SAND2021-6806R

PREPRINT

# Adaptive resource allocation for surrogate modeling of systems comprised of multiple disciplines with varying fidelity

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

[1]J. Mike Walker '66 Department of Mechanical Engineering, Texas A&M University, College Station, TX, USA
[2]Optimization and Uncertainty Quantification, Sandia National Laboratories, Albuquerque, NM, 87123
[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

## ABSTRACT

We present an adaptive algorithm for constructing surrogate models for integrated systems composed of a set of coupled components. With this goal we introduce 'coupling' variables with a priori unknown distributions that allow approximations of each component to be built independently. Once built, the surrogates of the components are combined and used to predict system-level quantities of interest (QoI) at a fraction of the cost of interrogating the full system model. We use a greedy experimental design procedure, based upon a modification of Multi-Index Stochastic Collocation (MISC), to minimize the error of the combined surrogate. This is achieved by refining each component surrogate in accordance with its relative contribution to error in the approximation of the system-level QoI. Our adaptation of MISC is a multi-fidelity procedure that can leverage ensembles of models of varying cost and accuracy, for one or more components, to produce estimates of system-level QoI. Several numerical examples demonstrate the efficacy of the proposed approach on systems involving feed-forward and feedback coupling. For a fixed computational budget, the proposed algorithm is able to produce approximations that are orders of magnitude more accurate than approximations that treat the integrated system as a black-box.

## ARTICLE INFO

Correspondence:
J.D. Jakeman
jdjakem@sandia.gov

Keywords:
Uncertainty quantification, integrated systems, surrogate, experimental design, dimension reduction

## 1 Introduction

Modeling complex systems often involves integrating numerous components from multiple disciplines. The components of the system are coupled by using the outputs of one or more components as the inputs to others. Uncertainty analysis of such integrated system models is often challenging: each component of the system model may utilize a high-dimensional, high-fidelity model that is computationally expensive to run, and when coupled together may render system-level analysis computationally intractable.

Surrogate methods, such as polynomial chaos [18, 49, 51], Gaussian processes [41, 45, 24], low-rank decompositions [13, 20, 38], sparse grid interpolation [37, 50, 29], reduced basis approximations [7, 15, 32, 44]

and neural networks [53, 40] 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.

The benefits of exploiting the structure of coupled systems have been demonstrated in a number of recent works [3, 1, 11, 46, 34, 5]. These methods significantly reduce the cost of outer-loop applications that require repeated interrogation of the coupled system, such as uncertainty quantification and optimization. The gains are obtained by decomposing the system analysis into analyses of individual components that can be combined to make system-level predictions.[1] Advances have been made for integrated systems with feed-forward and feedback coupling. Feed-forward coupling refers to the one-directional 'feeding forward' of information from an upstream component to a downstream component. In contrast, feedback coupling is present when the state of all components affects the state of all other components, indicating the presence of feedback among components.

In this paper, we focus on surrogate-based methods that decouple the high-dimensional integrated model into a set of (typically) lower-dimensional multi-fidelity component models, for example [2, 10, 34, 47, 30, 33, 8, 17, 25]. Each component model is a function of both external 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 output of other components (Figure 1). The $\boldsymbol{\xi}$ are functions of the random variables $\boldsymbol{z}$ and are also therefore random, but their distributions are unknown prior to simulation. Once surrogates of each component have been constructed, they can be used in an integrated fashion to predict system-level Quantities of Interest (QoI) by accurately determining the values of the coupling variables for a given realization of the random variables. In this paper, fixed-point iteration is used to determine the state of components with feedback coupling. The values of the feed-forward coupling variables are obtained by passing the outputs of upstream component surrogates to the inputs of the downstream components.

Significant computational benefits can be observed by treating the coupling variables as inputs and building individual surrogates for each component as functions of both the external inputs and the coupling variables. Specifically, the introduction of coupling variables effectively decouples the component surrogates, allowing them to be constructed independently. Moreover, the surrogates of each component are often lower-dimensional and less non-linear than a black-box surrogate of the integrated system.

