Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

Adaptive experimental design for multi-fidelity surrogate modeling of multi-disciplinary systems

J.D. Jakeman*[1] | S. Friedman[2] | M. Eldred[1] | L. Tamellini[3] | A.A. Gorodetsky[4] | D. Allaire[2]

[1]Optimization and Uncertainty Quantification, Sandia National Laboratories, Albuquerque, NM, 87123

[2]J. Mike Walker '66 Department of Mechanical Engineering, Texas A&M University, College Station, TX, USA

[3]Consiglio Nazionale delle Ricerche, Istituto di Matematica Applicata e Tecnologie Informatiche "E. Magenes"(CNR-IMATI), Via Ferrata 1, 27100, Pavia, Italy

[4]University of Michigan, 3053 FXB, 1320 Beal Avenue, Ann Arbor, MI. 48109, USA

Correspondence
*J.D. Jakeman, Optimization and Uncertainty Quantification, Sandia National Laboratories, Albuquerque, NM, 87123.
Email: jdjakem@sandia.gov

Abstract

We present an adaptive algorithm for constructing surrogate models of multi-disciplinary systems composed of a set of coupled components. With this goal we introduce 'coupling' variables with *a priori* unknown distributions that allow surrogates of each component to be built independently. Once built, the surrogates of the components are combined to form an integrated-surrogate that can be used to predict system-level quantities of interest (QoI) at a fraction of the cost of the original model. The error in the integrated-surrogate is greedily minimized using an experimental design procedure that allocates the amount of training data, used to construct each component-surrogate, based on the contribution of those surrogates to the error of the integrated-surrogate. The multi-fidelity procedure presented is a generalization of Multi-Index Stochastic Collocation (MISC) that can leverage ensembles of models of varying cost and accuracy, for one or more components, to reduce the computational cost of constructing the integrated-surrogate. Extensive numerical results demonstrate that, for a fixed computational budget, our algorithm is able to produce surrogates that are orders of magnitude more accurate than methods that treat the integrated system as a black-box.

KEYWORDS:
Uncertainty quantification, integrated systems, surrogate, experimental design, dimension reduction, multi-disciplinary, multi-physics, multi-fidelity

1 | INTRODUCTION

Modeling complex systems often involves integrating numerous components from multiple disciplines. The components of the system can be coupled by either feed-forward or feed-back coupling. For a single evaluation of the system inputs, feed-forward coupling requires passing the outputs of upstream components to downstream components, whereas feedback coupling requires relaxation methods, such as fixed point iteration (FPI), to determine the component outputs that are interdependent. Consequently, outer-loop problems such as uncertainty quantification and design, which require repeated interrogation of the coupled system, can be intractable when one or more component-models are computationally expensive to simulate.

Surrogate methods, such as polynomial chaos [1,2,3], Gaussian processes [4,5,6], low-rank decompositions [7,8,9], sparse grid interpolation [10,11,12], reduced basis approximations [13,14,15,16] and neural networks [17,18] have all been used successfully to reduce the cost of analyzing computationally expensive models. However, these methods can be inefficient when applied to integrated systems because they treat the system-model as a black-box and do not exploit the coupling structure linking components.

Several recent works [19,20,21,22,23,24] have demonstrated the benefits of exploiting the structure of coupled systems for reducing the cost of outer-loop applications. These methods decompose system analysis into analyses of individual components that are then combined to make system-level predictions. Decoupling of the integrated system is achieved by expressing each of the K component-models of a system as a function of both exogeneous inputs $\boldsymbol{z}$ controlled by the user/modeler, e.g. random or design variables, and inputs $\boldsymbol{\xi}$ that we call coupling variables, whose values are determined by the outputs $\boldsymbol{y}$ of the other components (Figure 1). Two classes of approaches are then used to interrogate each component and combine the evaluations to make predictions of the integrated system.

The first class of methods build a single surrogate that maps all the system inputs $\boldsymbol{z}$ to all the coupling variables $\boldsymbol{\xi}$ [25,26,27]. The training data used to build the surrogate is obtained by evaluating the coupled system at realizations of the exogeneous variables and collecting the values of the coupling variables computed by fixed point iteration during each simulation. Once constructed, the surrogate of the coupling variables removes the need to use FPI when evaluating a multi-disciplinary model with feedback coupling. This can substantially reduce the cost of predicting system level outputs, but gains are ultimately limited because the expensive component-models still must be evaluated using values of the coupling variables obtained from the surrogate.

The second class of approaches builds multiple surrogates, each one approximating a map from the local inputs of a component $(\boldsymbol{z}_k, \boldsymbol{\xi}_k)$ to the local component outputs $\boldsymbol{y}_k$, $k = 1, \ldots, K$. The training data used to build each surrogate is obtained via independent evaluations of the associated component-model and does not require evaluation of the coupled system. Once constructed, the inexpensive component-surrogates are used in place of the original expensive numerical component-models when evaluating the multi-disciplinary system. For example, [28,29] build surrogates of each component in a feed-forward system consisting of a chain of one-directional couplings and pass the outputs from an upstream component-surrogate to the next downstream component-surrogate. Such so called integrated-surrogates have also been used in a similar fashion for systems with feedback coupling [19,21,23]. Unlike the first class of methods, the second class of methods still require FPI to determine the value of the coupling variables when feedback-coupling is present, however FPI is only performed using the surrogates at negligible cost.

Class-two methods are typically more efficient than black box (and even class-one) approaches because: they construct several low-dimensional surrogate models of the system components instead of a single high-dimensional surrogate of a black-box system; they can take advantage of simpler mappings from component inputs to outputs which can be less non-linear than the coupled system map, which is the composition of the component maps; and unlike the first class of methods, relaxation methods are never applied to the original expensive component-models. However, the accuracy of class two integrated-surrogates is heavily dependent on the amount of training data used to train each component-surrogate because not all components impact the prediction of system quantities of interest (QoI) equally. Thus, experimental design strategies are needed to reduce the error in each component-surrogate commensurate with its impact on the accuracy of system QoI predictions. But to date, experimental design algorithms have only been developed for refining class-two surrogates of systems consisting of a chain of purely feed-forward couplings [29].

In this work, we propose a novel adaptive surrogate and experimental design strategy for building class-two system surrogates, based on Multi-Index Stochastic Collocation (MISC) [30,31,32,33,34], which can be used for systems with either, or both, feed-forward and feedback coupling. Every iteration of the sequential algorithm greedily generates candidate training data from the single component predicted to produce the greatest change in the integrated-surrogate, relative to the cost incurred by evaluating the candidate data set. Optimizing the investment in the constituent components significantly reduces the cumulative computational cost of building surrogates of each component, which, to the authors' knowledge, has never been demonstrated for class-one or class-two methods. Indeed, such an approach is not even possible when building class-one surrogates because they require evaluations of the integrated system to generate training data. However, building surrogates of component outputs typically requires defining ranges for the coupling variables which are not known *a priori*. This limits the practical application of class-two methods. To address this challenge we embed an iterative procedure to estimate the ranges of the coupling variables within our experimental design algorithm. Our approach significantly improves the efficiency of building component-surrogates when compared to procedures that use conservative estimates of the coupling variable ranges.

The algorithm proposed in this paper significantly reduces the cost of building surrogates for integrated system-models when one model is available for each component. These gains are further amplified when a selection of simulators of varying fidelity and computational cost are available for one or more of these system components. In such situations, our experimental design algorithm enriches a small number of high-fidelity simulations with larger numbers of simulations from models of lower accuracy and cost, to enable greater exploration and resolution of uncertainty while maintaining deterministic prediction accuracy. Our method for using multiple models of varying fidelity to increase the accuracy of integrated component-surrogates is the first of its kind.

The single-fidelity version of the method we propose, and the preceding works we cited, possess similarities to domain decomposition[35,36,37,38] and localized model reduction (LMR)[39,40,41,42]. These methods efficiently solve partial differential equations (PDEs) by solving independent local problems on subdomains and computing a global solution via an appropriate coupling of the subdomains; LMR is domain decomposition technique that uses a localized reduced basis in each subdomain. In contrast to domain decomposition which is used to approximate the entire solution of a PDE and is able to set the number of subdomains (components) to reduce computational cost, our approach targets estimation of a small number of QoI of a multi-disciplinary system, with a fixed number of components, which may or may not involve the solving PDEs.

The remainder of this paper is organized as follows. Section 2 discusses the procedures used to evaluate an integrated system of coupled components. In Section 3 we discuss how to use surrogates of each component to predict system-level QoI and the approximation error this induces. Section 4 presents a greedy experimental design procedure that minimizes the error in the surrogate of each component in a manner that minimizes error in predictions of system-level QoI for a fixed budget. Finally, the efficacy of the proposed approach is demonstrated using a number of numerical examples in Section 5 and conclusions are presented in Section 6.

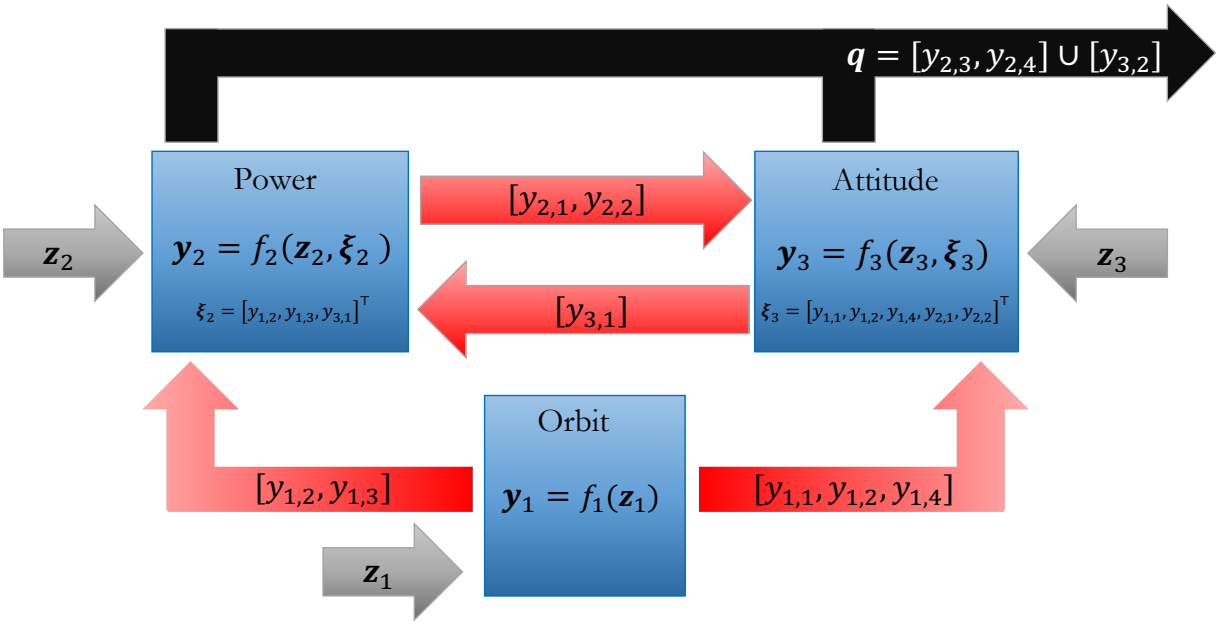

FIGURE 1 A fire detection satellite system consisting of three components. Coupling variables are depicted in red, external inputs in gray and system-level QoI in black. Here $\boldsymbol{z}_1 \cap \boldsymbol{z}_2 = \emptyset$, $\boldsymbol{z}_2 \cap \boldsymbol{z}_3 = [z_4]$, $\boldsymbol{z}_1 \cap \boldsymbol{z}_3 = [z_1]$ and $\boldsymbol{\xi}_2 \cap \boldsymbol{\xi}_3 = [y_{1,2}]$.

2 | EVALUATING COUPLED SYSTEMS

2.1 | Problem formulation

This paper is concerned with efficiently predicting QoI obtained from models of integrated systems with coupled components. With this goal, let

$$\boldsymbol{y} = f(\boldsymbol{z}) : \Gamma \to \Upsilon \tag{1}$$

denote the map from exogeneous parameters $\boldsymbol{z} = [z_1, \dots, z_D]^\top \in \Gamma \subseteq \mathbb{R}^D$ with probability density (PDF) $\rho_z(\boldsymbol{z})$, to a set of Q outputs $\boldsymbol{y} = [y_1, \dots, y_Q]^\top \in \Upsilon \subseteq \mathbb{R}^Q$, where the QoI $\boldsymbol{q} \subseteq \boldsymbol{y}$, $\boldsymbol{q} \in \mathbb{R}^{Q^{sys}}$, we wish to predict are a subset of the system-outputs.

The so-called system-model (1) consists of K component-models of the form

$$\boldsymbol{y}_k = f_k(\boldsymbol{z}_k, \boldsymbol{\xi}_k) : \Gamma_k \times \Xi_k \to \Upsilon_k, \quad k = 1, \dots, K. \tag{2}$$

Each component-model returns a vector of Q_k outputs $\boldsymbol{y}_k = [y_{k,1}, \dots, y_{k,Q_k}]^\top \in \Upsilon_k \subseteq \mathbb{R}^{Q_k}$, $\Upsilon_k \subseteq \Upsilon$, and is a function of a vector of D_k exogeneous random variables $\boldsymbol{z}_k \in \Gamma_k \subseteq \mathbb{R}^{D_k}$, with support $\Gamma_k \subseteq \Gamma$ and joint PDF $\rho_{\boldsymbol{z},k}$ and of a set of S_k coupling random variables $\boldsymbol{\xi}_k \in \Xi_k \subseteq \mathbb{R}^{S_k}$, with support $\Xi_k \subseteq \Upsilon$, and joint (unknown) PDF $\rho_{\boldsymbol{\xi},k}$, which are a subset of the outputs produced by other components $\bigcup_{j=1,j\neq k}^{K} \boldsymbol{y}_j$. In the following, it will sometimes be useful to refer to the inputs of the k-th component without distinction between exogenous and coupling variables. To this end, we further introduce the notation

$$\boldsymbol{u}_k := [\boldsymbol{z}_k^\top, \boldsymbol{\xi}_k^\top]^\top \in \Gamma_k \times \Xi_k \subset \mathbb{R}^{D_k+S_k} \tag{3}$$

to denote the concatenation of $\boldsymbol{z}_k$ and $\boldsymbol{\xi}_k$, so that the k-th component-model in (2) can be compactly rewritten as

$$\boldsymbol{y}_k = f_k(\boldsymbol{u}_k).$$

We will refer to $\boldsymbol{u}_k$ as the parameters of the component-model. The notation for the aforementioned quantities and all that follow is summarized in Appendix B.

Figure 1 graphically depicts an example of a multi-disciplinary system comprised of coupled components. The system outputs are $\boldsymbol{y} = [y_{1,1}, y_{1,2}, y_{1,3}, y_{1,4}, y_{2,1}, y_{2,2}, y_{2,3}, y_{2,4}, y_{3,1}, y_{3,2}]^\top$, where the first index denotes the component and the second index denotes the QoI from that component. Three of these outputs are QoI, specifically $\boldsymbol{q} = [y_{2,3}, y_{2,4}, y_{3,2}]^\top$, such that $Q^{\text{sys}} = 3$. Components $k = 1, 2, 3$ have 2, 2 and 6 exogeneous variables respectively, specifically $\boldsymbol{z}_1 = [z_1, z_2]^\top$, $\boldsymbol{z}_2 = [z_3, z_4]^\top$, $\boldsymbol{z}_3 = [z_1, z_4, z_5, z_6, z_7, z_8]^\top$. Some exogeneous system-model variables $\boldsymbol{z}$ are unique to a single component and others are shared between components so that $D \leq \sum_{k=1}^{K} D_k$. For example, components 1 and 2 share no common exogeneous variables, i.e. $\boldsymbol{z}_1 \cap \boldsymbol{z}_2 = \emptyset$, whereas components 2 and 3 share one common exogeneous variable, i.e. $\boldsymbol{z}_2 \cap \boldsymbol{z}_3 = [z_4]$, as do components 1 and 3, i.e. $\boldsymbol{z}_1 \cap \boldsymbol{z}_3 = [z_1]$.

For general systems, the coupling variables $\boldsymbol{\xi}_k$ of the k-th component are determined by subsets of the system-outputs $\boldsymbol{y}$ of connected component models. In Figure 1, the subset $[y_{1,2}]$ of the outputs of the first component is used to provide input, in the form of feed-forward coupling to the second and third components, such that $\boldsymbol{\xi}_2 \cap \boldsymbol{\xi}_3 = [y_{1,2}]$. Feedback coupling exists between components 2 and 3, indicated by the fact that some outputs of component 2 are inputs to component 3 and vice-versa. In summary, $\boldsymbol{\xi}_1 = [\emptyset]$ (i.e., component 1 has no incoming coupling variables), $\boldsymbol{\xi}_2 = [y_{1,2}, y_{1,3}, y_{3,1}]^\top$, and $\boldsymbol{\xi}_3 = [y_{1,1}, y_{1,2}, y_{1,4}, y_{2,1}, y_{2,2}]^\top$.

Following[24] we use extraction matrices to encode the relationships between the inputs, outputs, and coupling variables of the component-models. Specifically, the exogeneous variables, coupling variables, and outputs of the k-th component satisfy

$$\boldsymbol{z}_k = \boldsymbol{A}_k^z \boldsymbol{z} \qquad \boldsymbol{\xi}_k = \boldsymbol{A}_k^\xi \boldsymbol{y}, \qquad \boldsymbol{y}_k = \boldsymbol{A}_k^y \boldsymbol{y},$$

where $\boldsymbol{A}_k^z \in \mathbb{R}^{D_k \times D}$, $\boldsymbol{A}_k^\xi \in \mathbb{R}^{S_k \times Q}$, and $\boldsymbol{A}_k^y \in \mathbb{R}^{Q_k \times Q}$ consist of unit row vectors that select a subset of entries from the vectors they are applied to. Similarly, we extract these system-level QoI via

$$\boldsymbol{q} = \boldsymbol{A}^q \boldsymbol{y} \in \mathbb{R}^{Q^{\text{sys}}} \qquad \boldsymbol{A}^q \in \mathbb{R}^{Q^{\text{sys}} \times Q}. \tag{4}$$

Letting $\boldsymbol{e}_{i,j} = [0, \dots, 0, 1, 0 \dots, 0]^\top$ denote the unit vector of length i with the j-th entry equal to 1, the extraction matrices of the multi-disciplinary system in Figure 1 are

$$\boldsymbol{A}_1^z = \begin{bmatrix} \boldsymbol{e}_{D,1}^\top \\ \boldsymbol{e}_{D,2}^\top \end{bmatrix} \qquad \boldsymbol{A}_2^z = \begin{bmatrix} \boldsymbol{e}_{D,3}^\top \\ \boldsymbol{e}_{D,4}^\top \end{bmatrix} \qquad \boldsymbol{A}_3^z = \begin{bmatrix} \boldsymbol{e}_{D,1}^\top \\ \boldsymbol{e}_{D,4}^\top \\ \boldsymbol{e}_{D,5}^\top \\ \boldsymbol{e}_{D,6}^\top \\ \boldsymbol{e}_{D,7}^\top \\ \boldsymbol{e}_{D,8}^\top \end{bmatrix} \qquad \boldsymbol{A}_1^\xi = \begin{bmatrix} \emptyset \end{bmatrix} \qquad \boldsymbol{A}_2^\xi = \begin{bmatrix} \boldsymbol{e}_{Q,2}^\top \\ \boldsymbol{e}_{Q,3}^\top \\ \boldsymbol{e}_{Q,9}^\top \end{bmatrix} \qquad \boldsymbol{A}_3^\xi = \begin{bmatrix} \boldsymbol{e}_{Q,1}^\top \\ \boldsymbol{e}_{Q,2}^\top \\ \boldsymbol{e}_{Q,4}^\top \\ \boldsymbol{e}_{Q,5}^\top \\ \boldsymbol{e}_{Q,6}^\top \end{bmatrix}$$

$$\boldsymbol{A}_1^y = \begin{bmatrix} \boldsymbol{e}_{Q,1}^\top \\ \boldsymbol{e}_{Q,2}^\top \\ \boldsymbol{e}_{Q,3}^\top \\ \boldsymbol{e}_{Q,4}^\top \end{bmatrix} \qquad \boldsymbol{A}_2^y = \begin{bmatrix} \boldsymbol{e}_{Q,5}^\top \\ \boldsymbol{e}_{Q,6}^\top \\ \boldsymbol{e}_{Q,7}^\top \\ \boldsymbol{e}_{Q,8}^\top \end{bmatrix} \qquad \boldsymbol{A}_3^y = \begin{bmatrix} \boldsymbol{e}_{Q,9}^\top \\ \boldsymbol{e}_{Q,10}^\top \end{bmatrix} \qquad \boldsymbol{A}^q = \begin{bmatrix} \boldsymbol{e}_{Q,7}^\top \\ \boldsymbol{e}_{Q,8}^\top \\ \boldsymbol{e}_{Q,10}^\top \end{bmatrix}$$

2.2 | Evaluating systems of components

Different approaches are needed to combine components linked by feed-forward coupling and those linked by feedback coupling. In this section we review the approaches we employ.

