidelity model is evaluated at  $\mathbf{z}'$ . The uncertainty in  $f_{\text{hi}}(\mathbf{z}')$  is modeled as a Gaussian random variable  $Y$  with mean and standard deviation given by the kriging (low-fidelity) model, see Section 1.3. The improvement at point  $\mathbf{z}'$  is then given by  $I(\mathbf{z}') = \max\{f_{\text{hi}}(\mathbf{z}) - Y, 0\}$ , and the expected improvement is

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

The expected improvement at a point  $\mathbf{z}'$  can be efficiently computed by exploiting the cheap computation of the MSE (and therefore standard deviation) of kriging models [72, 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].

EGO can be made globally convergent and does not require high-fidelity model derivatives [108]. However, as discussed in [108, 133], 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, 72] for details and possible improvements. The work in [173], 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 [40], where EGO is equipped with an adaptive target method that learns from previous iterations how much improvement to expect in the next iteration.

**5.2.2. Other approaches to combine multifidelity models in the context of global optimization.** In the work [3], 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 [213]. The authors of [81] 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.

In [84], Gaussian-process low-fidelity models are adapted via active learning, instead of EGO. The approach samples the input space and efficiently fits local Gaussian-process models. These local models are obtained via the treed Gaussian-process implementation [83] that partitions the input space into several regions and

builds separate Gaussian-process models in each region. The approach in [84] is particularly suited to the case where sampling the high-fidelity model entails large-scale, parallel computations. In [82], Gaussian-process models and expected improvement are developed for constrained optimization. The augmented Lagrangian is used to reduce the constrained optimization problem into an unconstrained one, for which Gaussian-process models can be derived.

There are multifidelity optimization methods based on pattern search [104, 201] that use filtering as model management strategy. For example, the multifidelity pattern search algorithm presented in [24] uses low-fidelity models to provide additional search directions, while preserving the convergence to a minimizer of the high-fidelity model. In [193], a Bayesian approach to pattern search with Gaussian-process models is presented. The low-fidelity model is used to guide the pattern search. We also refer to [172] for a discussion of pattern search algorithms with low-fidelity models.

**5.3. 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. In each optimization iteration, the high-fidelity model is evaluated at the minimizer of the low-fidelity model. The corresponding high-fidelity model output is then used to adapt the low-fidelity model. Convergence to the minimizer of the high-fidelity model can be guaranteed in limited situations, see [15, 172, 72] for details. 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, which we discuss in more detail now.

**5.3.1. 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 [46] for an introduction, theory, and implementation of these classical trust region methods. Early formulations of trust region ideas can be found in the papers by Levenberg [126] and Marquardt [136], with the papers by Powell [168, 169, 170] providing the important contribution of establishing and proving convergence of trust-region algorithms.

Classical trust-region methods use quadratic approximations of the objective in the trust region. The work [1] established a multifidelity trust region framework for more general low-fidelity models. In particular, [1] 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.

**5.3.2. Multifidelity trust-region algorithm.** The multifidelity trust-region approach of [1] is summarized in Algorithm 4. Inputs are the high-fidelity model  $f_{\text{hi}}$ , 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

---

**Algorithm 4** A multifidelity trust-region algorithm

---

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

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

- 5:     Find  $\mathbf{s}_i \in \mathcal{Z}$  that solves

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

- 6:     Update point

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

- 7:     Compare actual and estimated decrease in  $f_{\text{hi}}$

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

- 8:     Update step size

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

- 9:     **end for**
  - 10:    **return**  $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \dots$
  - 11: **end procedure**
- 