Little attention has been given to formulating experimental design strategies in the context of coupled systems [6, 47, 19] that minimize the error in the system prediction for a fixed computational budget. In this work, we propose an adaptive experimental design strategy, based on Adaptive Multi-Index Stochastic Collocation (MISC) [28], which greedily chooses the most cost-effective component surrogate to refine. The method selects the set of training data that, when added to a component surrogate, produces the greatest reduction in the error in the approximation of the system-level QoI, relative to the cost incurred by evaluating the set of component data. The algorithm requires bounds on the coupling variables $\boldsymbol{\xi}$ and in this paper we embed an adaptive procedure for estimating the ranges of the coupling variables within our sampling algorithm.

The sampling algorithm proposed in this paper significantly reduces the cost of building surrogates for integrated system models by optimizing the investment in the constituent components. These gains are 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, which is an extension of MISC [22, 4, 39, 23], 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. MISC reduces the cost of constructing surrogates of a black-box multi-fidelity model by orders of magnitude. Our numerical results demonstrate that similar advantages can be obtained when constructing system-level predictions composed of component surrogates.

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

[1] This class of methods is distinct from stochastic domain decomposition methods for solving partial differential equations, such as those discussed in [48, 14, 35, 12].

demonstrated using a number of numerical examples in Section 5 and conclusions are presented in Section 6.

## 2 Evaluating integrated systems

In this section we discuss how to make predictions of a system of coupled components.

### 2.1 Problem formulation

This paper is concerned with the prediction of integrated systems with coupled components. With this goal, let

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

denote the map from the uncertain system parameters $\boldsymbol{z} = (z_1, \ldots, z_D)^\top \in \Gamma \subseteq \mathbb{R}^D$, with probability density (PDF) $\rho(\boldsymbol{z})$, to a set of $Q$ outputs $\boldsymbol{y} = [y_1, \ldots, y_Q]^\top \in \Upsilon \subseteq \mathbb{R}^Q$. Assume that the integrated system consists of $K$ components represented by models of the form

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

Each component returns a vector of $Q_k$ outputs $\boldsymbol{y}_k = [y_{k,1}, \ldots, y_{k,Q_k}]^\top \in \Upsilon_k \subseteq \mathbb{R}^{Q_k}$, $\Upsilon_k \subseteq \Upsilon$ and is a function of a vector of $D_k$ random variables $\boldsymbol{z}_k \in \Gamma_k \subseteq \mathbb{R}^{D_k}$, $\Gamma_k \subseteq \Gamma$ and a set of $S_k$ coupling variables $\boldsymbol{\xi}_k \in \Xi_k \subseteq \mathbb{R}^{S_k}$, $\Xi_k \subseteq \Upsilon$ which are a subset of the outputs produced by other components. We define the outputs $\boldsymbol{y}$ in (1) to be the collection of all component outputs, i.e., $\boldsymbol{y} = \bigcup_{k=1}^K \boldsymbol{y}_k$.

Following [5] we use adjacency matrices to encode the relationships between the system outputs and inputs of the aggregated system and of each component. Specifically, the random and coupling variables of the $k$-th component satisfy

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

where $\boldsymbol{R}_k^z \in \mathbb{R}^{D_k \times D}$ and $\boldsymbol{R}_k^\xi \in \mathbb{R}^{S_k \times Q}$ consist of unit row vectors that select a subset of entries from the vectors they are applied to. This encoding allows random variables to be unique to a single component or shared between components so that $D \le \sum_{k=1}^K D_k$. Similarly, this formulation allows for three cases relating outputs to coupling variables. More specifically, the $q$-th output of the $k$-th component can be a coupling variable of zero, one, or multiple components.