2.2.1 | Feed-forward coupling

Feed-forward coupling refers to the situation when the output(s) of a component are input(s) to another component; the coupling between components 1 and 2 in Figure 1 is an example of such a coupling. Without loss of generality, consider feed-forward coupling between two components coupled in the following way:

$$\boldsymbol{y}_k = f_k(\boldsymbol{z}_k, \boldsymbol{\xi}_k) \qquad \boldsymbol{\xi}_k = \boldsymbol{y}_{k-1} = f_{k-1}(\boldsymbol{z}_{k-1}),$$

such that the output of f_{k-1} is input to f_k. To evaluate the output of the k-th component at a sample $\boldsymbol{z}$, we simply evaluate f_{k-1} at $\boldsymbol{z}_{k-1} \subseteq \boldsymbol{z}$ and then evaluate f_k using the values $\boldsymbol{y}_{k-1}$ along with $\boldsymbol{z}_k \subseteq \boldsymbol{z}$. This procedure can naturally be extended to a chain of components, that is when $f = f_K \circ f_{K-1} \circ \cdots \circ f_1$. It is common for multiple components to be inputs to another component. For these general situations we pass information through the system of components by traversing a directed acyclic graph.

2.2.2 | Feedback coupling

Without loss of generality, consider two components with feedback coupling

$$\begin{cases} \boldsymbol{y}_j = f_j(\boldsymbol{z}_j, \boldsymbol{\xi}_j), & \boldsymbol{\xi}_j = \boldsymbol{y}_k \\ \boldsymbol{y}_k = f_k(\boldsymbol{z}_k, \boldsymbol{\xi}_k), & \boldsymbol{\xi}_k = \boldsymbol{y}_j. \end{cases} \tag{5}$$

The coupling between components 2 and 3 in Figure 1 is an example of feedback coupling, where in (5) we have for simplicity ignored any dependencies on any feed-forward coupling variables. To solve this system of non-linear equations we use fixed-point iteration (FPI). For a given realization of the random variables $\boldsymbol{z}$, FPI iteratively finds the values of the coupling variables that produce consistent solutions[43]. Using the iteration function

$$F(\boldsymbol{\xi}) = \begin{bmatrix} f_j(\boldsymbol{z}_j, \boldsymbol{\xi}_j) \\ f_k(\boldsymbol{z}_k, \boldsymbol{\xi}_k) \end{bmatrix} \qquad \boldsymbol{\xi} = [\boldsymbol{\xi}_j, \boldsymbol{\xi}_k]^\top \tag{6}$$

and starting from an initial guess $\boldsymbol{\xi}^0$ we evaluate

$$\boldsymbol{\xi}^p = F(\boldsymbol{\xi}^{p-1})$$

until $\|\boldsymbol{\xi}^p - \boldsymbol{\xi}^{p-1}\| < \eta$, for some accuracy tolerance $\eta \geq 0$. In this paper we assume that the iterating function F in (6) is a contraction, which guarantees convergence of FPI[43].

When a system consists of both feed-forward and feedback coupling, we proceed by partitioning the components into groups that, when considered together as a single "macro-component", transform the system-model into a purely feed-forward system; FPI is needed to exchange information within a subgroup. Such system grouping can be achieved using methods such as Design Manager's Aid for Intelligent Decomposition (DeMAID)[44]. For the system depicted in Figure 1, there are two groups: one containing model 1 and the other containing models 2 and 3, for which we feed the output of group 1 to the second group and then use FPI to determine the remaining coupling variables.

3 | INTEGRATED-SURROGATES OF COUPLED SYSTEMS

The goal of this paper is to present a method for designing the computer experiments need to construct a surrogate of the coupled system-model in (1). With this goal, we seek an approximation (surrogate model) of each component-model (2) in the system, with explicit functional dependence on the coupling variables. Once built, these surrogates can replace the true component-models when evaluating the system using the strategies presented in Sections 2.2.1 and 2.2.2. We refer to the resulting approximation as an integrated-surrogate.

Gaussian processes[29,28,45] and polynomial chaos expansions[46] have been used to generate decoupled surrogates in the past. In this work, we choose instead to use an approach based upon adaptive versions[30,33] of MISC[32,31,34], because it provides the features necessary to develop an experimental design strategy for allocating resources to components of integrated systems. The details of our specific algorithm are presented in Section 4, but first, in this section we discuss important considerations that impact the accuracy of predictions made using surrogates of components.

3.1 | Multi-fidelity modeling

For many practical applications, a number of viable models of varying cost and accuracy may be available to simulate each component in an integrated system. In this paper, we assume each component can be simulated using a numerical model that approximates the solution of some governing equations for a given *fixed* $\boldsymbol{z}$ and $\boldsymbol{\xi}$. We also assume that this model has a set of hyper-parameters — mesh size, time step, maximum number of iterations, convergence tolerance, etc. — that can be used to simulate the component with varying accuracy and cost. Changing the values of these hyper-parameters produces simulations of *varying fidelities (resolution)* and computational cost. We refer to approaches that leverage only one model or solver setting as single-fidelity methods, and approaches that leverage multiple models and settings as multi-fidelity methods.

Formally, we assume that each component $k = 1 \ldots, K$ of a coupled system has R_k hyper-parameters and introduce the multi-index $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_{R_k}] \in \mathbb{N}^{R_k}$ to distinguish between the different model fidelities of the k-th component, which we denote

$$f_{k,\boldsymbol{\alpha}}(\boldsymbol{z}, \boldsymbol{\xi}) \approx f_k(\boldsymbol{z}, \boldsymbol{\xi}).$$

The entries $\alpha_i \in \mathbb{N}$ are integer values that dictate the value of each hyper-parameter; for example, a time-step size proportional to $2^{-\alpha_1}$ and a mesh discretization proportional to $2^{-\alpha_2}$. In the following we assume that as the entries of $\boldsymbol{\alpha}$ increase, the model fidelity increases and the error in the successive approximations of f_k decreases, i.e. $\|f_{k,\boldsymbol{\alpha}^\star} - f_k\| \leq \|f_{k,\boldsymbol{\alpha}} - f_k\|$ in some suitable norm if $\boldsymbol{\alpha}^\star \geq \boldsymbol{\alpha}$.[1]

To provide further intuition on the role of the multi-index $\boldsymbol{\alpha}$, consider a model that simulates heat transfer within a cooled turbine in the path of heated gas flow using a finite element model (FEM); we use this model as a component in a system-model presented in Section 5.3. For this model, we use a single hyper-parameter that dictates the mesh resolution used to solve the governing equations, i.e. $R_1 = 1$, $\boldsymbol{\alpha} = [\alpha_1]$. Three meshes of increasing resolution are available, thus $\alpha_1 \in \{1, 2, 3\}$. The FEM solution on the coarsest mesh is plotted in Figure 2. Here the subscript 1 of R_1 is used because the heat-transfer model is the first component of the system. The computational cost of evaluating the heat transfer model is dependent on the number of degrees of freedom used by the FEM. The number of degrees of freedom and cost (in seconds) is presented in Table 1.

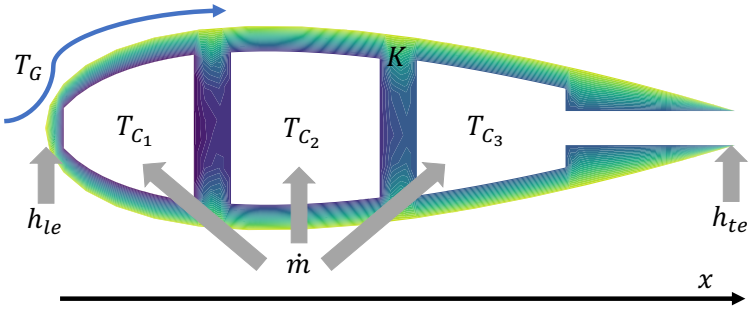

FIGURE 2 Finite element solution and parameterization of the turbine component-model.

TABLE 1 The computational cost (seconds) of solving the heat transfer model for varying discretizations.

α_1	1	2	3
Cost (sec)	0.26388454	1.1500591	4.41993904
DOF	4998	17435	66549

3.2 | Surrogate modeling

In this paper we construct surrogates of each component of a system-model so as to reduce the cost of repeated interrogation of the system. Recalling the compact notation for the inputs of the k-th component in (3), we denote the single-fidelity surrogate of the fidelity at level $\boldsymbol{\alpha}$ of the k-th component by

$$f_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}(\boldsymbol{u}_k) \approx f_{k,\boldsymbol{\alpha}}(\boldsymbol{u}_k), \tag{7}$$

where the multi-index $\boldsymbol{\beta}$ controls the number of samples used to construct the surrogate and thus its computational cost and accuracy. Given a specified model fidelity $\boldsymbol{\alpha}$ and surrogate fidelity $\boldsymbol{\beta}$, we construct this surrogate using a set of $M_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$ samples

[1] $\boldsymbol{\alpha}^\star \geq \boldsymbol{\alpha}$ if $\exists j$ s.t. $\boldsymbol{\alpha}_j^\star > \boldsymbol{\alpha}_j$ and $\boldsymbol{\alpha}_i^\star = \boldsymbol{\alpha}_i$ for $i \neq j$.

of $\boldsymbol{u}_k$, denoted by $\mathcal{U}_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]} = \{\boldsymbol{u}_k^{(m)}\}_{m=1}^{M_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}}$, and evaluations $\mathcal{Y}_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]} = \{f_{k,\boldsymbol{\alpha}}(\boldsymbol{u}_k^{(m)})\}_{m=1}^{M_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}}$ at those samples. In the following we assume $\boldsymbol{\beta} \in \mathbb{N}^{N_k}$, $N_k := D_k + S_k$ so that, each entry of $\boldsymbol{\beta}$ specifies the density of samples allocated to each dimension of $\boldsymbol{u}_k$.

The total error in the surrogate $f_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}(\boldsymbol{u}_k)$ of a component-model f_k can be decomposed into two components

$$\|f_k - f_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}\| \le \|f_k - f_{k,\boldsymbol{\alpha}}\| + \|f_{k,\boldsymbol{\alpha}} - f_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}\|. \tag{8}$$

The first term on the right-hand side represents the so-called deterministic error and quantifies the discretization error introduced by the numerical model used to solve the governing equations of the component-model for any fixed value of the parameters. The second term on the right represents the parametric error which quantifies the error of approximating the numerical model of the governing equations with a surrogate intended for fast evaluation of the governing equation at different values of the parameters. The previous inequality implies that a cost-effective experimental design strategy must balance these two sources of error. Simply fixing the fidelity of a numerical model *a priori* , as often done in the literature, is inefficient.

Figure 3 depicts the impact of parameteric and deterministic errors for a simple example. The surrogates (dotted black) approximate the true function (solid red) with different numbers of evaluations of either a low-fidelity model $f_{k,[1]}$ or a high-fidelity model $f_{k,[2]}$ (both dashed blue). As β increases, we add two additional training data (i.e., evaluations of $f_{k,\alpha}$), which allows the surrogate $f_{k,[\alpha,\beta]}$ to more accurately approximate $f_{k,\alpha}$ and thus the parametric error to decrease. When $\alpha = 1, \beta = 3$ the surrogate $f_{k,[1,3]}$ approximates $f_{k,1}$ well, but does not approximate f_k as accurately, that is the deterministic error dominates. In comparison, when $\alpha = 2, \beta = 1$ the surrogate $f_{k,[2,1]}$ is also poor, but this time it is because the parametric error dominates.

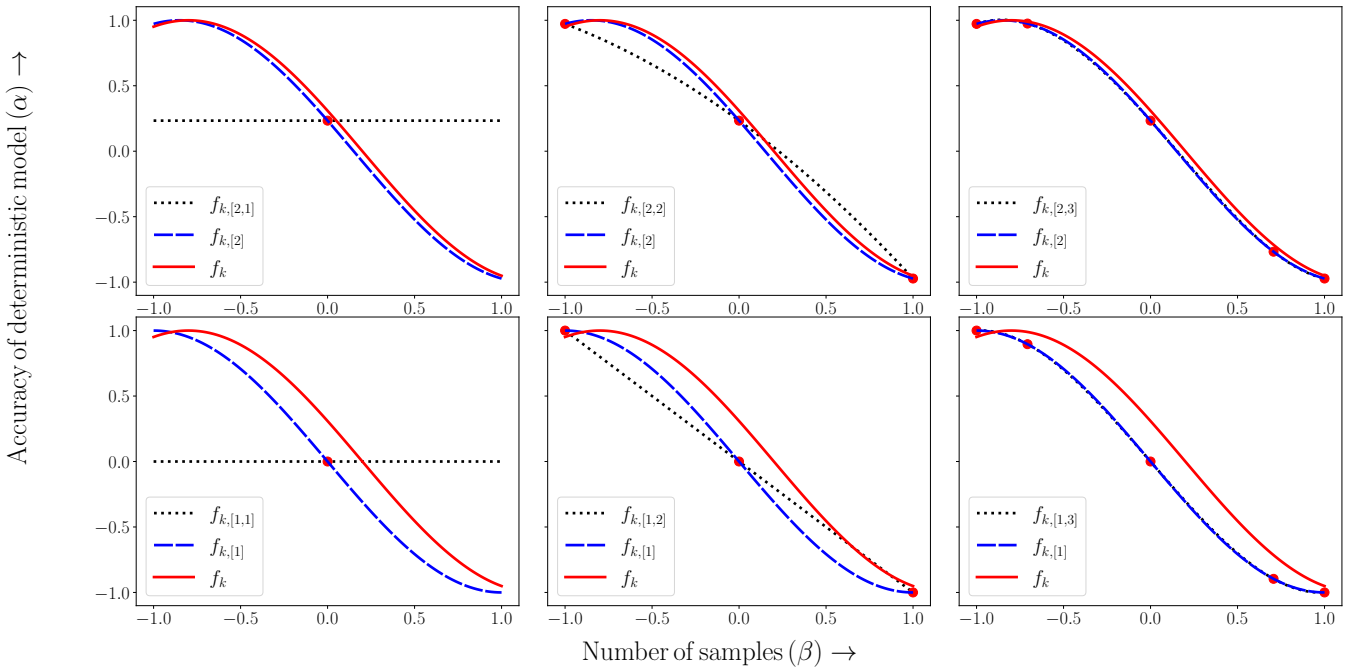

FIGURE 3 Approximations $f_{k,[\alpha,\beta]}$ (dotted black) of the one-dimensional function $f_k^{\epsilon}(\boldsymbol{u}) = \cos(\frac{1}{2}\pi(u_1 + \epsilon) + \frac{2}{5}\pi)$. The true model (solid red) corresponds to $\epsilon = 0$, and the two fidelities $f_\alpha, \alpha \in \{1, 2\}$, (both plotted in dashed blue) are obtained by setting $\epsilon = 0.2$ (low-fidelity f_1) and $\epsilon = 0.05$ (high-fidelity f_2). Red dots depict samples used to build the interpolants. The cost of constructing $f_{k,[\alpha,\beta]}$ increases with α and β.

Ideally we would use the approximation $f_{k,[2,3]}$ in the top right panel; however, that surrogate uses numerous high-fidelity model evaluations, that are typically more expensive than lower-fidelity evaluations. Cost-effective experimental design strategies for constructing a component-surrogate are therefore needed to balance the parameteric and deterministic errors. With this goal, in Section 4 we propose a strategy that combines multiple surrogates $f_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$ of each component built using differing numbers of evaluations and fidelities (e.g. $f_{k,[1,3]}$ and $f_{k,[2,2]}$). The number of samples, that is the different $\boldsymbol{\beta}$, used to build the surrogates of a given model fidelity is dependent on the predictive utility of each model fidelity, dictated by $\boldsymbol{\alpha}$, relative to the

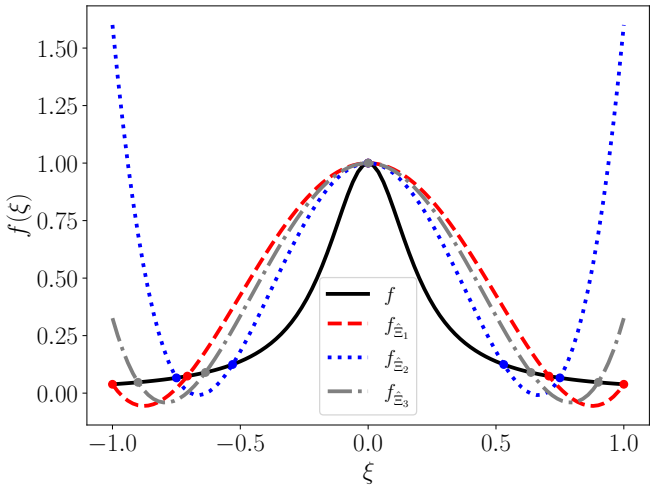

FIGURE 4 Lagrange polynomial interpolants of the Runge function $f(\xi) = (1 + 25\xi^2)^{-1}$ constructed using five function evaluations on different intervals $\hat{\Xi}_1 = [-1.0, 1.0]$, $\hat{\Xi}_2 = [-0.75, 0.75]$, and $\hat{\Xi}_3 = [-0.9, 0.9]$.

cost of evaluating the model. Typically, less samples are assigned to higher model fidelities, that is $\boldsymbol{\alpha}$ with larger entries. To facilitate the use of multi-fideltiy approaches for building component-models, moving forward we will denote the surrogate of a component by

$$f_{k,\mathcal{I}_k}(\boldsymbol{u}_k) \approx f_k(\boldsymbol{u}_k)$$

where $\mathcal{I}_k$ is a set of concatenated multi-indices $[\boldsymbol{\alpha}, \boldsymbol{\beta}]$.

3.3 | Characterizing the coupling variables

Constructing a surrogate $f_{k,\mathcal{I}_k}(\boldsymbol{u}_k)$ of a component requires specifying the ranges of the coupling variables. The coupling variables $\boldsymbol{\xi}_k$ are functions of the exogeneous variables $\boldsymbol{z}$ (either explicitly or via their dependence on other components) and are thus themselves random, but their distribution are unknown prior to simulation. Consequently, following a procedure proposed for black-box models[47,48], we construct an approximation of each component of the system utilizing a prior distribution $\nu(\boldsymbol{\xi}_k)$ defined over an estimated range $\hat{\Xi}_k$.

The following lemma characterizes the accuracy of a surrogate model in a ω-weighted norm, associated with the correct unknown distribution of the coupling variables, when the surrogate is built minimizing a prior ν-weighted norm.