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 4 for the ease of exposition. In each iteration  $i = 1, 2, 3, \dots$ , a low-fidelity model  $f_{\text{lo}}^{(i)}$  is constructed so that it satisfies the first-order consistency requirement. Thus, the low-fidelity model output  $f_{\text{lo}}^{(i)}(\mathbf{z}_i) = f_{\text{hi}}(\mathbf{z}_i)$  equals the high-fidelity model output at the current point  $\mathbf{z}_i$ , and the gradient of the low-fidelity model  $\nabla f_{\text{lo}}^{(i)}(\mathbf{z}_i) = \nabla f_{\text{hi}}(\mathbf{z}_i)$  equals the gradient of the high-fidelity model. It is shown in [1] 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 [94]. Then, the step  $\mathbf{s}_i$  is computed and accepted if the point  $\mathbf{z}_i + \mathbf{s}_i$  decreases the high-fidelity model  $f_{\text{hi}}$  with respect to the current point  $\mathbf{z}_i$ , i.e., if  $f_{\text{hi}}(\mathbf{z}_i + \mathbf{s}_i) < f_{\text{hi}}(\mathbf{z}_i)$ . The size of the trust region is updated depending on the ratio of the actual decrease  $f_{\text{hi}}(\mathbf{z}_i) - f_{\text{hi}}(\mathbf{z}_i + \mathbf{s}_i)$  to the estimated decrease  $f_{\text{hi}}(\mathbf{z}_i) - f_{\text{lo}}^{(i)}(\mathbf{z}_i + \mathbf{s}_i)$ . The parameters  $\gamma_1$  and  $\gamma_2$  control

when to shrink and when to expand the trust region, respectively. The parameters  $\eta_1$  and  $\eta_2$  define the size of the contracted and expanded trust region, respectively. The algorithm returns the points  $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \dots$

**5.3.3. Generalized multifidelity trust-region methods.** In Algorithm 4, the low-fidelity model is corrected in each iteration. In [8, 66], 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 [20, 218], projection-based models are adapted within similar trust region frameworks. The work [217] 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. In [68], a correction model is used to correct low-fidelity model outputs in a trust-region model management optimization scheme. The correction model is constructed in the trust region and utilized to calculate additive and multiplicative adjustments to the low-fidelity model. The work [180] develops a trust-region multifidelity approach for combining low- and high-fidelity models that are defined over different design spaces. For example, the number of design variables used with the low-fidelity model can be different than the number of design variables with the high-fidelity model. Space mappings are derived that map between the design spaces. The space mappings are corrected to obtain a provably convergent method.

If gradients are unavailable, or too costly to approximate, then the framework developed by [1] relying on the first-order consistency of the low-fidelity model cannot be applied. In [134], 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 [47, 48] and the tailored radial basis function modeling of [212]. 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.

**5.4. Multifidelity optimization under uncertainty.** In their simplest forms, optimization under uncertainty formulations consider a similar problem to (5.1), 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. [150] uses the control variate-based multifidelity Monte Carlo method of Section 3.2 to derive a multifidelity optimization under uncertainty method that provides substantial reductions in computational cost using a variety of low-fidelity model types. In [151], 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.

**6. 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 [114]. 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 [219, 176] to statistical information criteria [129, 34] to quantifying the difference between experimental data and model outputs [158]. The probabilities are then used to fuse the model outputs with, e.g., Bayesian model averaging [124] or adjustment factors [176]. The fundamental work by Kennedy and O’Hagan [114] treats model inadequacy in a Bayesian setting and led to a series of papers on correcting model inadequacy and validating models [115, 100, 157, 18]. The work [18] introduces a six-step process to validate models. In [3], Bayesian estimation is employed to fuse model 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 [113] 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 and US Air Force Office of Research grant FA9550-15-1-0001.

#### REFERENCES

- [1] 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.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] D. G. Anderson. Iterative procedures for nonlinear integral equations. *J. ACM*, 12(4):547–560, Oct. 1965.
- [6] A. E. Annels. Geostatistical Ore-reserve Estimation. In *Mineral Deposit Evaluation*, pages 175–245. Springer Netherlands, 1991.
- [7] A. C. Antoulas. *Approximation of Large-scale Dynamical Systems*. SIAM, 2005.
- [8] 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.
- [9] W. E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics*, 9(1):17–29, 1951.
- [10] S. R. Arridge, J. P. Kaipio, V. Kolehmainen, M. Schweiger, E. Somersalo, T. Tarvainen, and M. Vauhkonen. Approximation errors and model reduction with an application in optical diffusion tomography. *Inverse Problems*, 22(1):175, 2006.
- [11] I. Babuška, F. Nobile, and R. Tempone. Worst case scenario analysis for elliptic problems with uncertainty. *Numerische Mathematik*, 101(2):185–219, 2005.
- [12] 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.
- [13] Z. Bai. Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems. *Applied Numerical Mathematics*, 43(1–2):9–44, 2002.
- [14] R. E. Bank, T. F. Dupont, and H. Yserentant. The hierarchical basis multigrid method. *Numerische Mathematik*, 52(4):427–458, 1988.
- [15] J. F. M. Barthelemy and R. T. Haftka. Approximation concepts for optimum structural design — a review. *Structural optimization*, 5(3):129–144, 1993.
- [16] 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.
- [17] 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.
- [18] M. J. Bayarri, J. O. Berger, R. Paulo, J. Sacks, J. A. Cafeo, J. Cavendish, C.-H. Lin, and J. Tu. A Framework for Validation of Computer Models. *Technometrics*, 49(2):138–154, May 2007.