Lemma 1 (Strong convergence[48])**.** Let $\nu : \hat{\Xi} \to \mathbb{R}$ and $\omega : \Xi \to \mathbb{R}$ denote two densities which satisfy

$$\delta = 1 - \int_{\Xi \cap \hat{\Xi}} \omega(\boldsymbol{u}) \mathrm{d}\boldsymbol{u}.$$

Given an approximation f_ν of f with approximation error ϵ, i.e.,

$$\epsilon := \|f - f_\nu\|_{L^p_\nu(\Xi)}, \quad p \geq 1, \tag{9}$$

then, if f is bounded with $C_f = \|f\|_{L^\infty(\Xi)}$, it holds that

$$\|f - f_\nu\|_{L^p_\omega(\Xi)} \leq C_r^{1/p}\epsilon + C_f\delta^{1/p}, \quad \text{provided} \quad C_r := \max_{\boldsymbol{u} \in \Xi \cup \hat{\Xi}} \frac{\omega(\boldsymbol{u})}{\nu(\boldsymbol{u})} < \infty. \tag{10}$$

The second term in (10) comes from truncating the tails of the true distribution of the coupling variables. In many cases, the coupling variables are bounded and so this term can be eliminated by using conservative estimates of the range. For unbounded domains, the tail truncation error can be made arbitrarily small by choosing a sufficiently large range $\hat{\Xi}_k$.

Provided the simpler ν is chosen to be non-zero wherever the original ω is non-zero, such that $\delta = 0$, Lemma 1 suggests that shape of the distribution ν used for the coupling variables does not affect the rate at which the error converges in a component-surrogate. Consequently, in the following we set ν to be the PDF of the uniform distribution over a pre-defined range $\hat{\Xi}_k$. For

some integrated systems, the ranges can be determined from analysis of the system components. However, for other systems, the ranges of the coupling variables must be estimated. Figure 4 demonstrates the importance of correctly estimating the range; underestimating the range, such that $\delta > 0$ can lead to large approximation errors outside $\hat{\Xi}$.

In this paper, we use an adaptive algorithm, presented in Section 4.6, to estimate the range of the coupling variables. We investigate the performance of this algorithm and the impact of over-estimating and under-estimating the range of the coupling variables in Section 5.1.1.

3.4 | Error analysis of integrated-surrogates

Once surrogates of each component have been constructed, they can be combined to make predictions of system-level QoI using the procedures discussed in Section 2. We denote the integrated-surrogate

$$\boldsymbol{y}_{\mathcal{J}} = f_{\mathcal{J}}(\boldsymbol{z}) \approx f(\boldsymbol{z}).$$

Here $\mathcal{J} = \{\mathcal{I}_1, \dots, \mathcal{I}_K\}$ are the index sets associated with each component-surrogates $f_{k,\mathcal{I}_k}$ $k = 1, \dots, K$. The accuracy of the integrated-surrogate's prediction of the system-level QoI $\boldsymbol{q}_{\mathcal{J}} = \boldsymbol{A}^q \boldsymbol{y}_{\mathcal{J}}$ depends on the accuracy of each component-surrogate. In this section, we provide theoretical bounds on the error for systems involving feed-forward and feedback coupling. For simplicity, we consider errors in systems with only single-fidelity component-models.

3.4.1 | Feed-forward coupling

For a system comprising a chain of feed-forward couplings (Section 2.2.1), the error in the system-level approximation is given by the following proposition. The proof of this proposition is given in Appendix A.

Proposition 1 (Feed-forward surrogate error)**.** Assume that each component $f_k(\boldsymbol{z}, \boldsymbol{\xi}_k)$ is Lipschitz continuous with respect to the coupling variables $\boldsymbol{\xi}_k$ with uniform Lipschitz constant L_k for all $\boldsymbol{\xi}_k \in \Xi_k$. Furthermore, let $f_{k,q}(\boldsymbol{z}, \boldsymbol{\xi}_k)$ denote the q-th output of the k-th component and let $f_{k,\mathcal{I}_k,q}(\boldsymbol{z}, \boldsymbol{\xi}_k)$ denote the associated surrogate output. If the trained surrogates satisfy $\|f_{k,q}(\boldsymbol{z}, \boldsymbol{\xi}_k) - f_{k,\mathcal{I}_k,q}(\boldsymbol{z}, \boldsymbol{\xi}_k)\|_{L^\infty(\Gamma)} \le \epsilon_k \forall q = 1, \dots, Q_k$, then we have for $f = f_K \circ f_{K-1} \circ \cdots \circ f_1$ and $f_{\mathcal{J}} = f_{K,\mathcal{I}_K} \circ f_{K-1,\mathcal{I}_{K-1}} \circ \cdots \circ f_{1,\mathcal{I}_1}$ that

$$\sup_{\boldsymbol{z}\in\Gamma} \max_{q=1,\dots,Q^{sys}} |f_q(\boldsymbol{z}) - f_{\mathcal{J},q}(\boldsymbol{z})| \le \epsilon \frac{1 - L^K}{1 - L},$$

where $L = \max_{k=1,\dots,K} L_k$ and $\epsilon = \max_{k=1,\dots,K} \epsilon_k$.

This result can be applied to any directed acyclic graph structure by applying the proposition to each branch of the graph and setting ϵ_k to be the largest of the errors in the surrogates for all upstream components providing inputs to the component under consideration. Lastly, note that ϵ_k can include both the deterministic and parametric errors of the k-th component-surrogate.

3.4.2 | Feedback coupling

By recognizing that FPI (Section 2.2.2) can be formulated as the composition of F from (6) with itself and that the coupling variables $\boldsymbol{\xi}(\boldsymbol{z})$ are subsets of the system-model outputs $f(\boldsymbol{z})$, we can modify the proof of Proposition 1 (see Appendix A) to guarantee convergence and to arrive at the following result.

Proposition 2 (Feedback surrogate error)**.** Let L be the Lipschitz constant of the fixed point iteration function (6), then under the assumptions of Proposition 1, the error in the coupling variables $\boldsymbol{\xi}^P$ after P fixed-point iterations, when approximated using the surrogate values $\boldsymbol{\xi}^P_{\mathcal{J}}$, satisfies

$$\sup_{\boldsymbol{z}\in\Gamma} \max_{q=1,\dots,Q} |\xi^P_q(\boldsymbol{z}) - \xi^P_{\mathcal{J},q}(\boldsymbol{z})| \le \epsilon \frac{1 - L^P}{1 - L},$$

where

$$\epsilon = \max_{k=1,\dots,K} \sup_{\boldsymbol{z}_k,\boldsymbol{\xi}_k} \max_{q=1,\dots,Q_k} |f_{k,q}(\boldsymbol{z}_k, \boldsymbol{\xi}_k) - f_{k,\mathcal{I}_k,q}(\boldsymbol{z}_k, \boldsymbol{\xi}_k)|$$

is the worst-case error over all outputs of all component-surrogates.

4 | GREEDY EXPERIMENTAL DESIGN FOR INTEGRATED-SURROGATES

Proposition 1 shows that the error in predictions of system-level QoI obtained using surrogates for each component can be decomposed into errors proportional to the errors in each component-surrogate. This section introduces a greedy algorithm that allocates resources to components and their varying fidelities in a manner that is commensurate with their impact on the system-level QoI $\boldsymbol{q}$. Pseudocode for this novel algorithm is presented in Algorithm 1.

We formally discuss each step of Algorithm 1 in the following subsections, but first we first provide a conceptual overview here. With this goal, consider Figure 5 which depicts two steps of Algorithm 1 applied to to a purely feed-forward system comprised of two algebraic component-models. The two component-models $f_1(z) = z\sin(\pi z)$ and $f_2(\xi_2) = 1/(1 + 25\xi_2^2)$ are coupled via $\xi_2 = f_1(z)$ so that the system-model is given by $f(z) = f_2(f_1(z_1))$; single-fidelity models are used for the component-models solely to facilitate visualization.

Given a set of component-models, Algorithm 1 is initialized by building constant (degree-zero polynomial) surrogates of each component; this initialization procedure is discussed in Section 4.2. and is performed on Line 2 of Algorithm 1. The accuracy of each component-surrogate is then improved by sequentially incrementing the amount of training data used. Figure 5 depicts the second and third iteration of Algorithm 1 after initialization. Each iteration begins with a current and a refined surrogate of each component-model; the construction of the surrogates is discussed in Section 4.1. The models and surrogates of the first and second components are depicted in the first (from left) and second columns, respectively. The solid-red lines represent the component-models, the black-dashed lines represent the current surrogates constructed using the data depicted by the black discs, and the refined surrogates, trained using the data depicted by the cyan or lime-green crosses, are depicted by the dotted-blue and dash-dotted-green lines.

When the component-surrogates are coupled together they produce the integrated-surrogates depicted in the third column. In this plot, the dashed-black line is produced by coupling the current surrogates of each component, the dotted-blue line is obtained by coupling the refined surrogate of the first component with the current surrogate of the second, and the dashed-dotted-green is constructed by coupling the current surrogate of the first component with the refined surrogate of the second component. No training data are plotted in this column because no evaluations of the true coupled system are required.

Once the integrated surrogates have been constructed, Line 8 of Algorithm 1 estimates the contribution of the error in each component-surrogate to the error in the integrated-surrogate. This is achieved by measuring the difference between the new-integrated-surrogates and the current integrated surrogate, that is between the dotted-blue and dash-dotted-green lines and the black-dashed depicted in the third column, respectively. This estimation procedure is discussed in Section 4.4. Estimating error, based on changes to the integrated surrogate, ensures the algorithm adds data to each component-surrogate based upon its impact on predictions of system-level QoI.

Once errors have been estimated, the component with the largest estimated error (shaded with a blue background) is identified (Line 5 of Algorithm 1) and then refined (Line 6). The refinement procedure is discussed in detail in Section 4.3, but in summary it collects the data needed to update the current and new surrogates of the component selected for refinement. For example in the first step (first-row), the second component is selected for refinement, consequently in the next step (second row) the current surrogate of component two is now built using three points (instead of one) and the new surrogate of that component is built with five points (instead of three). In contrast, the current and new component-surrogates of the first component remain the same. However, the contribution of the first component-surrogate to the error in the integrated surrogate changes because the down-stream component has changed. This leads to the first component being selected for refinement in the second step and is the reason that the error contributions (indicators) of all components is re-estimated every iteration on Line 8.

In the following sections we provide a more detailed discussion of each step of Algorithm 1, referring to Figure 5 when appropriate. Note the right most column of this figure have not yet been discussed because they are specific to the implementation details introduced below.

4.1 | Multi-index stochastic collocation (MISC)

The ability to accurately construct an integrated-surrogate depends on the accuracy of the surrogates of each component. MISC provides an effective mechanism to create a multi-fidelity surrogate of each component f_k by combining evaluations of varying fidelity $f_{k,\boldsymbol{\alpha}}$.[30] More specifically, MISC approximates each component-model as a linear combination of multiple surrogates

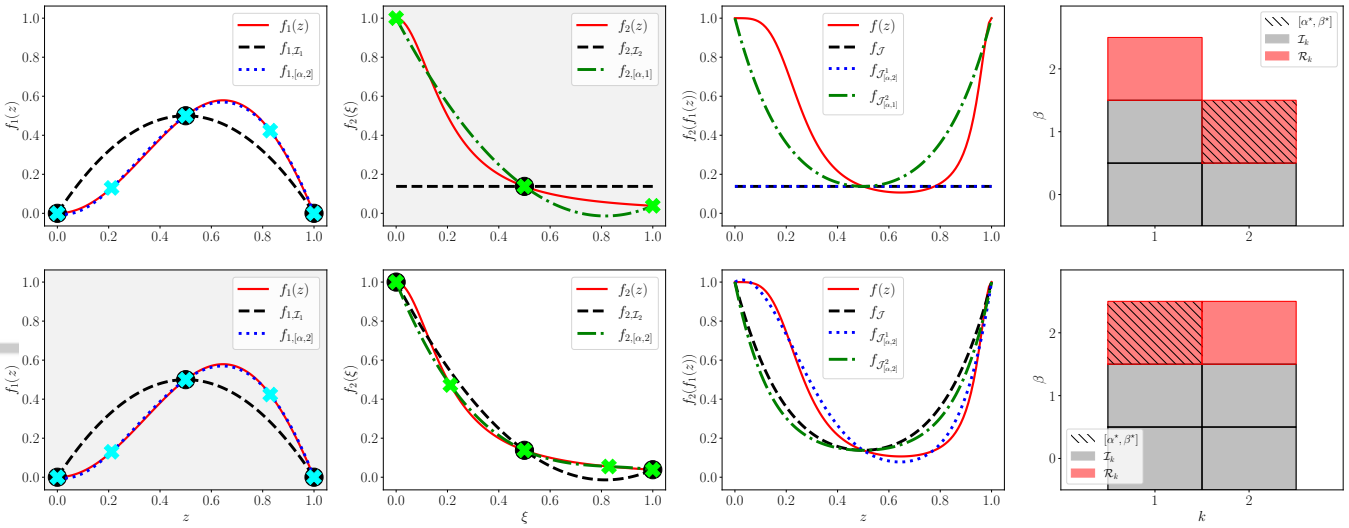

FIGURE 5 Two steps of Algorithm 1 applied to a system with two component-models $f_1(z) = z\sin(\pi z)$, $f_2(\xi_1) = 1/(1+25\xi_1^2)$. From left to right, each row depicts: (first panel) the first component-model f_1, the current component-surrogate $f_{1,\mathcal{I}_1}$, and a new surrogate $f_{1,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$ that can be used to improve $f_{1,\mathcal{I}_1}$; (second panel) the second component-model f_2, the current component-surrogate $f_{2,\mathcal{I}_2}$, and a new surrogate $f_{2,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$ that can be used to improve $f_{2,\mathcal{I}_2}$; (third panel) the system model f, the current integrated surrogate $f_{\mathcal{J}}$, and the new integrated surrogates that are obtained by updating either the first $f_{\mathcal{J}^1_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}}$ or second component-surrogate $f_{\mathcal{J}^2_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}}$; and (final panel) the index sets defining the integrated surrogates. Here $\mathcal{J}^j_{[\boldsymbol{\alpha},\boldsymbol{\beta}]} := \{\mathcal{I}_k\}_{k=1,k\neq j}^K \cup \left(\mathcal{I}_j \cup \{[\boldsymbol{\alpha},\boldsymbol{\beta}]\}\right)$. Discs and crosses represent training data used to respectively build the current and refined component-surrogates. Gray boxes represent the indices in the sets $\mathcal{I}_k$ and gray boxes reprsent the set $\mathcal{R}_k$ of possible indices $[\boldsymbol{\alpha},\boldsymbol{\beta}]$ to add to the component-surrogate. The striped box represents the index $[\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star]$ with the largest error indicator. Gray shaded panels indicate that the associated component-surrogate has been chosen for refinement. Because each component only has a single fidelity model the index $\boldsymbol{\alpha}$ is redundant but we keep it here for consistency with our more general multi-fidelity formulation. Also note that the dotted-blue and dashed-black lines are identical in the third panel of the first row.

Algorithm 1 CONSTRUCT_COMPONENT_SURROGATES$[\{\mathcal{F}_k\}_{k=1}^K, \tau, W_{\max}, \{\hat{\Xi}_k\}_{k=1}^K, \rho_z] \rightarrow f_{\mathcal{J}}$

for $k = 1,\dots,K$ **do**
  $\mathcal{I}_k, \mathcal{R}_k, \mathcal{E}_k, \mathcal{U}_k, \mathcal{Y}_k, \mathcal{C}_k$:= INITIALIZE_SURROGATE$(\mathcal{F}_k, \hat{\Xi}_k, \rho_z)$ ▷ Initialize k-th component-surrogates
end for
while not TERMINATE$[\{\mathcal{I}_k\}_{k=1}^K, \tau, W_{\max}]$ **do**
  $\{l, [\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star]\} := \operatorname{argmax}_{k\in[1,K],[\boldsymbol{\alpha},\boldsymbol{\beta}]\in\mathcal{R}_k} \gamma_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$ ▷ Find component with largest error indicator $\gamma_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$
  $\mathcal{I}_l, \mathcal{R}_l, \mathcal{E}_l, \mathcal{U}_l, \mathcal{Y}_l, \mathcal{C}_k$:= REFINE$[[\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star], \mathcal{I}_l, \mathcal{R}_l, \mathcal{E}_l, \mathcal{U}_l, \mathcal{Y}_l, \mathcal{C}_k, \mathcal{F}_l, \hat{\Xi}_l, \rho_z]$
  for $j = 1,\dots,K$ **do** ▷ Recompute indicators of candidate indices of all K components
    $\mathcal{E}_j, \{\hat{\Xi}_k\}_{k=1}^K$:= COMPUTE_ERROR_INDICATORS$[\{\mathcal{I}_k\}_{k=1}^K, \mathcal{R}_j, \mathcal{E}_j, \{\mathcal{U}_k\}_{k=1}^K, \{\mathcal{Y}_k\}_{k=1}^K, \{\hat{\Xi}_k\}_{k=1}^K, \rho_z]$
  end for
end while

$f_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$, that is

$$f_k(\boldsymbol{u}_k) \approx f_{k,\mathcal{I}_k}(\boldsymbol{u}_k) = \sum_{[\boldsymbol{\alpha},\boldsymbol{\beta}]\in\mathcal{I}_k} c_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}\, f_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}(\boldsymbol{u}_k). \tag{11}$$

Formula (11) uses multiple model fidelities $f_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$ to balance deterministic and parametric errors (see (8)) and to reduce the computational cost of achieving a specified level of accuracy. The accuracy of the MISC approximation (11) is dictated by the

set $\mathcal{I}_k \subset \mathbb{N}^{R_k+N_k}$, which contains multiple concatenated multi-indices $[\boldsymbol{\alpha}, \boldsymbol{\beta}]$. The index set $\mathcal{I}_k$ must be downward-closed, that is

$$\text{if } [\boldsymbol{\gamma}, \boldsymbol{\delta}] \leq [\boldsymbol{\alpha}, \boldsymbol{\beta}] \text{ and } [\boldsymbol{\alpha}, \boldsymbol{\beta}] \in \mathcal{I}_k \Rightarrow [\boldsymbol{\gamma}, \boldsymbol{\delta}] \in \mathcal{I}_k. \tag{12}$$

Under this assumption, the coefficients $c_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$ are given by the so-called combination technique formula[2]

$$c_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]} = \sum_{\substack{[\boldsymbol{i},\boldsymbol{j}]\in\{0,1\}^{R_k+N_k} \\ [\boldsymbol{\alpha}+\boldsymbol{i},\boldsymbol{\beta}+\boldsymbol{j}]\in\mathcal{I}_k}} (-1)^{\|[\boldsymbol{i},\boldsymbol{j}]\|_1}. \tag{13}$$

We use a greedy procedure to construct the index-sets $\mathcal{I}_k$, which is outlined in Sections 4.3 and 4.4. In the remainder of this section we discuss construction of the individual $f_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}(\boldsymbol{u}_k)$.

The single fidelity surrogates $f_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}(\boldsymbol{u}_k)$ are tensor-product Lagrangian interpolants, constructed using evaluations of the $\boldsymbol{\alpha}$-fidelity model $f_{k,\boldsymbol{\alpha}}(\boldsymbol{u})$ on a Cartesian grid defined on the parametric domain $\Gamma_k \times \Xi_k$. The sizes of the univariate sets of coordinates used to construct the grid are prescribed by the components of the multi-index $\boldsymbol{\beta}$. The use of a multi-index $\boldsymbol{\beta}$, instead of a scalar, allows the MISC approximation to exploit anisotropy in a component-model by assigning different numbers of samples to resolve each dimension commensurate with the sensitivity of the function to each dimension. Letting $m : \mathbb{N} \to \mathbb{N}$ be an increasing function, used for all K components, we construct a set of $m(\beta_n)$ univariate interpolation points

$$\mathcal{U}_{k,n,\boldsymbol{\alpha},\beta_n} = \{u^{(j)}_{k,n,\boldsymbol{\alpha},\beta_n}\}_{j=1}^{m(\beta_n)}, \tag{14}$$

and build $m(\beta_n)$ univariate Lagrange polynomials that are given by

$$\mathcal{L}^{(j)}_{k,n,\boldsymbol{\alpha},\beta_n}(u_{k,n}) = \prod_{l=1,l\neq j}^{m(\beta_n)} \frac{u_{k,n} - u^{(l)}_{k,n,\boldsymbol{\alpha},\beta_n}}{u^{(j)}_{k,n,\boldsymbol{\alpha},\beta_n} - u^{(l)}_{k,n,\boldsymbol{\alpha},\beta_n}}, \quad j = 1, \dots, m(\beta_n).$$

Note $\boldsymbol{\alpha}$ does not influence the number or locations of the points in these univariate sets, but we annotate these sets with this multi-index to make clear different sets can be used for different fidelities.

In the following we use univariate weighted Leja sequences[49,50] tailored to the probability distribution function of $u_{k,n}$ as univariate interpolation points in (14). Leja sequences $\mathcal{U}_{k,n,\boldsymbol{\alpha},\beta_n}$ are nested, that is, $\mathcal{U}_{k,n,\boldsymbol{\alpha},\beta_n} \subset \mathcal{U}_{k,n,\boldsymbol{\alpha},\beta_n^\star}$ if $\beta_n < \beta_n^\star$. We also define $m(\beta) = 2\beta + 1$ in (14) and set the maximum level of the univariate sequence to $\beta = 15$. For more details on the construction of Leja sequences, refer to Section 4.6. .

Given a set of univariate Leja sequences, we construct a multivariate set of samples of taking the Cartesian product of the univariate sets $\mathcal{U}_{k,n,\boldsymbol{\alpha},\beta_n}$, yielding

$$\mathcal{U}_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]} = \times_{n=1}^{N_k} \mathcal{U}_{k,n,\boldsymbol{\alpha},\beta_n} := \{\boldsymbol{u}^{(j)}_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}\}_{\boldsymbol{j}\leq m(\boldsymbol{\beta})}, \tag{15}$$

which consists of $M_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]} = \prod_{n=1}^{N_k} m(\beta_n)$ points $\boldsymbol{u}^{(\boldsymbol{j})}_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]} = [u^{(j_1)}_{k,1,\boldsymbol{\alpha},\beta_1}, u^{(j_2)}_{k,2,\boldsymbol{\alpha},\beta_2}, \cdots, u^{(j_{N_k})}_{k,N_k,\boldsymbol{\alpha},\beta_{N_k}}]$, where $m(\boldsymbol{\beta}) = [m(\beta_1), m(\beta_2), \cdots, m(\beta_{N_k}))]$. For each point in $\mathcal{U}_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$ we construct a multi-variate Lagrange polynomial via

$$\mathcal{L}^{(\boldsymbol{j})}_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}(\boldsymbol{u}_k) = \prod_{n=1}^{N_k} \mathcal{L}^{(j_n)}_{k,n,\boldsymbol{\alpha},\beta_n}(u_{k,n}), \quad \boldsymbol{j} \leq m(\boldsymbol{\beta}),$$

and finally define the tensor-product interpolant appearing in (11) as

$$f_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}(\boldsymbol{u}_k) = \sum_{\boldsymbol{j}\leq m(\boldsymbol{\beta})} f_{k,\boldsymbol{\alpha}}(\boldsymbol{u}^{(\boldsymbol{j})}_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}) \mathcal{L}^{(\boldsymbol{j})}_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}(\boldsymbol{u}_k).$$

Two different MISC surrogates, $f_{1,\mathcal{I}_1}$ ($\mathcal{I}_1 = [0,0], [0,1]$) and $f_{1,\mathcal{I}_1\cup\{[\boldsymbol{\alpha},2]\}}$, are shown in each of the upper-left panel of Figure 5. The Leja sequences used to build these interpolants are depicted by discs and crosses, respectively.

Note that for univariate MISC approximations, such as those shown, the combination coefficients (13) of all but the largest β become zero (this is typically not true for higher dimensional functions), such that $f_{k,\mathcal{I}_k}$ is simply the most accurate tensor-product interpolant, for example in the top left panel $f_{1,\mathcal{I}_1} = f_{1,[\boldsymbol{\alpha},2]}$, where $\boldsymbol{\alpha}$ is redundant because we are only considering single fidelity functions in this example. Also note that, while we use tensor product interpolants for our single-fidelity surrogates $f_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$ the MISC approximation (11) uses a combination of these tensor products, which when selected judicously via specification of the index $\mathcal{I}_k$, does not suffer the curse of dimensionality.

[2] $\{0,1\}^s$ is the set of s-dimensional vectors containing all combinations of zero and one and $\|\boldsymbol{\delta}\|_1 = \sum_i |\delta_i|$

4.2 | Initialization

Now that the form of the component-surrogates has been defined, we can discuss each step of Algorithm 1. Letting $\mathcal{F}_k$ denote the set of all model fidelities $f_{k,\boldsymbol{\alpha}}$ for the k-th component, Algorithm 1 takes as input, the set $\{\mathcal{F}_k\}_{k=1}^K$ containing all the model fidelities used to model each component, a maximum computational budget $W_{\max}$, a desired error tolerance τ, estimated ranges for the coupling variables of each component $\{\hat{\Xi}_k\}_{k=1}^K$, and the joint probability density ρ_z of the exogeneous variables $\boldsymbol{z}$. The algorithm then begins by initializing a set of surrogates $f_{k,\mathcal{I}_k}$ for each component in the system. The initialization routine is summarized in Algorithm 2.

The first step of the initialization algorithm sets $\mathcal{I}_k = \emptyset$. The second step specifies a set $\mathcal{R}_k$ of possible indices $[\boldsymbol{\alpha}, \boldsymbol{\beta}]$ to add to the component-surrogate. This set grows as the algorithm advances, but initially only contains a single index $[\boldsymbol{\alpha}, \boldsymbol{\beta}] := [1, 1, \cdots]$. The next two steps construct the set $\mathcal{U}_k$ containing the training points used to build the interpolants corresponding to the indices in both $\mathcal{I}_k$ and $\mathcal{R}_k$ and the set of evaluations $\mathcal{Y}_k$ at the points in $\mathcal{U}_k$. Having set $[\boldsymbol{\alpha}, \boldsymbol{\beta}] := [1, 1, \cdots]$ implies that the initial sets of training data $\mathcal{U}_k, \mathcal{Y}_k$ consist solely of the lowest fidelity model for each component evaluated at the center of $\Gamma_k \times \hat{\Xi}_k$, i.e., the midpoint of the ranges of $\boldsymbol{u}_k$. Step 6 defines the set $\mathcal{C}_k$ containing the combinations coefficients $c_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$ of the MISC approximation, see (13). This set is empty because $\mathcal{I}_k$ is empty. In the last step, we define the set $\mathcal{E}_k$ that contains error indicators $\gamma_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$ estimating the reduction in error of the system-level QoI obtained by adding each candidate index in $\mathcal{R}_k$. During initialization, we artificially set the posteriori error indicators $\gamma_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$ in the set $\mathcal{E}_k$ to be infinitely large to ensure that the initial index $[1, 1, \cdots]$ is added to $\mathcal{I}_k$ in the refinement step we describe next.[3]

Algorithm 2 `INITIALIZE_SURROGATE`$[\mathcal{F}_k, \hat{\Xi}_k, \rho_z] \rightarrow \mathcal{I}_k, \mathcal{R}_k, \mathcal{E}_k, \mathcal{U}_k, \mathcal{Y}_k, \mathcal{C}_k$

1: $\mathcal{I}_k := \emptyset$
2: $[\boldsymbol{\alpha}, \boldsymbol{\beta}] := [1, 1, \cdots] \in \mathbb{R}^{R_k+N_k}$
3: $\mathcal{R}_k := \{[\boldsymbol{\alpha}, \boldsymbol{\beta}]\}$ ▷ Define the candidates to be added to $\mathcal{I}_k$
4: $\mathcal{U}_k := \mathcal{U}_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$ ▷ Determine the initial training samples using $\hat{\Xi}_k, \rho_z$
5: $\mathcal{Y}_k := f_{k,\boldsymbol{\alpha}}(\mathcal{U}_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]})$ ▷ Evaluate the component-model at the training samples
6: $\mathcal{C}_k := \emptyset$ ▷ Initialize empty combination coefficients of MISC approximation
7: $\mathcal{E}_k := \{\infty\}$ ▷ Set error indicators infinitely large to ensure initial index $[\boldsymbol{\alpha}, \boldsymbol{\beta}]$ is added to $\mathcal{I}_k$ before all others

4.3 | Refinement

Once initialized, Line 5 of Algorithm 1 chooses the best individual component-surrogate to refine. Each component-surrogate has a set of indices $\mathcal{R}_k$ (depicted by red-boxes in the right-most panels of Figure 5) associated with tensor-product interpolants $f_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$ (depicted by the blue-dotted and green-dash-dotted lines in Figure 5), which can be added to the MISC approximation. Each of these candidate indices is also associated with error indicators $\gamma_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]} \in \mathcal{E}_k$ that estimate the change in the error in the system-level QoI obtained from refinement of each surrogate. Thus, Line 5 determines the component-surrogate whose largest error indicator is larger than all other indicators in all sets $\mathcal{E}_k, k = 1, \ldots, K$ (this index is depicted by the striped box in Figure 5) .

When the surrogate of component l is selected for refinement, the `REFINE` routine (see Algorithm 3) removes the chosen index $[\boldsymbol{\alpha}^\star, \boldsymbol{\beta}^\star]$ and associated error indicator from the sets $\mathcal{R}_l$ and $\mathcal{E}_l$, respectively. The chosen index is then added to $\mathcal{I}_l$ (the striped box in the right panel of the first row in Figure 5 becomes shaded gray in the next iteration depicted in the second row) and the combination coefficients of the MISC approximation are computed using (13). New candidate indices $[\boldsymbol{\gamma}, \boldsymbol{\delta}]$ are obtained by incrementing each element of the multi-index $[\boldsymbol{\alpha}^\star, \boldsymbol{\beta}^\star]$ in turn (by adding the unit vector $e_{T_k,i}$ on Line 7). Candidate indices that satisfy the downward closed admissibility criterion (12) are then added to $\mathcal{R}_l$ (e.g. the fourth panel of the second row of Figure 5 has a new red box in the index set $\mathcal{R}_2$ not present in the fourth panel of the first row). In Figure 5 only one new index can be added to either component surrogate because the component-models are one-dimensional. However, for higher-dimensional functions multiple new indices may be added, as determined by the index set admissibility criterion (12).

The training samples $\mathcal{U}_{l,[\boldsymbol{\gamma},\boldsymbol{\delta}]}$ needed to build the new approximation $f_{l,[\boldsymbol{\gamma},\boldsymbol{\delta}]}$ and the associated evaluations of $f_{l,\boldsymbol{\gamma}}$ are generated for each admissible candidate index (e.g. in the second-from-left panel of Figure 5 there are two new samples in the second row

[3]The set $\mathcal{R}_k$ is dependent on $\mathcal{I}_k$ and $\mathcal{E}_k$ is dependent on $\mathcal{R}_k$, however we do not explicitly include this dependence in the notation of these sets for simplicity.

that are not in the first row). The samples of the exogeneous and coupling variables are generated according to the PDF ρ_z of the exogeneous variables and the estimated range of the coupling variables $\hat{\Xi}_k$, respectively. The exact number of new samples in $\mathcal{U}_{l,[\boldsymbol{\gamma},\boldsymbol{\delta}]}$ is dictated by $\boldsymbol{\delta}$ and Equations (14) and (15). When using nested Leja sequences, many points in $\mathcal{U}_{l,[\boldsymbol{\gamma},\boldsymbol{\delta}]}$, needed to construct $f_{l,[\boldsymbol{\gamma},\boldsymbol{\delta}]}$, may have already been evaluated and can simply be reused. Consequently the function $f_{l,\boldsymbol{\gamma}}$ is evaluated on the set of new points $\mathcal{U}_{k,[\boldsymbol{\gamma},\boldsymbol{\delta}]} \setminus \mathcal{U}_k$.

Algorithm 3 REFINE$[[\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star],\mathcal{I}_k,\mathcal{R}_k,\mathcal{E}_k,\mathcal{U}_k,\mathcal{Y}_k,\mathcal{C}_k,\mathcal{F}_k,\hat{\Xi}_k,\rho_z] \to \mathcal{I}_k,\mathcal{R}_k,\mathcal{E}_k,\mathcal{U}_k,\mathcal{Y}_k,\mathcal{C}_k$

$T_k := R_k + N_k$
$\mathcal{R}_k := \mathcal{R}_k \setminus [\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star]$ ▷ Remove selected index from candidate set
$\mathcal{E}_k := \mathcal{E}_k \setminus \gamma_{k,[\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star]}$ ▷ Remove selected error indicator
$\mathcal{I}_k := \mathcal{I}_k \cup [\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star]$ ▷ Add selected index from MISC approximation
$\mathcal{C}_k := \{c_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}\}_{[\boldsymbol{\alpha},\boldsymbol{\beta}]\in\mathcal{I}_k}$ ▷ Compute MISC combination coefficients using (13)
for $i = 1,\dots,T_k$ **do**
  $[\boldsymbol{\gamma},\boldsymbol{\delta}] := [\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star] + e_{T_k,i}$ ▷ Refine index in one coordinate direction to increase deterministic or parametric fidelity
  if $[\boldsymbol{\gamma},\boldsymbol{\delta}] - e_{T_k,j} \in \mathcal{I}_k \; \forall j = 1,\dots,T_k$ **then** ▷ Check downwards closed condition (12)
    $\mathcal{R}_k := \mathcal{R}_k \cup \{[\boldsymbol{\gamma},\boldsymbol{\delta}]\}$ ▷ Add new refinement candidate
    $\mathcal{U}_k := \mathcal{U}_k \cup \mathcal{U}_{k,[\boldsymbol{\gamma},\boldsymbol{\delta}]}$ ▷ Add new training points using $\hat{\Xi}_k, \rho_z$
    $\mathcal{Y}_k := \mathcal{Y}_k \cup f_{k,\boldsymbol{\gamma}}(\mathcal{U}_{k,[\boldsymbol{\gamma},\boldsymbol{\delta}]} \setminus \mathcal{U}_k)$ ▷ Add evaluations of the component-model on the new points
  end if
end for

4.4 | Estimating Error

Once new candidate indices have been generated by refining the l-th component-surrogate, we must estimate their impact on the error in predictions of the system-model QoI. We quantify this impact via error indicators that measure the change in error in system QoI caused by adding a new single-fidelity surrogate to the multi-fidelity MISC approximation of a component, relative to the cost of training the new surrogate. The coupled nature of the integrated-surrogate means that changes in one component-surrogate impact some or all other components. Consequently, the final steps of the adaptive algorithm use COMPUTE_ERROR_INDICATORS (see Algorithm 4) to compute the error indicators associated with all the candidate indices of all K components, not just the newly added candidates of the selected l-th component.

To construct our error indicator, we generate a set $\mathcal{Z}_{\text{refine}} = \{\boldsymbol{z}^{(l)}\}_{l=1}^{L_{\text{refine}}}$ of L_{refine} realizations of the input random variables $\boldsymbol{z}$. We then evaluate the integrated-surrogate at these samples, using the current set of component-surrogates, yielding $\boldsymbol{y}_{\mathcal{J}} = f_{\mathcal{J}}(\boldsymbol{z})$ and, for each new possible $f_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$, compare these evaluations with the output of the refined surrogate $\boldsymbol{y}_{\mathcal{J}^k_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}} = f_{\mathcal{J}^k_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}}(\boldsymbol{z})$ obtained by adding $f_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$ to (11), where

$$\mathcal{J}^j_{[\boldsymbol{\alpha},\boldsymbol{\beta}]} := \{\mathcal{I}_k\}_{k=1,k\neq j}^{K} \cup \left(\mathcal{I}_j \cup \{[\boldsymbol{\alpha},\boldsymbol{\beta}]\}\right).$$

These evaluations are carried out using the EVALUATE_INTEGRATED_SURROGATE procedure (not shown) which integrates the predictions of each component-surrogate, obtained from (11), using the algorithms outlined in Section 2.2.

Focusing on the prediction of system QoI, that is $\boldsymbol{q}^{(l)}_{\mathcal{J}^k_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}} = \boldsymbol{A}^q \boldsymbol{y}_{\mathcal{J}^k_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}}$ and $\boldsymbol{q}^{(l)}_{\mathcal{J}} = \boldsymbol{A}^q \boldsymbol{y}_{\mathcal{J}}$, and recalling that $\boldsymbol{q}_i$ denotes the i-th entry of QoI vector $\boldsymbol{q}$, we use the error indicators

$$\gamma_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]} = \frac{\Delta E_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}}{\Delta W_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}}, \tag{16}$$

uniquely defined by the two quantities

$$\Delta E_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]} = \max_{i=1,\dots,Q^{\text{sys}}} \left(\frac{1}{L_{\text{refine}}} \sum_{l=1}^{L_{\text{refine}}} \left(\boldsymbol{q}^{(l)}_{\mathcal{J}^k_{[\boldsymbol{\alpha},\boldsymbol{\beta}]},i} - \boldsymbol{q}^{(l)}_{\mathcal{J},i} \right)^2 \right)^{\frac{1}{2}} \qquad \Delta W_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]} = \left| \text{Work}[Y_{\mathcal{J}^k_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}}] - \text{Work}[Y_{\mathcal{J}}] \right|, \tag{17}$$

which respectively denote the worst-case root mean squared error (over all system-level QoI) between two successive approximations, and the work needed to update the approximation. Here $\Delta E_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$ is a measure of the "difference" between the black-dashed

approximation and the blue-dotted and green-dash-dotted approximations of the system QoI in the 3rd panels of both rows in Figure 5. Due to the nestedness of the Leja points and making the often reasonable assumption that the cost of a simulation $W_{k,\boldsymbol{\alpha}}$ for a given fidelity $\boldsymbol{\alpha}$ is fixed (i.e. does not change with $\boldsymbol{u}$), the quantity $\Delta W_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$ has the closed form expression

$$\Delta W_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]} = W_{k,\boldsymbol{\alpha}}\text{card}(\mathcal{U}_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]} \setminus \mathcal{U}_k), \tag{18}$$

where $\mathcal{U}_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]} \setminus \mathcal{U}_k$ is the set of new evaluations needed to complete the construction of $f_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$. The evaluation of the error indicator in (16) is implemented by the `ERROR_INDICATOR` function (not shown).

The refinement indicator (16) ranks component-surrogates based upon their contribution to the error in the approximation of the system-level QoI. Unfortunately, using (16) will give misleading results when the surrogate of a downstream component l is a constant function (which happens after initializing Algorithm 1) and no outputs from the k-th component are present in the system QoI $\boldsymbol{q}$. In these situations the indicator of the k-th component will predict no system improvement when improving the surrogate of the upstream component k, i.e. $\Delta E_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$ will be zero. Such a situation occurs in the first row of Figure 5; the current approximation of the second component (black dashed line in second panel) is a constant so improvements in the first component surrogate (i.e. moving from black-dashed approximation to blue-dotted approximation in first panel) yield no change in the integrated-surrogate (the black-dashed and blue-dotted lines are the same in the third panel of Figure 5). To avoid this pathological issue associated with initialization, we compute the following local component-based error indicator (which is a measure of the difference between the current and new component-surrogates in the two left-most panels of Figure 5)

$$\begin{aligned}
\gamma_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]} &= \frac{1}{\Delta W_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}}\left(\theta\Delta E^{\mu}_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]} + (1-\theta)\Delta E^{\sigma}_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}\right), \quad \theta \in [0,1] \\
\Delta E^{\mu}_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]} &= \frac{1}{\left|f_{k,\mathbf{0},\mathbf{0}}\right|}\left|\mathbb{E}\left[f_{k,\mathcal{I}_k\cup\{[\boldsymbol{\alpha},\boldsymbol{\beta}]\}}\right] - \mathbb{E}\left[f_{k,\mathcal{I}_k}\right]\right|, \\
\Delta E^{\sigma}_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]} &= \frac{1}{\left|f_{k,\mathbf{0},\mathbf{0}}\right|}\sqrt{\left|\mathbb{V}\left[f_{k,\mathcal{I}_k\cup\{[\boldsymbol{\alpha},\boldsymbol{\beta}]\}}\right] - \mathbb{V}\left[f_{k,\mathcal{I}_k}\right]\right|},
\end{aligned} \tag{19}$$

until all downstream surrogates are no longer constants. Here the operator $\mathbb{E}[\cdot]$ denotes expectation with respect to both the external inputs $\boldsymbol{z}_k$ of the current model as well as to its coupling variables $\boldsymbol{\xi}_k$; the normalization factor $\left|f_{k,\mathbf{0},\mathbf{0}}\right|$ is the value of the coarsest fidelity at the center of the parametric domain $\Gamma_k \times \hat{\Xi}_k$ and thus can be understood as a coarse approximation of $\mathbb{E}\left[f_k\right]$; and the operator $\mathbb{V}[\cdot]$ denotes variance with respect to the same variables. In the numerical tests to follow, we always use $\theta = 0.5$. Heuristics, such as the one used here, are often employed to overcoming initialization of single-fidelity sparse grid approaches which form the basis of the algorithm presented here.