- [19] 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.
- [20] 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.
- [21] 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.
- [22] 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.
- [23] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [24] 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.
- [25] 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.
- [26] 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.
- [27] J. H. Bramble, J. E. Pasciak, and J. Xu. Parallel multilevel preconditioners. *Mathematics of Computation*, 55(191):1–22, 1990.
- [28] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31:333–390, 1977.
- [29] P. Bratley, B. L. Fox, and L. E. Schrage. *A Guide to Simulation*. Springer, 1987.
- [30] W. Briggs, V. E. Henson, and S. F. McCormick. *A Multigrid Tutorial*. SIAM, 2000.
- [31] C. G. Broyden. A class of methods for solving nonlinear simultaneous equations. *Mathematics of Computation*, 19(92):577–593, 1965.
- [32] H.-J. Bungartz and M. Griebel. Sparse grids. *Acta Numerica*, 13:147–269, 2004.
- [33] 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.
- [34] K. Burnham and D. Anderson. *Model Selection and Multi-model Inference: A Practical Guide Information-theoretic Approach*. Springer, 2002.
- [35] Y. Cao, M. Gunzburger, X. Hu, F. Hua, X. Wang, and W. Zhao. Finite element approximations for stokes–darcy flow with beavers–joseph interface conditions. *SIAM Journal on Numerical Analysis*, 47(6):4239–4256, 2010.
- [36] Y. Cao, M. Gunzburger, F. Hua, and X. Wang. Analysis and finite element approximation of a coupled, continuum pipe-flow/darcy model for flow in porous media with embedded conduits. *Numerical Methods for Partial Differential Equations*, 27(5):1242–1252, 2011.
- [37] K. Carlberg. Adaptive h-refinement for reduced-order models. *International Journal for Numerical Methods in Engineering*, 102(5):1192–1210, 2015.
- [38] 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.
- [39] S. Chaturantabut and D. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.
- [40] A. Chaudhuri and R. T. Haftka. Efficient global optimization with adaptive target setting. *AIAA Journal*, 52(7):1573–1578, 2014.
- [41] 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.
- [42] S. Choi, J. J. Alonso, and I. M. Kroo. Two-Level Multifidelity Design Optimization Studies for Supersonic Jets. *Journal of Aircraft*, 46(3):776–790, May 2009.
- [43] S. Choi, J. J. Alonso, I. M. Kroo, and M. Wintzer. Multifidelity Design Optimization of Low-Boom Supersonic Jets. *Journal of Aircraft*, 45(1):106–118, Jan. 2008.
- [44] J. A. Christen and C. Fox. Markov chain Monte Carlo using an approximation. *Journal of Computational and Graphical Statistics*, 14(4):795–810, 2005.
- [45] 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.
- [46] A. Conn, N. Gould, and P. Toint. *Trust Region Methods*. MOS-SIAM Series on Optimization.