Algorithm 4 `COMPUTE_ERROR_INDICATORS`$[\{\mathcal{I}_k\}_{k=1}^K, \mathcal{R}_j, \mathcal{E}_j, \{\mathcal{U}_k\}_{k=1}^K, \{\mathcal{Y}_k\}_{k=1}^K, \{\hat{\Xi}_k\}_{k=1}^K, \rho_z] \rightarrow \mathcal{E}_j, \{\hat{\Xi}_k\}_{k=1}^K$

$\mathcal{E}_j := \emptyset$ ▷ Clear current error indicators so they can be overwritten
$\mathcal{Z}_{\text{refine}} :=$ `RANDOM_SAMPLE`$[\rho_z, L_{\text{refine}}]$
$\mathcal{J} := \{\mathcal{I}_k\}_{k=1}^K$ ▷ Define index set of current MISC approximation of j-th component
$\boldsymbol{y}_{\mathcal{J}} :=$ `EVALUATE_INTEGRATED_SURROGATE`$[\mathcal{J}, \{\mathcal{U}_k\}_{k=1}^K, \{\mathcal{Y}_k\}_{k=1}^K, \{C_k\}_{k=1}^K, \mathcal{Z}_{\text{refine}}]$
$\boldsymbol{q}_{\mathcal{J}} := A^q\boldsymbol{y}_{\mathcal{J}}$ ▷ Extract system QoI
for $[\boldsymbol{\alpha},\boldsymbol{\beta}] \in \mathcal{R}_j$ **do**
  $\mathcal{J}^j_{[\boldsymbol{\alpha},\boldsymbol{\beta}]} := \{\mathcal{I}_k\}_{k=1,k\neq j}^K \cup \left(\mathcal{I}_j \cup \{[\boldsymbol{\alpha},\boldsymbol{\beta}]\}\right)$ ▷ Define index set of refined MISC approximation of j-th component
  $C_{j,[\boldsymbol{\alpha},\boldsymbol{\beta}]} := \{c_{[\boldsymbol{\gamma},\boldsymbol{\delta}]}\}_{[\boldsymbol{\gamma},\boldsymbol{\delta}]\in\mathcal{I}_j\cup\{[\boldsymbol{\alpha},\boldsymbol{\beta}]\}}$ ▷ Compute coefficients of refined j-th MISC approximation using (13)
  $\boldsymbol{y}_{\mathcal{J}^j_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}} :=$ `EVALUATE_INTEGRATED_SURROGATE`$[\mathcal{J}^j_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}, \{\mathcal{U}_k\}_{k=1}^K, \{\mathcal{Y}_k\}_{k=1}^K, \{C_k\}_{k=1,k\neq j}^K \cup C_{j,[\boldsymbol{\alpha}\cdot\boldsymbol{\beta}]}, \mathcal{Z}_{\text{refine}}]$
  $\boldsymbol{q}_{\mathcal{J}^j_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}} := A^q\boldsymbol{y}_{\mathcal{J}^j_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}}$ ▷ Extract system QoI
  $\gamma_{j,[\boldsymbol{\alpha},\boldsymbol{\beta}]} :=$ `ERROR_INDICATOR`$[\boldsymbol{q}_{\mathcal{J}}, \boldsymbol{q}_{\mathcal{J}^j_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}}]$ ▷ Use (16)
  $\mathcal{E}_j := \mathcal{E}_j \cup \{\gamma_{j,[\boldsymbol{\alpha},\boldsymbol{\beta}]}\}$
  $\{\hat{\Xi}_k\}_{k=1}^K :=$ `UPDATE_COUPLING_VARIABLE_RANGES`$[\boldsymbol{y}_{\mathcal{J}^j_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}}, \{\hat{\Xi}_k\}_{k=1}^K]$
end for

4.5 | Termination

Steps 4-9 of Algorithm 1 are repeated until a computational budget $W_{\max}$ is exceeded or the pre-specified accuracy tolerance τ, specified by the user based upon their accuracy requirements, is met. For example, one could stop as soon as all error indicators are below such tolerance, $\max_{k\in 1,\dots,K} \max_{[\boldsymbol{\alpha},\boldsymbol{\beta}]\in\mathcal{R}_k} \gamma_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]} \leq \tau$. Such exit conditions are encapsulated by the `TERMINATE` routine (not shown).

The final output of Algorithm 1 is a set of surrogates $f_{k,\mathcal{I}_k}$, defined by the index sets $\mathcal{J} = \{\mathcal{I}_1, \dots, \mathcal{I}_K\}$, which can be used to accurately predict the system-level QoI by evaluating the joint surrogate $f_{\mathcal{J}}$. In practice the sets $\mathcal{I}_k \in \mathcal{J}$ returned by Algorithm 1 are $\mathcal{I}_k \cup \mathcal{R}_{\mathcal{I}_k}$ rather than $\mathcal{I}_k$. This is because Algorithm 1 can compute the error indicator of a multi-index only after having added it to the approximation. Thus, a final post-processing step augments the final approximation with all remaining candidates that have been evaluated but not yet selected.

4.6 | Building Leja sequences and estimating the range of the coupling variables

Lemma 1 indicates that the estimated ranges $\hat{\Xi}_k$ of the coupling variables can significantly impact the accuracy of a component-surrogate and thus the integrated surrogate. In this section we present an adaptive algorithm to iteratively learn the ranges of the coupling variables of component-surrogates constructed using tensor-product interpolation, The algorithm leverages the nested property of Leja sequences to dynamically adjust the quadrature rules used to construct the MISC approximation for each component.

Univariate weighted Leja sequences over a range $I \subset \mathbb{R}$ are constructed sequentially. Given a sequence of $m(\beta_n)$ points $\mathcal{U}_{k,n,\boldsymbol{\alpha},\beta_n} = \{u^{(j)}_{k,n,\boldsymbol{\alpha},\beta_n}\}_{j=1}^{m(\beta_n)}$, the sequence with $m(\beta_n)+1$ points is obtained adding to the current sequence the following point:

$$u^\star = \underset{u\in I}{\operatorname{argmax}}\, v(u) \prod_{j=1}^{m(\beta_n)} |u - u^{(j)}_{k,\boldsymbol{\alpha},n,\beta_n}| \tag{20}$$

for some weight function $v(u)$. In this paper, we follow[49] and set $v(u) = \sqrt{\nu(u)}$ where $\nu(u)$ is the PDF of the variable u. For uniform variables used to represent coupling variables (cf. Section 3.3), the PDF is a constant and does not affect the Leja sequence. However, the PDF does effect the Leja sequences used for the exogeneous variables.

By construction, Leja sequences are nested, that is $\mathcal{U}_{k,n,\boldsymbol{\alpha},\beta_n} \subset \mathcal{U}_{k,n,\boldsymbol{\alpha},\beta_n+1}$. Moreover, the initial point $u^{(1)}$ may be arbitrarily chosen. Indeed, we can extend any set of initial points $\mathcal{U}_{k,n,\boldsymbol{\alpha},\beta_n}$. Thus, given a Leja sequence $\mathcal{U}_{k,n,\boldsymbol{\alpha},\beta_n}$ constructed on a range I, we can generate the next point $u^\star$ of the Leja sequence over a larger range I' by simply searching for the next point in the interval I' instead of I. We can utilize this approach to adapt the surrogates of system components to dynamically changing estimates of the coupling variable ranges.

Iteratively estimating the ranges of the coupling variables requires the introduction of the `UPDATE_COUPLING_VARIABLE_RANGES` routine in Algorithm 4 (not shown). This routine estimates the ranges of $\boldsymbol{\xi}$ using the values $\boldsymbol{y}_{\mathcal{J}}$ and of $\boldsymbol{y}_{\mathcal{J}^j_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}}$ computed on Lines 4 and 9, respectively. Specifically, consider the i-th coupling variable of the k-th component $\xi_{k,i}$ and assume that its value is determined by the q-th output of the m-th component, i.e., $\xi_{k,i} = y_{m,q}$. Then using $\hat{\Xi}_{k,i} = [a_{k,i}, b_{k,i}]$ to denote the current range of $\xi_{k,i}$, the new range $\hat{\Xi}'_{k,i} = [a'_{k,i}, b'_{k,i}]$ is updated as follows:

$$a'_{k,i} = \min\left(a_{k,i}, \min\left(\boldsymbol{y}_{\mathcal{J}}, \boldsymbol{y}_{\mathcal{J}^j_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}}\right)\right) \qquad b'_{k,i} = \max\left(b_{k,i}, \max\left(\boldsymbol{y}_{\mathcal{J}}, \boldsymbol{y}_{\mathcal{J}^j_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}}\right)\right).$$

In the numerical examples to follow, we evaluate the entire system at $L_{\text{refine}} = 100$ random realizations of the input $\boldsymbol{z}$, and set the initial ranges $\Xi^0_{k,i}$ of each coupling variable to be the minimum and maximum values over the 100 samples. Despite the inaccuracy of this initial guess, the performance obtained using this approach along with the subsequent learning procedure was found to be very similar to that obtained using more carefully constructed bounds that were set *a priori*.

4.7 | Remarks

Algorithm 1 is a generalization of the adaptive MISC algorithm developed for black box-models proposed in[30,33]. When $K = 1$ Algorithm 1 recovers the original algorithm as a special case. This original algorithm can be applied to system-models by ignoring the coupled nature of the system-level input-output map (1) and treating the system model as a single component. When used in this way, the original algorithm must employ a multi-index $\boldsymbol{\alpha}$ controlling the physical fidelities of the system-model,

such that

$$\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \dots, \boldsymbol{\alpha}_K].$$

is the concatenation of the hyper-parameters $\boldsymbol{\alpha}_k$ for each component. In the numerical examples that follow we will refer to this specialized case as the black-box approach.

Unfortunately, to date we have been unable to make strong theoretical statements about convergence of Algorithm 1. This is partly to be expected, since convergence has only been partially addressed even for the black-box variant[31,34]. These initial results focus on application of MISC to elliptic PDEs with random coefficients. The machinery to prove convergence borrows from the tools used to prove convergence of (single-fidelity) sparse grids approximations (see e.g.[51,52,53,54]) and requires certain regularity (boundedness of mixed spatial-parametric derivatives) of the function to be approximated, as well as suitable choices of the error-indicators (slightly different from the one we use here, since our choice is tailored to multi-component systems).

5 | NUMERICAL EXAMPLES

In this section, we investigate the performance of the proposed method using several numerical examples. In all examples, we build surrogates of each component and report the error in the predictions of the integrated surrogate against the total cost of running the algorithm. For comparison purposes, we also report error vs cost of the black-box version of the MISC algorithm discussed in Section 4.7. All the numerical results were produced using the PyApprox software package[55].

Error is measured by drawing 1000 random samples from the PDF of the exogeneous variables $\boldsymbol{z}$ and computing the relative ℓ^2 (root mean squared) error between the exact system output and the surrogate approximation, normalized by the ℓ^2 norm of the validation data. For computationally expensive component-models, the computational cost of constructing integrated surrogates using Algorithm 1 dominates the cost of evaluating the resulting surrogate at validation samples, which we therefore consider negligible. Consequently, the total cost W_{tot} of the algorithm is the computational cost required to generate the training data $\mathcal{Y}_1, \mathcal{Y}_2, \dots, \mathcal{Y}_K$. More formally, the total cost is the sum of the $\Delta W_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$ computed for the indices $[\boldsymbol{\alpha}, \boldsymbol{\beta}]$ visited during the execution of the algorithm(see (18)), that is

$$W_{tot} = \sum_{k=1}^{K} \sum_{[\boldsymbol{\alpha},\boldsymbol{\beta}] \in \mathcal{I}_k \cup \mathcal{R}_k} \Delta W_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}.$$

The cost of propagating samples through feed-forward coupling and FPI used to resolve feedback-coupling is negligible because both tasks are performed on the surrogates. In contrast, the black-box MISC algorithm requires evaluations of the system-model. Consequently the cost of fixed-point iterations must be included, because they require evaluations of the expensive component-models. Assuming that a fixed number of fixed point iterations P are performed, the total cost of the black-box approach is

$$W_{tot} = \sum_{[\boldsymbol{\alpha},\boldsymbol{\beta}] \in \mathcal{I} \cup \mathcal{R}} \left(P \Delta W_{[\boldsymbol{\alpha},\boldsymbol{\beta}]} \right), \quad \text{for black-box MISC.} \tag{21}$$

Here we have dropped the dependence on k from $W, \mathcal{I}$ and $\mathcal{R}$ to emphasize the black-box approach does not consider the existence of components. In the following examples we report the total cost in terms of the number of equivalent highest-fidelity evaluations.

5.1 | Algebraic single-fidelity feed-forward system

Consider the coupled system depicted in Figure 6. The system consists of three components in a chain with vector-valued feed-forward coupling, where $f_1(\boldsymbol{z}_1) = (y_{1,1}, \dots, y_{1,Q_1})^\top$, $f_2(\boldsymbol{z}_2, \boldsymbol{\xi}_2) = (y_{2,1}, \dots, y_{2,Q_2})^\top$ and $f_3(\boldsymbol{z}_3, \boldsymbol{\xi}_3) = [y_{3,1}]$, and the expressions

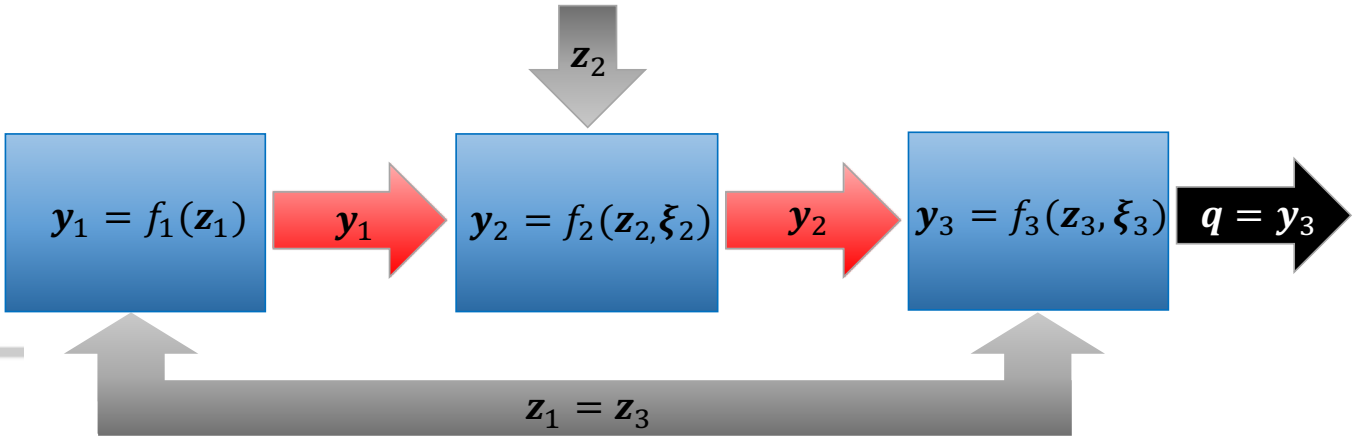

FIGURE 6 A feed-forward system consisting of three components. Coupling variables are depicted in red, external inputs in gray and system-level QoI in black. The first and third components share the same random variables, i.e. $\boldsymbol{z}_1 = \boldsymbol{z}_3$, and the inputs to the second component are unique to that component i.e. $\boldsymbol{z}_2 \cap \boldsymbol{z}_k = \emptyset$, $k = 1, 3$.

of the outputs $y_{i,j}$ are given by

$$
\begin{aligned}
y_{1,\alpha_1,q} &= f_{1,\alpha_1,q}(\boldsymbol{z}_1) = z_1^q \sin\left(\sum_{d=1}^{D_1} z_{1,d} + \epsilon_{\alpha_1}\right), \quad q = 1, \ldots, Q_1 \\
y_{2,\alpha_2,q} &= f_{2,\alpha_2,q}(\boldsymbol{z}_2, \boldsymbol{\xi}_2) = \left(\prod_{s=1}^{S_2}(\xi_s^{q+1} - \epsilon_{\alpha_2})\right)\left(\prod_{d=1}^{D_2} z_{2,d}\right), \quad q = 1, \ldots, Q_2 \\
y_{3,\alpha_3,q} &= f_{3,\alpha_3}(\boldsymbol{z}_3, \boldsymbol{\xi}_3) = \exp\left[-\sum_{s=1}^{S_3}(\xi_s - \epsilon_{\alpha_3})^2\right] \frac{1}{1 + \frac{25}{16}\left(\sum_{d=1}^{D_3} z_{3,d}\right)^2},
\end{aligned}
\tag{22}
$$

with $\boldsymbol{\xi}_2 = (y_1, \ldots, y_{1,Q_1})^\top$, $\boldsymbol{\xi}_3 = (y_2, \ldots, y_{2,Q_2})^\top$ and the discretization parameters $\alpha_k = 1, 2, \ldots$ control the values of ϵ_{α_k} and thus the accuracy of f_{k,α_k}.

The coupled system is parameterized by $D_1 + D_2$ independent and identically distributed uniform random variables on $[0, 1]$. The first and third components are parameterized by the same two random variables, that is $\boldsymbol{z}_1 = \boldsymbol{z}_3$ and $D_1 = D_3$ in (22). The second component is parameterized by another two variables such that $\boldsymbol{z}_1 \cap \boldsymbol{z}_2 = \emptyset$. The number of random variables and outputs of each component is scalable. Here we set $Q_3 = 1$ and consider three cases: Case 1 with $Q_1 = Q_2 = 1$, $D_1 = D_2 = 1$, $\epsilon_{\alpha_k} = 0\ \forall k$, Case 2 with $Q_1 = Q_2 = 4$, $D_1 = D_2 = 2$, $\epsilon_{\alpha_k} = 0\ \forall k$, and Case 3 with $Q_1 = Q_2 = 2$, $D_1 = D_2 = 2$, $\epsilon_{\alpha_k} \geq 0\ \forall k$). The first two cases use single-fidelity models for each system component, and are devised to analyze the impact of range estimation (Case 1), and of dimension reduction and non-linearity (Case 2). We set $\alpha_k \to \infty$ such that $\epsilon_k = 0$, $k = 1, \ldots K$ and assume that evaluating each component has the same computational cost, which means evaluating the entire system costs three times as much as evaluating a single component. Conversely, Case 3 is designed to investigate the effectiveness of the multi-fidelity approach and therefore utilizes models of varying accuracy cost for each component.

5.1.1 | Case 1: The impact of range estimation

In Figure 7 (left), we plot the error in the predictions of f_3 for Case 1 using different ranges for the coupling variables; the true PDF of the coupling variables and the output of the third component are shown in Figure 7 (right). The intervals in the legend denote the *a priori* fixed range of the coupling variables, while the legend element "Estimated" refers to the approximation obtained when using the adaptive range estimation procedure outlined in Section 4.6. Note that in this example the true range of the coupling variables can be determined from inspection of the component functions. The exact ranges of the coupling are all $[0, 1]$, therefore the corresponding convergence curve can be considered the best possible performance that any method employing range estimation can achieve. Figure 7 also plots the convergence of error in a single surrogate that treats the system as a black-box.