- Society for Industrial and Applied Mathematics, 2000.
- [47] 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.
  - [48] 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.
  - [49] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
  - [50] 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.
  - [51] 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.
  - [52] T. Cui, K. J. Law, and Y. M. Marzouk. Dimension-independent likelihood-informed MCMC. *Journal of Computational Physics*, 304:109 – 137, 2016.
  - [53] 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.
  - [54] 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.
  - [55] W. Dahmen and A. Kunoth. Multilevel preconditioning. *Numerische Mathematik*, 63(1):315–344, 1992.
  - [56] I. Daubechies. *Ten Lectures on Wavelets*. SIAM, 1992.
  - [57] W. C. Davidon. Variable metric method for minimization. *SIAM Journal on Optimization*, 1(1):1–17, 1991.
  - [58] 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.
  - [59] C. Dribusch, S. Missoum, and P. Beran. A multifidelity approach for the construction of explicit decision boundaries: application to aeroelasticity. *Structural and Multidisciplinary Optimization*, 42(5):693–705, Nov. 2010.
  - [60] Q. Du, V. Faber, and M. Gunzburger. Centroidal voronoi tessellations: Applications and algorithms. *SIAM Review*, 41(4):637–676, 1999.
  - [61] 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.
  - [62] 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.
  - [63] 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.
  - [64] M. S. Eldred, L. W. T. Ng, M. F. Barone, and S. P. Domino. Multifidelity uncertainty quantification using spectral stochastic discrepancy models. In R. Ghanem, D. Higdon, and H. Owhadi, editors, *Handbook of Uncertainty Quantification*, pages 1–45. Cham, 2016. Springer International Publishing.
  - [65] M. Emsermann and B. Simon. Improving simulation efficiency with quasi control variates. *Stochastic Models*, 18(3):425–448, 2002.
  - [66] 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.
  - [67] 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.
  - [68] C. C. Fischer, R. V. Grandhi, and P. S. Beran. Bayesian Low-Fidelity Correction Approach to Multi-Fidelity Aerospace Design. In *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA SciTech Forum. American Institute of Aeronautics and Astronautics, Jan. 2017.
  - [69] R. Fletcher and M. J. D. Powell. A rapidly convergent descent method for minimization. *The Computer Journal*, 6(2):163, 1963.

- [70] A. I. J. Forrester and A. J. Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1–3):50–79, 2009.
- [71] 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.
- [72] A. I. J. Forrester, A. Sobester, and A. J. Keane. *Engineering design via surrogate modelling: a practical guide*. Wiley, 2008.
- [73] 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.
- [74] R. W. Freund. Model reduction methods based on Krylov subspaces. *Acta Numerica*, 12:267–319, 2003.
- [75] 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.
- [76] M. Giles. Multi-level Monte Carlo path simulation. *Operations Research*, 56(3):607–617, 2008.
- [77] W. Gilks, S. Richardson, and D. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman & Hall, 1996.
- [78] W. R. Gilks, G. O. Roberts, and E. I. George. Adaptive Direction Sampling. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 43(1):179–189, 1994.
- [79] W. R. Gilks, G. O. Roberts, and S. K. Sahu. Adaptive Markov Chain Monte Carlo through Regeneration. *Journal of the American Statistical Association*, 93(443):1045–1054, Sept. 1998.
- [80] 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.
- [81] T. Goel, R. T. Haftka, W. Shyy, and N. V. Queipo. Ensemble of surrogates. *Structural and Multidisciplinary Optimization*, 33(3):199–216, 2007.
- [82] R. B. Gramacy, G. A. Gray, S. L. Digabel, H. K. H. Lee, P. Ranjan, G. Wells, and S. M. Wild. Modeling an Augmented Lagrangian for Blackbox Constrained Optimization. *Technometrics*, 58(1):1–11, Jan. 2016.
- [83] R. B. Gramacy and H. K. H. Lee. Bayesian Treed Gaussian Process Models With an Application to Computer Modeling. *Journal of the American Statistical Association*, 103(483):1119–1130, Sept. 2008.
- [84] R. B. Gramacy and H. K. H. Lee. Adaptive Design and Analysis of Supercomputer Experiments. *Technometrics*, 51(2):130–145, May 2009.
- [85] P. J. Green and A. Mira. Delayed Rejection in Reversible Jump Metropolis-Hastings. *Biometrika*, 88(4):1035–1053, 2001.
- [86] 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.
- [87] 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.
- [88] 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.
- [89] 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.
- [90] H. Haario, E. Saksman, and J. Tamminen. An adaptive Metropolis algorithm. *Bernoulli*, 7(2):223–242, 2001.
- [91] B. Haasdonk. Convergence rates of the POD-Greedy method. *ESAIM: Mathematical Modelling and Numerical Analysis*, 47:859–873, 2013.
- [92] B. Haasdonk and M. Ohlberger. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM: M2AN*, 42(2):277–302, 2008.
- [93] W. Hackbusch. *Multi-Grid Methods and Applications*. Springer, 1985.
- [94] R. T. Haftka. Combining global and local approximations. *AIAA Journal*, 29(9):1523–1525, 1991.
- [95] A.-L. Haji-Ali, F. Nobile, and R. Tempone. Multi-index Monte Carlo: when sparsity meets sampling. *Numerische Mathematik*, 132(4):767–806, 2015.
- [96] J. M. Hammersley and D. C. Handscomb. *Monte Carlo Methods*. Methuen London, 1964.
- [97] 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.
- [98] 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.
- [99] I. M. Held. The gap between simulation and understanding in climate modeling. *Bulletin of the American Meteorological Society*, 86(11):1609–1614, 2005.
- [100] D. Higdon, M. Kennedy, J. Cavendish, J. Cafoe, and R. Ryne. Combining Field Data and Computer Simulations for Calibration and Prediction. *SIAM Journal on Scientific Computing*, 26(2):448–466, Jan. 2004.
- [101] 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.
- [102] 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.
- [103] 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.
- [104] R. Hooke and T. A. Jeeves. “Direct search” solution of numerical and statistical problems. *Journal of the ACM*, 8(2):212–229, 1961.
- [105] J. Huttunen and J. P. Kaipio. Approximation errors in nonstationary inverse problems. *Inverse Problems and Imaging*, 1(1):77–93, Jan. 2007.
- [106] J. M. J. Huttunen and J. P. Kaipio. Approximation error analysis in nonlinear state estimation with an application to state-space identification. *Inverse Problems*, 23(5):2141, 2007.
- [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] J. Kaipio and E. Somersalo. *Statistical and computational inverse problems*. Springer, 2005.
- [111] J. Kaipio and E. Somersalo. Statistical inverse problems: Discretization, model reduction and inverse crimes. *Journal of Computational and Applied Mathematics*, 198(2):493–504, Jan. 2007.
- [112] A. J. Keane. Wing optimization using design of experiment, response surface, and data fusion methods. *Journal of Aircraft*, 40(4):741–750, 2003.
- [113] 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.
- [114] 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.
- [115] M. C. Kennedy, A. O’Hagan, and N. Higgins. Bayesian Analysis of Computer Code Outputs. In *Quantitative Methods for Current Environmental Issues*, pages 227–243. Springer, London, 2002.
- [116] D. E. Keyes. Exaflop/s: The why and the how. *Comptes Rendus Mécanique*, 339(2–3):70–77, 2011.
- [117] V. Kolehmainen, M. Schweiger, I. Nissilä, T. Tarvainen, S. R. Arridge, and J. P. Kaipio. Approximation Errors and Model Reduction in Three-Dimensional Diffuse Optical Tomography. In *Biomedical Optics and 3-D Imaging (2012)*, paper BTu3A.5, page BTu3A.5. Optical Society of America, Apr. 2012.
- [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] C. Lanczos. An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators. *Journal of Research of the National Bureau of Standards*, 45(4):255–282, 1950.
- [122] C. Lanczos. Solution of Systems of Linear Equations by Minimized Iterations. *Journal of Research of the National Bureau of Standards*, 49(1):33–53, 1952.
- [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] A. Lehtikoinen, S. Finsterle, A. Voutilainen, L. M. Heikkinen, M. Vauhkonen, and J. P. Kaipio. Approximation errors and truncation of computational domains with application to geophysical tomography. *Inverse Problems and Imaging*, 1(2):371–389, Apr. 2007.
- [126] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2:164–168, 1944.
- [127] 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.
- [128] J. Li and D. Xiu. Evaluation of failure probability via surrogate models. *Journal of Computational Physics*, 229(23):8966–8980, 2010.
- [129] W. A. Link and R. J. Barker. Model weights and the foundations of multimodel inference. *Ecology*, 87(10):2626–2635, 2006.
- [130] J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, 2008.
- [131] 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.
- [132] 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.
- [133] A. March. *Multifidelity methods for multidisciplinary system design*. Phd thesis, Massachusetts Institute of Technology, June 2012.
- [134] A. March and K. Willcox. Constrained multifidelity optimization using model calibration. *Structural and Multidisciplinary Optimization*, 46(1):93–109, 2012.
- [135] A. March and K. Willcox. Provably convergent multifidelity optimization algorithm not requiring high-fidelity derivatives. *AIAA Journal*, 50(5):1079–1089, 2012.
- [136] D. Marquardt. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, June 1963.
- [137] 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.
- [138] J. D. Martin and T. W. Simpson. Use of Kriging models to approximate deterministic computer models. *AIAA Journal*, 43(4):853–863, 2005.
- [139] 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.
- [140] Y. Marzouk and K. Willcox. Uncertainty quantification. In N. Higham, editor, *The Princeton Companion to Applied Mathematics*. Princeton University Press, 2015.
- [141] G. Matheron. Principles of geostatistics. *Economic Geology*, 58(8):1246–1266, 1963.
- [142] S. F. McCormick. Multigrid methods, 1987.
- [143] P. Mills. Efficient statistical classification of satellite measurements. *International Journal of Remote Sensing*, 32(21):6109–6132, 2011.
- [144] A. Mira. On Metropolis-Hastings algorithms with delayed rejection. *Metron - International Journal of Statistics*, LIX(3-4):231–241, 2001.
- [145] B. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, 26(1):17–32, 1981.
- [146] 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.
- [147] D. E. Myers. Matrix formulation of co-kriging. *Journal of the International Association for Mathematical Geology*, 14(3):249–257, 1982.
- [148] A. Narayan, C. Gittelsohn, and D. Xiu. A stochastic collocation algorithm with multifidelity models. *SIAM Journal on Scientific Computing*, 36(2):A495–A521, 2014.
- [149] B. L. Nelson. On control variate estimators. *Computers & Operations Research*, 14(3):219–225, 1987.
- [150] L. W. Ng and K. Willcox. Multifidelity approaches for optimization under uncertainty. *International Journal for Numerical Methods in Engineering*, 100(10):746–772, 2014.
- [151] 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.
- [152] A. Nissinen, L. M. Heikkinen, and J. P. Kaipio. The Bayesian approximation error approach for electrical impedance tomography—experimental results. *Measurement Science and*