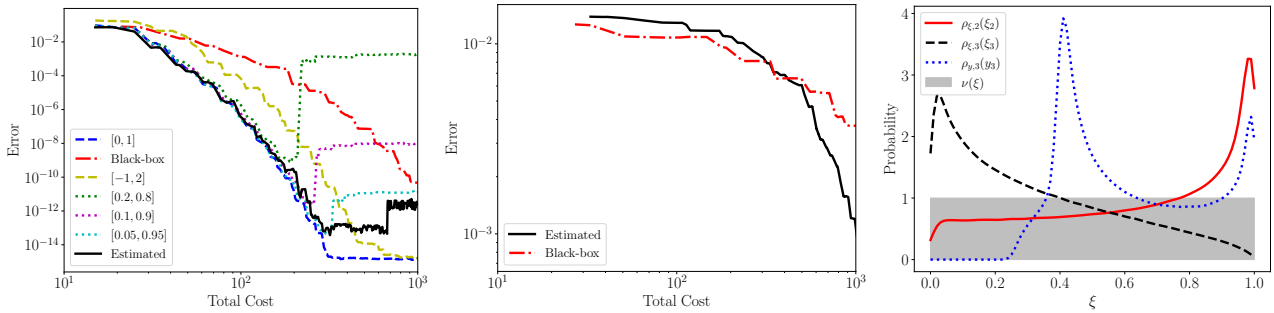

FIGURE 7 Error vs cost for the feed-forward system (22) for (left) Case 1 with ($Q_1 = Q_2 = 1$, $D_1 = D_2 = 1$) and (middle) Case 2 with ($Q_1 = Q_2 = 4$, $D_1 = D_2 = 2$). (Right) True PDFs of the coupling variables $\rho_{\xi,k}$ and output $\rho_{y,k}$ of the integrated system for Case 1. The dominating measure (ν from Lemma 1), used to compute the Leja sequences for the coupling variables on $[0, 1]$, is depicted in gray.

All integrated-surrogates converge much faster than the case of a single black-box surrogate. However, the error in the integrated-surrogates is impacted by the ranges used for the coupling variables. The errors for all surrogate cases that under-estimate the range of the coupling variables, saturate at a level proportional to δ in Lemma 1, where δ reflects the severity of the under-estimation. When the range of the coupling variables is over-estimated, the errors do not saturate; however, the constant of convergence is increased, that is the curve shifts right relative to the black curve based on the true ranges.

The dynamic estimation of the coupling ranges is effective. It identifies the coupling ranges of ξ_1 and ξ_2 to be $[-0.02, 1.00]$ and $[-0.23, 0.92]$, respectively. The error saturates because we only use $L_{\text{refine}} = 100$ samples to estimate the ranges of the coupling variables and thus under-estimate the upper bound of ξ_2. Although not shown, the saturation of error can be removed by increasing the number of samples L_{refine}. Again, we emphasize that this does not increase the number of evaluation of the true component functions. We choose such a small value of L_{refine} to show that, even for very crude estimates of the ranges, the dynamic estimation of the ranges works well. We also remark that the algorithm can sometimes over-estimate the ranges of the coupling variables, as was the case here. This is because the estimation procedure is based on evaluations of the component-surrogates and not the true components. Thus, at early stages of the algorithm, an inaccurate approximation can lead to the bounds being over-estimated. However, for this example and all that follow, we found that any over-estimation did not significantly affect results. Under-estimation is more important to avoid and the algorithm does this effectively. In all remaining numerical studies, we dynamically estimate the range of the coupling variables.

Note that in Figure 7, when the range of the coupling variables is under-estimated, the error in the integrated-surrogates decreases before rapidly increasing and finally saturating. This behavior occurs when the polynomial degree of the third component f_3 surrogate is increased. The third component is based upon a scaled version of the Runge function and so exhibits a "Runge type phenomena", where oscillations in the approximation occur outside the ranges of the coupling variables. Some samples used to estimate the errors reported in Figure 7 require extrapolation in these oscillatory regions. Consequently error, in the approximation of the system-level QoI, decreases until oscillations in the surrogate of the third component start to dominate estimates of error.

5.1.2 | Case 2: dimension reduction and non-linearity

The middle plot of Figure 7 compares the accuracy of integrated-surrogates with system-level black-box surrogates for Case 2. At lower levels of total cost, the system-level black-box surrogate is much more competitive than when used for Case 1. However, the rate of convergence is still much slower than for the integrated-surrogates. In general, there are two reasons for the increased convergence rate of integrated component-surrogates: (i) the components may be lower-dimensional than the entire system and (ii) the components may be less non-linear than the entire system. We expand on both these points below.

Dimension reduction.

The number of evaluations needed to build a component-surrogate increases with the dimension of the component and not the dimensionality of the system. In many cases the number of inputs (coupling and random variables) of a component is smaller than

the number of random variables for the entire system, that is $D_k + S_k < D$. In these situations, constructing approximations for components of a system can be cheaper than a surrogate that treats the system as a black-box. To explain this behavior, consider a tensor-product interpolation of a function, which requires $O\left(\epsilon^{-D}\right)$ black-box evaluations of the entire system[4] to achieve an error ϵ. The cost of using this method to build component-surrogates over both the coupling and random variables satisfies

$$O\left(\sum_{k=1}^{K} \epsilon^{-(D_k+S_k)}\right) < O\left(\epsilon^{-D}\right) \qquad \text{if} \qquad \max_{k=1,\ldots,K} D_k + S_k < D + \frac{\log(K)}{\log(\epsilon)} \tag{23}$$

Here we used $O\left(\sum_{k=1}^{K} \epsilon^{-(D_k+S_k)}\right) < O\left(K\epsilon^{-T}\right)$, were $T = \max_{k=1,\ldots,K} D_k + S_k$.

Non-linearity.

As discussed previously, integrated-surrogates can be represented as a composition of functions. Thus, the composition of the system-level QoI can be more non-linear than any single component. For example, consider a composition of K quadratic functions; the system-level QoI will have degree $2K$ and so will be much more difficult to evaluate than any component. Note that it is theoretically possible for the system-level QoI to be less non-linear than a component; however this phenomenon did not occur in any of our numerical examples.

Case 1 and Case 2 were specifically tailored to highlight the improved expressivity of treating systems as compositions of functions. The difference in performance between system-level black-box and integrated component-surrogates will decrease as the non-linearity of the components decrease. However, several of the following examples show the benefits of our approach even on systems that were not tailored to amplify its benefits.

5.1.3 | Case 3: multi-fidelity approximation

In this section, we investigate the use of an ensemble of models, of varying fidelity, within our integrated-surrogate framework. Specifically, we consider Case 3, that is (22) with $Q_1 = Q_2 = 2$ and $D_1 = D_2 = 2$. By varying α_k, we can produce an ensemble of models of varying cost and accuracy. With this goal, we set $\epsilon_k = 10^{-\alpha_k}$. The effectiveness of multi-fidelity methods is strongly dependent on the cost-to-accuracy ratio and the true cost of evaluating each algebraic component is negligible. For demonstration purposes, however, we define the work needed to evaluate f_{k,α_k} to be $W_{k,\alpha_k} = 1.25^{\alpha_k}$, enabling illustration of the impact of the proposed multi-fidelity sampling algorithm for a cost model that is representative of what might be encountered in practice.

The left graphic of Figure 8 plots the error in the predictions of the integrated multi-fidelity component-surrogates ("MF Integrated"). We also compare that approach with single-fidelity integrated component-surrogates ("SF Integrated") for fixed $\alpha_k = 6$ and single-fidelity and multi-fidelity system-level black-box surrogates, labeled "SF Black-box" and "MF Black-box" respectively. Both integrated approaches are more accurate than their black-box counterparts. However, the greatest gains are made from introducing multiple models and using our multi-fidelity approach.

The middle and right plots of Figure 8 depict the percentage of the computational work allocated to the various model discretizations of each component when the total work is ≈ 484 and ≈ 1005, respectively. The two numbered boxes in the left plot indicate the points on the convergence curve that are associated respectively with the middle and right plots. In the middle plot, evaluations of the second and third components contribute a similar amount to the total work. In the later stages of the algorithm (right plot), more resources are allocated to the third component. This behavior cannot be achieved without considering the effect of each component on the system-level QoI, which is one of the novel aspects of our proposed approach. These two plots also show how work is distributed among each model fidelity for each component. In the early stages of the algorithm, lower-fidelity model evaluations are predominant. However, as the total work increases, the algorithm identifies that increasing amounts of higher-fidelity model evaluations are required to further reduce error. For example, the middle plot shows only four fidelities have been used for component 2 but in the right plot, six fidelities have been evaluated. Higher-fidelity (larger α_k) evaluations are needed when the parametric error $\|f_{k,\alpha} - f_{k,\mathcal{I}_k}\|$ of a component-surrogate becomes commensurate with the deterministic error $\|f_k - f_{k,\alpha}\|$ induced by using the model approximation $f_{k,\alpha}$.

[4]Note the error estimate here is for tensor-product methods. The complexity of sparse grids, upon which MISC is based, grows more slowly with dimension. The exact rate depends on the regularity of the function and so we focus our exposition on tensor-product interpolants. Furthermore, adaptive MISC can produce tensor-product interpolants if all variables and all their combinations are important for the QoI.

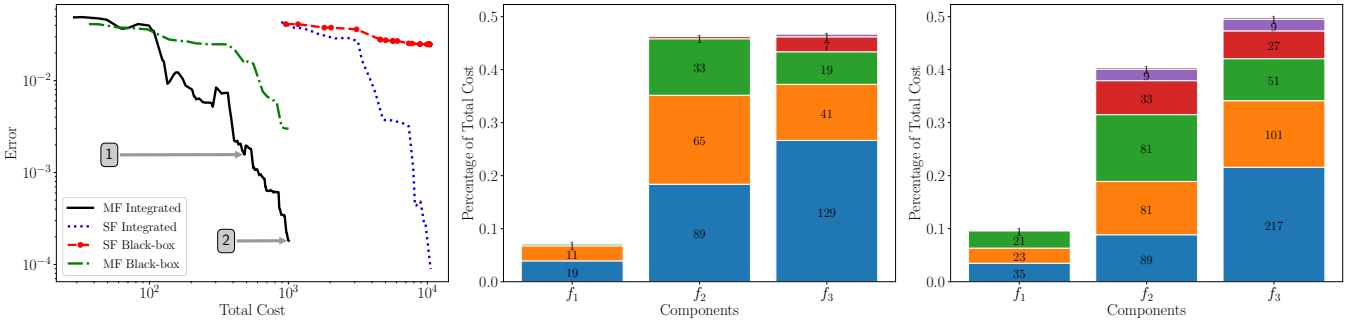

FIGURE 8 (Left) Error vs cost for the feed-forward multi-fidelity system (22). Evaluations allocated by the MF Integrated sampling procedure to the varying fidelity models of each component when the total cost is approximately 484 (middle) and 1005 (right). The two numbered boxes in the left plot indicate the points on the convergence curve that are associated respectively with the middle and right plots.

5.2 | Fire detection satellite

In this section we apply the proposed methodology to a fire detection satellite designed to detect, identify and monitor forest fires. Figure 1 depicts a conceptual diagram of the system-model and its couplings. The definitions of the coupling variables are given in Table 3. The system consists of three components with both feed-forward and feedback coupling. The model equations of these components are documented in[56]. The system has eight random variables (see Table 2) and seven coupling variables (see Table 3). Model constants are reported in[25]. In the following, we use integrated-surrogates to accurately approximate three outputs of the system: the total torque τ_{tot} $(y_{3,2})$ coming from the attitude control component, the total power output P_{tot} $(y_{2,3})$, and the area of the solar array A_{sa} $(y_{2,4})$ coming from the power component. We use fixed-point iteration (see Section 2.2.2) to solve for the feedback coupling variables.

TABLE 2 Random variables of the fire detection satellite system depicted in Figure 1. The System Index denotes the index of the variable in the aggregated set of system random variables $\boldsymbol{z}$. Each variable enters the component variables $\boldsymbol{z}_k$ in the column entitled Component Variables. Arguments of the Gaussian distributions are mean and standard deviation.

System Index	Random Parameter Name	Symbol	Component Variables	Distribution
1	Satellite altitude	H	$\boldsymbol{z}_1, \boldsymbol{z}_3$	$\mathcal{N}\left(18\times10^6, 1\times10^6\right)$
2	Target diameter	ϕ	$\boldsymbol{z}_1$	$\mathcal{N}\left(235, 10\right)$
3	Other power sources	P_o	$\boldsymbol{z}_2$	$\mathcal{N}\left(1000, 50\right)$
4	Solar flux	F_s	$\boldsymbol{z}_2, \boldsymbol{z}_3$	$\mathcal{N}\left(1400, 20\right)$
5	Moment arm for solar radiation torque	L_{sp}	$\boldsymbol{z}_3$	$\mathcal{N}\left(2, 0.4\right)$
6	Reflectance factor	q	$\boldsymbol{z}_3$	$\mathcal{N}\left(0.5, 0.1\right)$
7	Moment arm for aerodynamic torque	L_a	$\boldsymbol{z}_3$	$\mathcal{N}\left(2, 0.4\right)$
8	Drag coefficient	C_d	$\boldsymbol{z}_3$	$\mathcal{N}\left(1, 0.2\right)$)

The left plot of Figure 9 plots the error in the integrated component-surrogates ("Integrated") as the total cost of building the three surrogates increases. For a given cost, the error is much smaller than the error of the surrogate that treats the system as a black-box ("Black-box"). The black-box approach requires $P = 3$ fixed-point iterations to estimate the coupling variables, cf. Equation (21). Assuming the cost of each component is one unit, the evaluation of the black-box requires one evaluation of the first component and three evaluations of the remaining components, that is $1 + 3 \times 2 = 7$ units. In comparison, the integrated component-surrogate approach assigns evaluations to each component individually, using estimates of the impact of component error on the approximation error of system-level QoI. The number of evaluations allocated to each component is depicted in the middle and right graphics of Figure 9 when the total cost is 131 and 229 respectively. The two numbered boxes in the left plot indicate the points on the convergence curve that are associated respectively with the middle and right plots. The algorithm

TABLE 3 Component outputs of the fire detection satellite system depicted in Figure 1. The System Index denotes the index of the output in the aggregated set of system outputs $\boldsymbol{y}$. Each output is present in the coupling variables $\boldsymbol{\xi}_k$ in the column entitled Coupling Variables. A dash in the coupling variable column indicates the output is a system-level QoI.

System Index	Output Name	Output Variable	Coupling variables
1	Satellite velocity	$y_{1,1}$	$\boldsymbol{\xi}_3$
2	Orbit period	$y_{1,2}$	$\boldsymbol{\xi}_2, \boldsymbol{\xi}_3$
3	Eclipse period	$y_{1,3}$	$\boldsymbol{\xi}_2$
4	Max slewing angle	$y_{1,4}$	$\boldsymbol{\xi}_3$
5	Minimum moment of inertia	$y_{2,1}$	$\boldsymbol{\xi}_3$
6	Maximum moment of inertia	$y_{2,2}$	$\boldsymbol{\xi}_3$
7	Total power output	$y_{2,3}$	—
8	Area of solar array	$y_{2,4}$	—
9	Attitude control power	$y_{3,1}$	$\boldsymbol{\xi}_2$
10	Total torque	$y_{3,2}$	—

allocates more computational resources to approximating the second and third component. Note, as previously mentioned at the beginning of Section 5, the total costs reported do not include the cost of the FPI needed to integrate the surrogates, which is negligible relative to evaluation of the true components. Also note that the saturation of the errors, present for both approaches depicted in the left plot of Figure 9, is due to numerical precision issues with solving the Attitude component. It is not due to FPI, which computes the values of the coupling variables to machine precision in three iterations.

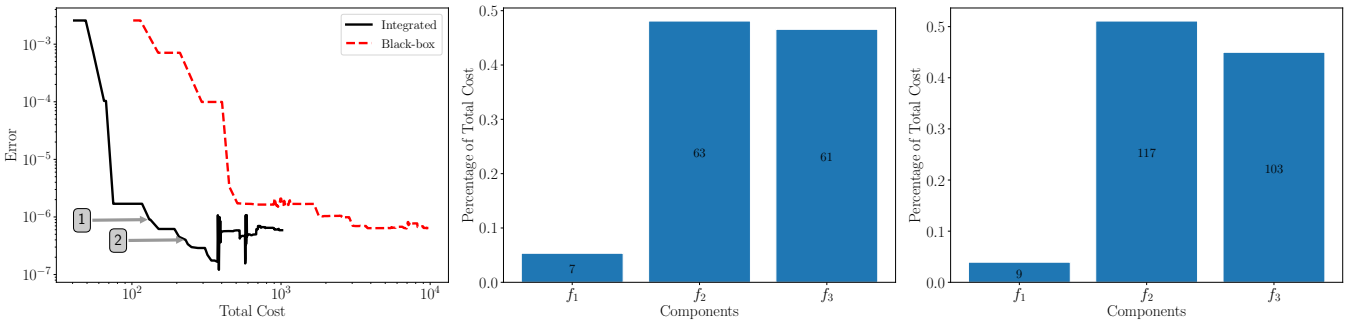

FIGURE 9 (Left) Error vs cost for the QoI P_{tot} obtained from the fire detection satellite system-model. (Middle) The percentage of the computational work allocated to evaluating each component when building the integrated-surrogates when total cost is 131 and (Right) 229. The two numbered boxes in the left plot indicate the points on the convergence curve that are associated respectively with the middle and right plots. Numbers inside the bars represent the absolute number of component evaluations.

5.3 | Economics-turbine model

In this section we investigate the performance of our methodology on a coupled financial model for a gas turbine as depicted in Figure 10[20,57]. The system consists of four component-models and is parameterized by 11 random variables. The distribution of the random variables is given in Table 4, where $\boldsymbol{z}_1 = [T_{c_1}, T_{c_2}, T_{c_3}, K, h_{le}, h_{te}]^\top, \boldsymbol{z}_2 = [P_{lm}]^\top, \boldsymbol{z}_3 = [\dot{m}, T_g, F_{perf}]^\top$ and $\boldsymbol{z}_4 = [F_{econ}]^\top$. In the following, we provide details on the models used for each component.

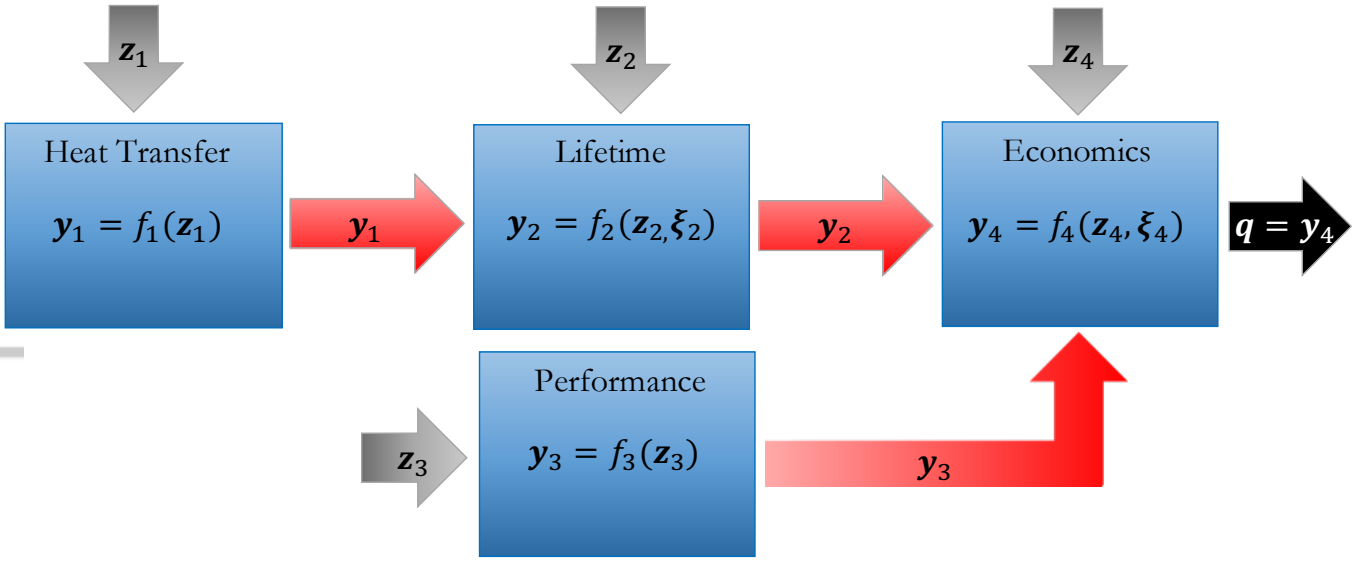

FIGURE 10 A multi-fidelity economics model of a turbine consisting of four components. Coupling variables are depicted in red, external inputs in gray and system-level QoI in black. The random variables are $\boldsymbol{z}_1 = [T_{c_1}, T_{c_2}, T_{c_3}, K, h_{le}, h_{te}]^\top$, $\boldsymbol{z}_2 = [P_{lm}]^\top$, $\boldsymbol{z}_3 = [\dot{m}, T_g, F_{perf}]^\top$ and $\boldsymbol{z}_4 = [F_{econ}]^\top$. The coupling variables are $\boldsymbol{\xi}_2 = \boldsymbol{y}_1 = [T_{bulk}]$ and $\boldsymbol{\xi}_3 = [\boldsymbol{y}_2^\top, \boldsymbol{y}_3^\top]^\top = [P_{eng}, t_{fail}]^\top$. No random variables are shared between components, that is $\boldsymbol{z}_j \cap \boldsymbol{z}_k = \emptyset \; \forall j, k$.