- Technology*, 19(1):015501, 2008.
- [153] A. Nissinen, V. P. Kolehmainen, and J. P. Kaipio. Compensation of Modelling Errors Due to Unknown Domain Boundary in Electrical Impedance Tomography. *IEEE Transactions on Medical Imaging*, 30(2):231–242, Feb. 2011.
- [154] 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.
- [155] 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.
- [156] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, 2006.
- [157] J. Oakley and A. O’Hagan. Bayesian Inference for the Uncertainty Distribution of Computer Model Outputs. *Biometrika*, 89(4):769–784, 2002.
- [158] 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.
- [159] 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.
- [160] B. Peherstorfer, T. Cui, Y. Marzouk, and K. Willcox. Multifidelity importance sampling. *Computer Methods in Applied Mechanics and Engineering*, 300:490 – 509, 2016.
- [161] 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.
- [162] B. Peherstorfer and K. Willcox. Dynamic data-driven reduced-order models. *Computer Methods in Applied Mechanics and Engineering*, 291:21–41, 2015.
- [163] 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.
- [164] B. Peherstorfer, K. Willcox, and M. Gunzburger. Optimal model management for multifidelity monte carlo estimation. *SIAM Journal on Scientific Computing*, 38(5):A3163–A3194, 2016.
- [165] 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.
- [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] M. J. D. Powell. A fortran subroutine for solving systems of nonlinear algebraic equations. In *Numerical Methods for Nonlinear Algebraic Equations*. Gordon and Breach, 1970.
- [169] M. J. D. Powell. A new algorithm for unconstrained optimization. In *Nonlinear Programming*. Academic press, 1970.
- [170] M. J. D. Powell. Convergence properties of a class of minimization algorithms. In *Nonlinear Programming 2*. Academic press, 1975.
- [171] 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.
- [172] 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.
- [173] D. Rajnarayan. *Trading Risk and Performance for Engineering Design Optimization using Multifidelity Analyses*. Phd thesis, Stanford University, 2009.
- [174] C. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.
- [175] M. Rathinam and L. Petzold. A new look at proper orthogonal decomposition. *SIAM Journal on Numerical Analysis*, 41(5):1893–1925, 2003.
- [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] T. D. Robinson, M. S. Eldred, K. E. Willcox, and R. Haimes. Surrogate-Based Optimization Using Multifidelity Models with Variable Parameterization and Corrected Space Mapping. *AIAA Journal*, 46(11):2814–2822, Nov. 2008.

- [181] 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.
- [182] 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.
- [183] 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.
- [184] Y. Saad and M. Schultz. GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, July 1986.
- [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] A. M. Stuart. Inverse problems: A Bayesian perspective. *Acta Numerica*, 19:451–559, 2010.
- [193] M. A. Taddy, H. K. H. Lee, G. A. Gray, and J. D. Griffin. Bayesian Guided Pattern Search for Robust Local Optimization. *Technometrics*, 51(4):389–401, Nov. 2009.
- [194] A. Tarantola. *Inverse Problem Theory*. Elsevier, 1987.
- [195] A. Tarantola and B. Valette. Inverse problems = Quest for information. *Journal of Geophysics*, pages 159–170, 1982.
- [196] T. Tarvainen, V. Kolehmainen, J. P. Kaipio, and S. R. Arridge. Corrections to linear methods for diffuse optical tomography using approximation error modelling. *Biomedical Optics Express*, 1(1):209–222, July 2010.
- [197] T. Tarvainen, V. Kolehmainen, A. Pulkkinen, M. Vauhkonen, M. Schweiger, S. R. Arridge, and J. P. Kaipio. An approximation error approach for compensating for modelling errors between the radiative transfer equation and the diffusion approximation in diffuse optical tomography. *Inverse Problems*, 26(1):015005, 2010.
- [198] 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.
- [199] 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.
- [200] L. Tierney and A. Mira. Some adaptive Monte Carlo methods for Bayesian inference. *Statistics in Medicine*, 18(17-18):2507–2515, Sept. 1999.
- [201] V. Torczon. On the convergence of pattern search algorithms. *SIAM Journal on Optimization*, 7(1):1–25, 1997.
- [202] A. Toth and C. T. Kelley. Convergence analysis for anderson acceleration. *SIAM Journal on Numerical Analysis*, 53(2):805–819, 2015.
- [203] B. Tracey, D. Wolpert, and J. J. Alonso. Using supervised learning to improve Monte-Carlo integral estimation. *AIAA Journal*, 51(8):2015–2023, 2013.
- [204] U. Trottenberg, C. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, 2001.
- [205] 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.
- [206] K. Urban and A. T. Patera. A new error bound for reduced basis approximation of parabolic partial differential equations. *Comptes Rendus Mathématique*, 350(3–4):203 – 207, 2012.
- [207] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.
- [208] G. Venter, R. T. Haftka, and J. H. Starnes. Construction of response surface approximations for design optimization. *AIAA Journal*, 36(12):2242–2249, 1998.

- [209] 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.
- [210] 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.
- [211] H. F. Walker and P. Ni. Anderson acceleration for fixed-point iterations. *SIAM Journal on Numerical Analysis*, 49(4):1715–1735, 2011.
- [212] 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.
- [213] R. L. Winkler. Combining probability distributions from dependent information sources. *Management Science*, 27(4):479–488, 1981.
- [214] W. Yamazaki and D. J. Mavriplis. Derivative-Enhanced Variable Fidelity Surrogate Modeling for Aerodynamic Functions. *AIAA Journal*, 51(1):126–137, Nov. 2012.
- [215] 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.
- [216] H. Yserentant. Hierarchical bases give conjugate gradient type methods a multigrid speed of convergence. *Applied Mathematics and Computation*, 19(1-4):347–358, 1986.
- [217] Y. Yue and K. Meerbergen. Accelerating optimization of parametric linear systems by model order reduction. *SIAM Journal on Optimization*, 23(2):1344–1370, 2013.
- [218] 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.
- [219] 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.