Heat transfer model.

The heat transfer model is used to predict the bulk temperature of a cooled turbine in the path of heated gas flow. We use quadratic finite elements to solve the stationary heat equation

$$\begin{aligned} \nabla(k \cdot \nabla h(x)) &= 0 \quad x \in \Omega \\ h(x) &= T_{c_j} \quad x \in \partial\Omega_j, \; j = 1, 2, 3 \\ h(x) &= h_{te} + (h_{le} - h_{te}) \exp\left(-4 \frac{x_1^2}{4 \times 10^{-6}}\right) \quad x \in \partial\Omega_4 \end{aligned}$$

on the blade geometry shown in Figure 2. Here we use Dirichlet boundary conditions to specify the effects of coolant running through the three blade passages. Heat transfer is imposed along the outer boundary as a function of the spatial chord-wise coordinate x. The output of this model is the bulk metal temperature

$$T_{bulk} = V^{-1} \int_\Omega h(x) dx,$$

where V is the volume of the blade.

We can solve the heat transfer model using three different meshes of increasing resolution. The number of degrees of freedom and cost (in seconds) is presented in Table 1. The other three components do not have models of varying fidelity, and we assume the cost of evaluating these components to be 0.1 seconds.

Lifetime model

The lifetime model predicts the expected time until blade failure assuming a Larson-Miller[58] nickel superalloy stress-to-failure ratio. The expected time until failure is given by

$$T_{fail} = \exp\left(P_{lm}/T_{bulk} - 20\right)$$

where P_{lm} is the Larson-Miller parameter.

Performance model

We use a simplified model to evaluate the maximum power of the turbine. Specifically, the engine performance is given by

$$P_{eng} = F_{perf}(\dot{m}_0 - N\dot{m})\, C_p\, T_0\, (1 + T_g/T_0 - 2\sqrt{T_g/T_0})$$

where the inlet compressor temperature $T_0 = 300$, the inlet compressor flow rate $\dot{m}_0 = 30$, the number of gas turbine blades $N = 90$, and the specific heat $C_p = 1003.5$ are constants, and the performance factor F_{perf}, the external gas temperature T_g and the coolant mass flow $\dot{m}$ are random parameters. The model penalizes coolant flow usage and rewards high external gas path temperatures.

Economics model

The economics model predicts the revenue from operating the gas turbine via

$$r_{econ} = F_{econ}\, t_{fail}\, P_{eng}\, (c_0/1000)$$

where $c_0 = 0.07$. The model penalizes a turbine that has a high-risk of failure and rewards high engine performance. The economic factor F_{econ} is a random parameter accounting for the variability with other gas turbine components not represented in the model.

TABLE 4 Random variables of the economics-turbine system depicted in Figure 10. The System Index denotes the index of the variable in the aggregated set of system random variables $\boldsymbol{z}$. Each variable enters the component variables $\boldsymbol{z}_k$ in the column entitled Component Variables. Arguments of the Uniform distributions are lower and upper bounds.

System Index	Random Parameter Name	Symbol	Component Variable	Distribution
1	First passage coolant temperature	T_{c_1}	$\boldsymbol{z}_1$	$\mathcal{U}[590, 610]$
2	Second passage coolant temperature	T_{c_2}	$\boldsymbol{z}_1$	$\mathcal{U}[640, 660]$
3	Third passage coolant temperature	T_{c_3}	$\boldsymbol{z}_1$	$\mathcal{U}[690, 710]$
4	Thermal conductivity	K	$\boldsymbol{z}_1$	$\mathcal{U}[29, 31]$
5	Leading edge heat transfer coefficient	h_{le}	$\boldsymbol{z}_1$	$\mathcal{U}[1975, 2025]$
6	Tail edge heat transfer coefficient	h_{te}	$\boldsymbol{z}_1$	$\mathcal{U}[975, 1025]$
7	Lars-Miller parameter	P_{lm}	$\boldsymbol{z}_2$	$\mathcal{U}[2.45 \times 10^4, 2.55 \times 10^4]$
8	Coolant mass flow rate	$\dot{m}$	$\boldsymbol{z}_3$	$\mathcal{U}[0.108, 0.132]$
9	External gas temperature	T_G	$\boldsymbol{z}_3$	$\mathcal{U}[1225, 1275]$
10	Performance factor	F_{perf}	$\boldsymbol{z}_3$	$\mathcal{U}[0.85, 0.95]$
11	Economic factor	F_{econ}	$\boldsymbol{z}_4$	$\mathcal{U}[0.9, 1.1]$

The left plot of Figure 11 compares the performance of adaptive multi-fidelity component-surrogates ("MF Integrated") with single-fidelity component-surrogates ("SF Integrated") and multi-fidelity and single-fidelity black-box models ("MF Black-box" and "SF Black-box"). Even though only one component has an ensemble of models available (unlike the previous multi-fidelity example), the multi-fidelity integrated procedure produces a significantly more accurate surrogate than the alternative approaches. As seen by comparing the resource allocations, depicted in the middle and right plots and associated respectively with the points indicated by the boxes labeled 1 and 2 in the left plot, the procedure only evaluates the finite element heat-transfer model until the error in the surrogate of that component is dominated by the errors of the other components. At that point, the error drops sharply because the costs of evaluating the other components are much smaller than the cost of running the heat transfer model. This is a major advantage of decoupling the component-models: the accuracy to which any component is resolved is commensurate with its impact on the system-level QoI. In situations when simple empirical models, such as the economic model used here, are used to inform decisions, this result suggests that the incorporation of high-resolution multi-physics models does not necessarily need to cause an explosion in system-analysis cost; rather, the computational resources used to run expensive component models can be limited to only those necessary to approximate other cheaper component models. Moreover, the required precision can be automatically determined.

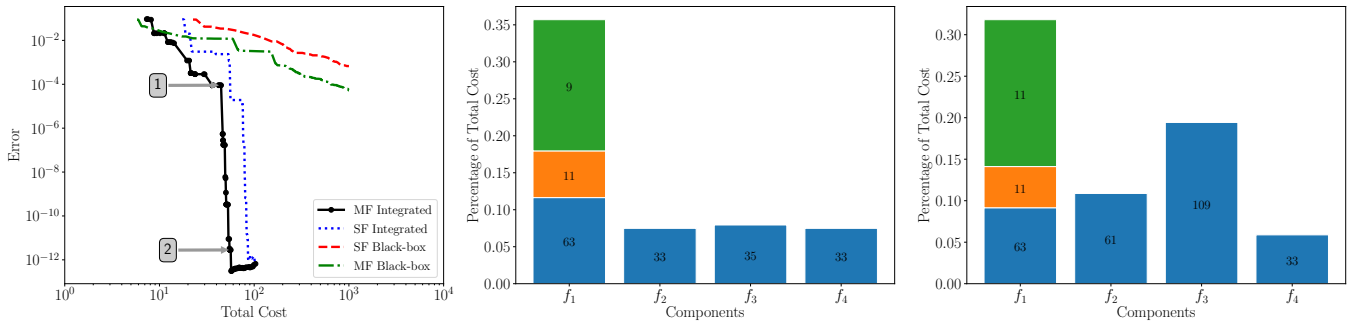

FIGURE 11 (Left) Error vs cost for the economics-turbine system-model. (Middle) Allocation of work for the MF integrated-surrogates when total cost is approximately 44 and (Right) 51. The two numbered boxes in the left plot indicate the points on the convergence curve that are associated respectively with the middle and right plots.

6 | CONCLUSIONS

This paper presented an algorithm for efficiently building surrogates for coupled/integrated multi-disciplinary systems. These surrogates can be used to significantly reduce the cost of outer-loop analyses, such as uncertainty quantification and design, which require repeated interrogation of the coupled system. The procedure introduces coupling variables with unknown distributions to allow the independent construction of surrogates for each component of a system. An adaptive sampling procedure is then used to allocate resources for training each component-surrogate in a manner that minimizes prediction error per unit cost. The proposed methodology was successfully applied to systems consisting solely of feed-forward coupling and systems with mixed feed-forward and feedback coupling.

Analysis was provided to bound the error predictions of system-level quantities of interest obtained from the integrated component-surrogates. Moreover, extensive numerical examples demonstrated that building approximations over the individual components can reduce the dimensionality and non-linearity of the surrogates being built. These properties, along with our method for adaptively allocating resources to the most important components, reduced the cost of system analysis by orders of magnitude on the examples tested. These gains were amplified when multi-fidelity models of varying accuracy and cost were available for at least one system component.

In this work we focused on scalar couplings between multi-disciplinary components. In future work we will investigate the use of our framework for coupling multi-scale, multi-physics problems that possess couplings that are infinite-dimensional random fields, for example that arise when coupling partial differential equations with a shared physical boundary. To be successful we will need to represent the field with a finite-dimensional basis, e.g., associated with a Karhunen-Loève expansion, and balance the truncation error of this expansion with the various approximation errors considered in this paper.

ACKNOWLEDGMENTS

John Jakeman, Michael Eldred and Alex Gorodetsky were supported by the Laboratory Directed Research Development (LDRD) program at Sandia National Laboratories. Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525. The views expressed in the article do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

Lorenzo Tamellini has been supported by the PRIN 2017 project 201752HKH8 "Numerical Analysis for Full and Reduced Order Methods for the efficient and accurate solution of complex systems governed by Partial Differential Equations (NA-FROM-PDEs)". Lorenzo Tamellini also acknowledges the support of GNCS-INdAM (Gruppo Nazionale Calcolo Scientifico - Istituto Nazionale di Alta Matematica).

Samuel Friedman and Douglas Allaire were supported by the AFOSR MURI on multi-information sources of multi-physics systems under Award Number FA9550-15-1-0038, program manager, Dr. Fariba Fahroo and by the National Science Foundation

under grant no. CMMI-1663130. Opinions expressed in this paper are of the authors and do not necessarily reflect the views of the National Science Foundation.

□

APPENDIX

A PROOF OF PROPOSITIONS

The following outlines the proof of Proposition 1.

Proof. First, for any $\boldsymbol{\xi}_k, \boldsymbol{\xi}_k^\star \in \Xi_k$, and $\boldsymbol{z}_k \in \Gamma_k$, we have

$$\begin{aligned}\|f_k(\boldsymbol{z}_k,\boldsymbol{\xi}_k) - f_{k,\mathcal{I}_k}(\boldsymbol{z}_k,\boldsymbol{\xi}_k^\star)\| &\le \|f_k(\boldsymbol{z}_k,\boldsymbol{\xi}_k) - f_k(\boldsymbol{z}_k,\boldsymbol{\xi}_k^\star)\| + \|f_k(\boldsymbol{z}_k,\boldsymbol{\xi}_k^\star) - f_{k,\mathcal{I}_k}(\boldsymbol{z}_k,\boldsymbol{\xi}_k^\star)\| \\ &\le L_k\|\boldsymbol{\xi}_k - \boldsymbol{\xi}_k^\star\| + \epsilon \end{aligned} \tag{A1}$$

where in the last step we have used the assumption of Lipschitz continuity to bound the first term and the definition of ϵ to bound the second one. Now without loss of generality set $\boldsymbol{z}_k = \boldsymbol{z}$ and $\text{card}(\boldsymbol{\xi}_k) = S_k = S$, $\forall k$, then by repeated use of (A1), where $\boldsymbol{\xi}_k = f_{k-1}(\cdot)$ and $\boldsymbol{\xi}_k^\star = f_{k-1,\mathcal{I}_{k-1}}(\cdot)$, we have

$$\begin{aligned}\|f(\boldsymbol{z}) - \hat{f}(\boldsymbol{z})\| &= \|f_K \circ \cdots \circ f_1(\boldsymbol{z}) - f_{K,\mathcal{I}_K} \circ \cdots \circ f_{1,\mathcal{I}_1}(\boldsymbol{z})\| \\ &\le \epsilon_K + L_K\|f_{K-1} \circ \cdots \circ f_1(\boldsymbol{z}) - f_{K-1,\mathcal{I}_{K-1}} \circ \cdots \circ f_{1,\mathcal{I}_1}(\boldsymbol{z})\| \\ &\le \epsilon_K + L_K\left(\epsilon_{K-1} + L_{K-1}\|f_{K-2} \circ \cdots \circ f_1(\boldsymbol{z}) - f_{K-2,\mathcal{I}_{K-2}} \circ \cdots \circ f_{1,\mathcal{I}_1}(\boldsymbol{z})\|\right) \\ &\le \cdots \\ &\le \epsilon_K + \epsilon_{K-1}L_K + \epsilon_{K-2}L_KL_{K-1} + \cdots + \epsilon_1\prod_{k=2}^{K} L_k \\ &\le \epsilon\left(1 + L_K + L_KL_{K-1} + \cdots + \prod_{k=2}^{K} L_k\right) \\ &\le \epsilon\left(1 + L + L^2 + \cdots L^{K-1}\right) \\ &= \epsilon\frac{1-L^K}{1-L},\end{aligned}$$

where $L = \max_{k=1,\ldots,K} L_k$ and $\epsilon = \max_{k=1,\ldots,K} \epsilon_k$. The last equality uses the well-known expression for the sum of a geometric series. □

The following outlines the proof of Proposition 2.

Proof. Recall that

$$F(\boldsymbol{\xi}) = \begin{bmatrix} f_j(\boldsymbol{z}_j,\boldsymbol{\xi}_j) \\ f_k(\boldsymbol{z}_k,\boldsymbol{\xi}_k)\end{bmatrix} \qquad \boldsymbol{\xi} = [\boldsymbol{\xi}_j,\boldsymbol{\xi}_k]^\top$$

denotes the fixed point iteration function using the true component-models and $F_{\mathcal{J}}$ similarly denotes the FPI function using the component-surrogates. Also let $\boldsymbol{\xi}^p$ and $\boldsymbol{\xi}^{p,\star}$ respectively denote the estimate of the coupling variables obtained after the p-th fixed point iteration, $p = 1, \ldots, P$, using the true component-model and its surrogate. If we initialize each FPI with the same guess so that $\boldsymbol{\xi}^0 - \boldsymbol{\xi}^{\star,0} = 0$, without loss of generality drop the dependence on $\boldsymbol{z}$, we obtain

$$\begin{aligned}\|\boldsymbol{\xi}^1 - \boldsymbol{\xi}^{\star,1}\| &= \|F(\boldsymbol{\xi}^0) - F_{\mathcal{J}}(\boldsymbol{\xi}^{\star,0})\| \\ &\le \|F(\boldsymbol{\xi}^0) - F(\boldsymbol{\xi}^{\star,0})\| + \|F(\boldsymbol{\xi}^{\star,0}) - F_{\mathcal{J}}(\boldsymbol{\xi}^{\star,0})\| \\ &\le L\|\boldsymbol{\xi}^0 - \boldsymbol{\xi}^{\star,0}\| + \epsilon = \epsilon,\end{aligned}$$

then for the second iteration we get

$$\begin{aligned}\|\xi^2 - \xi^{\star,2}\| &= \|F(\xi^1) - F_{\mathcal{J}}(\xi^{\star,1})\| \\ &\le \|F(\xi^1) - F(\xi^{\star,1})\| + \|F(\xi^{\star,1}) - F_{\mathcal{J}}(\xi^{\star,1})\| \\ &\le L\|\xi^1 - \xi^{\star,1}\| + \epsilon = L\epsilon + \epsilon,\end{aligned}$$

and finally, repeating for P iterations we obtain

$$\begin{aligned}\|\xi^P - \xi^{\star,P}\| &\le \epsilon\left(1 + L + L^2 + \cdots + L^{P-1}\right) \\ &= \epsilon\frac{1 - L^P}{1 - L}.\end{aligned}$$

□

B NOMENCLATURE

Notation	Definition
D	The number of all unique exogeneous inputs to the system
Γ	The range of all unique exogeneous inputs to the system
$\boldsymbol{z}$	All exogeneous inputs to the system, $\boldsymbol{z} \in \Gamma \subset \mathbb{R}^D$
$\rho_z(\boldsymbol{z})$	The joint density of all exogeneous inputs
$f(\boldsymbol{z})$	The black-box model returning all component outputs
$\boldsymbol{y}$	Union of all component outputs $\boldsymbol{y} = f(\boldsymbol{z})$
Q	The number of combined outputs from all components, $\boldsymbol{y} \in \mathbb{R}^Q$
Υ	The range of all component outputs, $\boldsymbol{y} \in \Upsilon \subset \mathbb{R}^Q$
Q^{sys}	The number of system-level Quantities of Interest (QoI), $Q^{sys} \le Q$
$\boldsymbol{q}$	The system-level QoI, $\boldsymbol{q} \subseteq \boldsymbol{y}$
K	The number of components in an integrated system
D_k	The number of exogeneous inputs to the k-th component
Γ_k	The range of all exogeneous inputs to the k-th component
$\boldsymbol{z}_k$	The exogeneous variables of the k-th component, $\boldsymbol{z}_k \in \Gamma_k \subset \mathbb{R}^{D_k}$
$\rho_{z,k}$	The joint PDF of the exogenous inputs of the k-th component
S_k	The number of coupling variables of the k-th component
Ξ_k	The range of all coupling variables of the k-th component
$\boldsymbol{\xi}_k$	The coupling variables of the k-th component, $\boldsymbol{\xi}_k \in \Xi_k \subset \mathbb{R}^{S_k}$
$\rho_{\xi,k}$	The joint PDF of the input coupling variables of the k-th component
$f_k(\boldsymbol{z}_k, \boldsymbol{\xi}_k)$	The model of the k-th component
Q_k	The number of outputs of the k-th component
Υ_k	The range of the outputs of the k-th component
$\boldsymbol{y}_k$	Outputs of the k-th component $\boldsymbol{y}_k = f(\boldsymbol{z}_k, \boldsymbol{\xi}_k)$
$\boldsymbol{u}_k$	Combined exogeneous and coupling variables of k-th component, $\boldsymbol{u}_k = [\boldsymbol{z}_k^\top, \boldsymbol{\xi}_k^\top]^\top$
$u_{k,n}$	n-th dimensional coordinate of $\boldsymbol{u}_k$
N_k	Number of variable in the combined variable $\boldsymbol{u}_k$, $N_k = D_k + S_k$
L_k	Lipschitz constant of the k-th component-model
$\boldsymbol{e}_{ij}$	The unit vector of length i with the j-th entry non-zero, $[0, \ldots, 0, 1, 0 \ldots, 0]^\top$
$\boldsymbol{A}_k^q$	Extraction matrix indexing component outputs $\boldsymbol{y}_k$ into system QoI $\boldsymbol{q}$
$\boldsymbol{A}_k^z$	Extraction matrix indexing component exogeneous inputs $\boldsymbol{z}_k$ into system inputs $\boldsymbol{z}$
$\boldsymbol{A}_k^\xi$	Extraction matrix indexing component coupling variables $\boldsymbol{\xi}_k$ into system outputs $\boldsymbol{y}$
$F(\boldsymbol{\xi})$	Fixed point iteration function used to solve for consistent coupling variables $\boldsymbol{\xi}$
$\boldsymbol{\xi}^p$	Value of the coupling variables at the p-th fixed point iteration
P	number of FP iterations
η	Fixed point iteration tolerance

R_k	Number of hyper-parameters controlling deterministic fidelity of k-th component-model
$\boldsymbol{\alpha}$	Multi-index specifying the deterministic fidelity of a model, $\boldsymbol{\alpha} \in \mathbb{N}^{R_k}$
$\boldsymbol{\alpha} \leq \boldsymbol{\gamma}$	component-wise inequality between vectors: $\boldsymbol{\alpha} \leq \boldsymbol{\gamma}$ if $\exists j$ s.t. $\alpha_j < \gamma_j$ and $\alpha_i = \gamma_i$ for $i \neq j$.
$f_{k,\boldsymbol{\alpha}}(\boldsymbol{z}_k, \boldsymbol{\xi}_k)$	k-th component-model with deterministic fidelity $\boldsymbol{\alpha}$.
$\boldsymbol{\beta}$	Multi-index specifying the parametric fidelity of a single-fidelity surrogate
$[\boldsymbol{\alpha}, \boldsymbol{\beta}]$	Concatenation of indices $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$
$f_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}(\boldsymbol{z}_k, \boldsymbol{\xi}_k)$	Single-fidelity surrogate of k-th component with deterministic/parametric fidelities $\boldsymbol{\alpha}, \boldsymbol{\beta}$
$\boldsymbol{u}^{(m)}_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$	Sample of the combined inputs to the k-th component used to build $f_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}(\boldsymbol{z}_k, \boldsymbol{\xi}_k)$
$\mathcal{U}_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$	Set of samples used to construct $f_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}(\boldsymbol{u}_k)$
$\mathcal{Y}_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$	Evaluations of $f_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$ at each sample in $\mathcal{U}_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$
$M_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$	Number of samples in $\mathcal{U}_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$ which is independent of $\boldsymbol{\alpha}$
$\mathcal{I}_k$	Set of indices $[\boldsymbol{\alpha}, \boldsymbol{\beta}]$, specifying fidelities of multiple surrogates of k-th component
$f_{k,\mathcal{I}_k}(\boldsymbol{z}_k, \boldsymbol{\xi}_k)$	Multi-fidelity surrogate of k-th combining fidelities specified by $\mathcal{I}_k$
$\hat{\Xi}_k$	Estimated ranges of the coupling variables used to build surrogate of the k-th component
$y_{k,q}$	q-th output of the k-th component-model
$y_{k,\mathcal{I}_k,q}$	q-th output of the k-th component-surrogate
$f_{k,q}(\boldsymbol{z}_k, \boldsymbol{\xi}_k)$	the single-fidelity (α ignored) function returning the q-th output of the k-th component-model
$f_{k,\mathcal{I}_k,q}(\boldsymbol{z}_k, \boldsymbol{\xi}_k)$	the single-fidelity (α ignored) function returning the q-th output of the k-th component-surrogate
$c_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$	A coefficient of the MISC approximation for the k-th component
$\mathcal{C}_k$	The set of all coefficients of the MISC approximation of the k-th component
$\{0,1\}^N$	Set consisting of size N consisting of ones or zeros
$m(\beta_n)$	Function determining number of univariate interpolation points from β_n
$u^{(j)}_{n,\boldsymbol{\alpha},\beta_n}$	Univariate interpolation point for fixed $\boldsymbol{\alpha}, \boldsymbol{\beta}$
$\mathcal{U}_{k,n,\boldsymbol{\alpha},\beta_n}$	Set of univariate interpolation points, $\mathcal{U}_{k,n,\boldsymbol{\alpha},\beta_n} = \{u^{(j)}_{n,\boldsymbol{\alpha},\beta_n}\}_{j=1}^{m(\beta_n)}$ for fixed $\boldsymbol{\alpha}, \boldsymbol{\beta}$
$\mathcal{L}^{(j)}_{k,n,\boldsymbol{\alpha},\beta_n}(u_n)$	Univariate Lagrange polynomial for for fixed $\boldsymbol{\alpha}, \boldsymbol{\beta}$
$\mathcal{U}_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$	Tensor-product grid of univariate interpolation point sets needed to construct $f_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$
τ	Accuracy termination tolerance for Algorithm 1
$W_{\max}$	Maximum work allowed for Algorithm 1
$\gamma_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$	Error indicator quantifying effect of adding surrogate the $f_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$ of the k-th component
$\mathcal{R}_k$	Set of possible indices $[\boldsymbol{\alpha}, \boldsymbol{\beta}]$ to add to the multi-fidelity surrogate of the k-th component
$\mathcal{E}_k$	Set of error indicators associated with each $[\boldsymbol{\alpha}, \boldsymbol{\beta}]$ in $\mathcal{R}_k$
$\mathcal{F}_k$	Set of all model fidelities $f_{k,\boldsymbol{\alpha}}$ for the k-th component
$\mathcal{J}$	Set of multi-fidelity index sets, $\mathcal{J} = \{\mathcal{I}_1, \ldots, \mathcal{I}_K\}$
T_k	total size of the concatenated multi-indices $[\boldsymbol{\alpha}\,\boldsymbol{\beta}]$
$f_{\mathcal{J}}(\boldsymbol{z})$	Multi-fidelity surrogate of the system outputs model $f(\boldsymbol{z})$
$\Delta E_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$	Change in estimates of system output incurred by updating surrogate of k-th component
$\Delta W_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$	Work needed to update the surrogate of the k-th component
$\|\cdot\|_1$	ℓ^1 norm
$\mathrm{card}(\cdot)$	Cardinality (number of elements) of a set
I_k	interval over which a generic Leja sequence is built
$\mathcal{Z}_{\mathrm{refine}}$	Samples of system exogeneous variables used to compute error indicator
L_{refine}	Number of samples of system exogeneous variables used to compute error indicator, i.e., cardinality of $\mathcal{Z}_{\mathrm{refine}}$
$f_{\mathcal{J}^k_{\boldsymbol{\alpha},\boldsymbol{\beta}}}$	Multi-fidelity system surrogate with the index $[\boldsymbol{\alpha}, \boldsymbol{\beta}]$ added to the k-th component-surrogate

References

1. Ghanem R, Spanos P. *Stochastic Finite Elements: A Spectral Approach*. New York, NY, USA: Springer-Verlag New York, Inc. . 1991.

2. Sudret B. Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering & System Safety* 2008; 93(7): 964–979. doi: 10.1016/i.ress.2007.04.002

3. Xiu D, Karniadakis G. The Wiener-Askey Polynomial Chaos for stochastic differential equations. *SIAM J. Sci. Comput.* 2002; 24(2): 619–644. doi: 10.1137/S1064827501387826

4. Rasmussen CE, Williams CKI. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press . 2005.

5. Sacks J, Welch WJ, Mitchell TJ, Wynn HP. Design and Analysis of Computer Experiments. *Statistical Science* 1989; 4(4): 409–423.

6. Harbrecht H, Jakeman J, Zaspel P. Cholesky-Based Experimental Design for Gaussian Process and Kernel-Based Emulation and Calibration. *Communications in Computational Physics* 2021; 29(4): 1152–1185. doi: 10.4208/cicp.OA-2020-0060

7. Doostan A, Validi A, Iaccarino G. Non-intrusive low-rank separated approximation of high-dimensional stochastic models. *Computer Methods in Applied Mechanics and Engineering* 2013; 263: 42-55. doi: 10.1016/j.cma.2013.04.003

8. Gorodetsky A, Jakeman J. Gradient-based optimization for regression in the functional tensor-train format. *Journal of Computational Physics* 2018; 374: 1219 - 1238. doi: 10.1016/j.jcp.2018.08.010

9. Oseledets IV. Tensor-Train Decomposition. *SIAM Journal on Scientific Computing* 2011; 33(5): 2295-2317. doi: 10.1137/090752286

10. Nobile F, Tempone R, Webster C. A Sparse Grid Stochastic Collocation Method for Partial Differential Equations with Random Input Data. *SIAM Journal on Numerical Analysis* 2008; 46(5): 2309–2345. doi: 10.1137/060663660

11. Xiu D, Hesthaven J. High-Order Collocation Methods for Differential Equations with Random Inputs. *SIAM Journal on Scientific Computing* 2005; 27(3): 1118-1139. doi: 10.1137/040615201

12. Jakeman J, Roberts S. Local and Dimension Adaptive Stochastic Collocation for Uncertainty Quantification. In: Garcke J, Griebel M., eds. *Sparse Grids and Applications*. 88 of *Lecture Notes in Computational Science and Engineering*. Springer Berlin Heidelberg. 2013 (pp. 181-203)

13. Chen P, Quarteroni A, Rozza G. Comparison Between Reduced Basis and Stochastic Collocation Methods for Elliptic Problems. *Journal of Scientific Computing* 2014; 59(1): 187–216. doi: 10.1007/s10915-013-9764-2

14. Elman HC, Liao Q. Reduced Basis Collocation Methods for Partial Differential Equations with Random Coefficients. *SIAM/ASA Journal on Uncertainty Quantification* 2013; 1(1): 192-217. doi: 10.1137/120881841

15. Manzoni A, Pagani S, Lassila T. Accurate Solution of Bayesian Inverse Uncertainty Quantification Problems Combining Reduced Basis Methods and Reduction Error Models. *SIAM/ASA Journal on Uncertainty Quantification* 2016; 4(1): 380-412. doi: 10.1137/140995817

16. Rozza G, Huynh DBP, Patera AT. Reduced Basis Approximation and a Posteriori Error Estimation for Affinely Parametrized Elliptic Coercive Partial Differential Equations. *Archives of Computational Methods in Engineering* 2008; 15(3): 229. doi: 10.1007/s11831-008-9019-9

17. Zhu Y, Zabaras N. Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification. *Journal of Computational Physics* 2018; 366: 415-447. doi: 10.1016/j.jcp.2018.04.018

18. Qin T, Chen Z, Jakeman J, Xiu D. DEEP LEARNING OF PARAMETERIZED EQUATIONS WITH APPLICATIONS TO UNCERTAINTY QUANTIFICATION. *International Journal for Uncertainty Quantification* 2021; 11(2): 63–82. doi: 10.1615/Int.J.UncertaintyQuantification.2020034123

19. Arnst M, Ghanem R, Phipps E, Red-Horse J. Measure transformation and efficient quadrature in reduced-dimensional stochastic modeling of coupled problems. *International Journal for Numerical Methods in Engineering* 2012; 92(12): 1044–1080. doi: 10.1002/nme.4368

20. Amaral S, Allaire D, Willcox K. A decomposition-based approach to uncertainty analysis of feed-forward multicomponent systems. *International Journal for Numerical Methods in Engineering* 2014; 100(13): 982–1005. doi: 10.1002/nme.4779

21. Constantine P, Phipps E, Wildey T. Efficient uncertainty propagation for network multiphysics systems. *International Journal for Numerical Methods in Engineering* 2014; 99(3): 183–202. doi: 10.1002/nme.4667

22. Sankararaman S, Mahadevan S. Likelihood-Based Approach to Multidisciplinary Analysis Under Uncertainty. *Journal of Mechanical Design* 2012; 134(3). doi: 10.1115/1.4005619

23. Mittal A, Chen X, Tong CH, Iaccarino G. A Flexible Uncertainty Propagation Framework for General Multiphysics Systems. *SIAM/ASA Journal on Uncertainty Quantification* 2016; 4(1): 218-243. doi: 10.1137/140981411

24. Carlberg K, Guzzetti S, Khalil M, Sargsyan K. The network uncertainty quantification method for propagating uncertainties in component-based systems. *arXiv* 2020.

25. Chaudhuri A, Lam R, Willcox K. Multifidelity Uncertainty Propagation via Adaptive Surrogates in Coupled Multidisciplinary Systems. *AIAA Journal* 2018; 56(1): 235-249. doi: 10.2514/1.J055678

26. Friedman S, Isaac B, Ghoreishi SF, Allaire DL. Efficient decoupling of multiphysics systems for uncertainty propagation. *Proceedings of the AIAA SciTech Forum* 2018. doi: 10.2514/6.2018-1661

27. Isaac B, Friedman S, Allaire DL. Efficient approximation of coupling variable fixed point sets for decoupling multidisciplinary systems. *Proceedings of the AIAA SciTech Forum* 2018. doi: 10.2514/6.2018-1908

28. Kyzyurova KN, Berger JO, Wolpert RL. Coupling Computer Models through Linking Their Statistical Emulators. *SIAM/ASA Journal on Uncertainty Quantification* 2018; 6(3): 1151-1171. doi: 10.1137/17M1157702

29. Sanson F, Maitre OL, Congedo PM. Systems of Gaussian process models for directed chains of solvers. *Computer Methods in Applied Mechanics and Engineering* 2019; 352: 32 - 55. doi: 10.1016/j.cma.2019.04.013

30. Jakeman J, Eldred M, Geraci G, Gorodetsky A. Adaptive Multi-index Collocation for Uncertainty Quantification and Sensitivity Analysis. *International Journal for Numerical Methods in Engineering* 2019. doi: 10.1002/nme.6268

31. Haji-Ali A, Nobile F, Tamellini L, Tempone R. Multi-Index Stochastic Collocation for random PDEs. *Computer Methods in Applied Mechanics and Engineering* 2016; 306: 95 - 122. doi: 10.1016/j.cma.2016.03.029

32. Beck J, Tamellini L, Tempone R. IGA-based multi-index stochastic collocation for random PDEs on arbitrary domains. *Computer Methods in Applied Mechanics and Engineering* 2019; 351: 330-350. doi: 10.1016/j.cma.2019.03.042

33. Piazzola C, Tamellini L, Pellegrini R, Broglia R, Serani A, Diez M. Comparing Multi-Index Stochastic Collocation and Multi-Fidelity Stochastic Radial Basis Functions for Forward Uncertainty Quantification of Ship Resistance. *Arxiv e-prints* 2021(2106.00591).

34. Haji-Ali AL, Nobile F, Tamellini L, Tempone R. Multi-index Stochastic Collocation convergence rates for random PDEs with parametric regularity. *Foundations of Computational Mathematics* 2016; 16(6): 1555–1605. doi: 10.1007/s10208-016-9327-7

35. Smetana K, Patera AT. Optimal Local Approximation Spaces for Component-Based Static Condensation Procedures. *SIAM Journal on Scientific Computing* 2016; 38(5): A3318-A3356. doi: 10.1137/15M1009603

36. Eigel M, Gruhlke R. A local hybrid surrogate-based finite element tearing interconnecting dual-primal method for nonsmooth random partial differential equations. *International Journal for Numerical Methods in Engineering* 2021; 122(4): 1001-1030. doi: 10.1002/nme.6571

37. Mu L, Zhang G. A Domain Decomposition Model Reduction Method for Linear Convection-Diffusion Equations with Random Coefficients. *SIAM Journal on Scientific Computing* 2019; 41(3): A1984-A2011. doi: 10.1137/18M1170601

38. Contreras AA, Mycek P, Le Maître OP, Rizzi F, Debusschere B, Knio OM. Parallel Domain Decomposition Strategies for Stochastic Elliptic Equations Part B: Accelerated Monte Carlo Sampling with Local PC Expansions. *SIAM Journal on Scientific Computing* 2018; 40(4): C547-C580. doi: 10.1137/17M1132197

39. Amsallem D, Zahr MJ, Farhat C. Nonlinear model order reduction based on local reduced-order bases. *International Journal for Numerical Methods in Engineering* 2012; 92(10): 891–916. doi: 10.1002/nme.437

40. Peherstorfer B, Butnaru D, Willcox K, Bungartz HJ. Localized Discrete Empirical Interpolation Method. *SIAM Journal on Scientific Computing* 2014; 36(1): A168-A192. doi: 10.1137/130924408

41. Buhr A, Iapichino L, Ohlberger M, Rave S, Schindler F, Smetana K. *Localized model reduction for parameterized problems*; Germany: Walter De Gruyter . 2019.

42. Chen Y, Jakeman J, Gittelson C, Xiu D. Local Polynomial Chaos Expansion for Linear Differential Equations with High Dimensional Random Inputs. *SIAM Journal on Scientific Computing* 2015; 37(1): A79-A102. doi: 10.1137/140970100

43. Granas A, Dugundji J. Fixed Point Theory. *Springer Monographs in Mathematics* 2003. doi: 10.1007/978-0-387-21593-8

44. Rogers J. *DeMAID/GA - An enhanced design manager's aid for intelligent decomposition*; 2012

45. Marque-Pucheu, Sophie , Perrin, Guillaume , Garnier, Josselin . Efficient sequential experimental design for surrogate modeling of nested codes. *ESAIM: PS* 2019; 23: 245-270. doi: 10.1051/ps/2018011

46. Chen X, Ng B, Sun Y, Tong C. A flexible uncertainty quantification method for linearly coupled multi-physics systems. *Journal of Computational Physics* 2013; 248: 383 - 401. doi: 10.1016/j.jcp.2013.04.009

47. Jakeman J, Eldred M, Xiu D. Numerical approach for quantification of epistemic uncertainty. *Journal of Computational Physics* 2010; 229(12): 4648–4663. doi: DOI: 10.1016/j.jcp.2010.03.003

48. Chen X, Park EJ, Xiu D. A flexible numerical approach for quantification of epistemic uncertainty. *Journal of Computational Physics* 2013; 240: 211 - 224. doi: 10.1016/j.jcp.2013.01.018

49. Narayan A, Jakeman JD. Adaptive Leja sparse grid constructions for stochastic collocation and high-dimensional approximation. *SIAM/J. Sci. Comput* 2014; 36(6): 2952–2983. doi: 10.1137/140966368

50. Ernst OG, Sprungk B, Tamellini L. On Expansions and Nodes for Sparse Grid Collocation of Lognormal Elliptic PDEs. *Arxiv e-prints* 2019(1906.01252).

51. Bungartz HJ, Griebel M. Sparse grids. *Acta Numerica* 2004; 13: 1–123. doi: 10.1017/S0962492904000182

52. Nobile F, Tamellini L, Tempone R. Convergence of quasi-optimal sparse-grid approximation of Hilbert-space-valued functions: application to random elliptic PDEs. *Numerische Mathematik* 2016; 134(2): 343–388. doi: 10.1007/s00211-015-0773-y

53. Eigel M, Ernst OG, Sprungk B, Tamellini L. On the convergence of adaptive stochastic collocation for elliptic partial differential equations with affine diffusion. *Arxiv e-prints* 2020(2008.07186).

54. Chkifa A, Cohen A, Schwab C. High-Dimensional Adaptive Sparse Polynomial Interpolation and Applications to Parametric PDEs. *Foundations of Computational Mathematics* 2014; 14(4): 601-633. doi: 10.1007/s10208-013-9154-z

55. Jakeman JD. PyApprox: Probabilistic analysis and approximation of data and simulation. https://sandialabs.github.io/pyapprox/index.html; 2021.

56. Zaman K, Mahadevan S. Robustness-Based Design Optimization of Multidisciplinary System Under Epistemic Uncertainty. *AIAA Journal* 2013; 51(5): 1021-1031. doi: 10.2514/1.J051372

57. Li K, Allaire D. A compressed sensing approach to uncertainty propagation for approximately additive functions. *Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* 2016. doi: 10.1115/DETC2016-60195

58. Reed R. *The Superalloys: Fundamentals and applications*. New York: Cambridge University Press . 2006.