Finally, one is often not interested in predicting all the outputs of each component, but rather a smaller subset of the QoI. We extract these system-level QoI via

$$\boldsymbol{q} = \boldsymbol{R}^y \boldsymbol{y} \in \mathbb{R}^{Q^{\text{sys}}} \qquad\qquad \boldsymbol{R}^y \in \mathbb{R}^{Q^{\text{sys}} \times Q}. \tag{3}$$

These different cases are all depicted in the example multi-component system shown in Figure 1. Letting $\boldsymbol{e}_{i,j} = [0, \ldots, 0, 1, 0 \ldots, 0]^\top$ denote the unit vector of length $i$ with the $j$-th entry equal to 1, and the combined component outputs be $\boldsymbol{y} = [y_{1,1}, y_{1,2}, y_{1,3}, y_{1,4}, y_{2,1}, y_{2,2}, y_{2,3}, y_{2,4}, y_{3,1}, y_{3,2}]^\top$, where the first index denotes the component and the second index denotes the QoI from that component, then

$$\boldsymbol{R}_1^z = \begin{bmatrix} \boldsymbol{e}_{D,1}^\top \\ \boldsymbol{e}_{D,2}^\top \end{bmatrix} \quad \boldsymbol{R}_2^z = \begin{bmatrix} \boldsymbol{e}_{D,3}^\top \\ \boldsymbol{e}_{D,4}^\top \end{bmatrix} \quad \boldsymbol{R}_3^z = \begin{bmatrix} \boldsymbol{e}_{D,1}^\top \\ \boldsymbol{e}_{D,4}^\top \\ \boldsymbol{e}_{D,5}^\top \\ \boldsymbol{e}_{D,6}^\top \\ \boldsymbol{e}_{D,7}^\top \\ \boldsymbol{e}_{D,8}^\top \end{bmatrix} \quad \boldsymbol{R}_1^\xi = \begin{bmatrix} \emptyset \end{bmatrix} \quad \boldsymbol{R}_2^\xi = \begin{bmatrix} \boldsymbol{e}_{Q,2}^\top \\ \boldsymbol{e}_{Q,3}^\top \\ \boldsymbol{e}_{Q,9}^\top \end{bmatrix} \quad \boldsymbol{R}_3^\xi = \begin{bmatrix} \boldsymbol{e}_{Q,1}^\top \\ \boldsymbol{e}_{Q,2}^\top \\ \boldsymbol{e}_{Q,4}^\top \\ \boldsymbol{e}_{Q,5}^\top \\ \boldsymbol{e}_{Q,6}^\top \end{bmatrix} \quad \boldsymbol{R}^y = \begin{bmatrix} \boldsymbol{e}_{Q,7}^\top \\ \boldsymbol{e}_{Q,8}^\top \\ \boldsymbol{e}_{Q,10}^\top \end{bmatrix}$$

In the remainder of this section we show how to interpret these adjacency matrices to make statements about the nature of the random variables, coupling variables and system-level QoI.

For this example, the components $k = 1, 2, 3$ have 2, 2 and 6 random variables respectively, specifically $\boldsymbol{z}_1 = [z_1, z_2]^\top$, $\boldsymbol{z}_2 = [z_3, z_4]^\top$, $\boldsymbol{z}_3 = [z_1, z_4, z_5, z_6, z_7, z_8]^\top$. Components 1 and 2 share no common random

variables, i.e. $\boldsymbol{z}_1 \cap \boldsymbol{z}_2 = \emptyset$, whereas components 2 and 3 share one common random variable, i.e. $\boldsymbol{z}_2 \cap \boldsymbol{z}_3 = [z_4]$, as do components 1 and 3, i.e. $\boldsymbol{z}_1 \cap \boldsymbol{z}_3 = [z_1]$. The physical interpretation of the random variables are given in Tables 2 and 3. The entries in the System Index column correspond to the second subscript of the $e_{D,i}$ and $e_{Q,i}$ column vectors above.

For any system, a subset of outputs from one component may be inputs to zero-, one- or more components. Here $\boldsymbol{\xi}_1 = [\emptyset]$, $\boldsymbol{\xi}_2 = [y_{1,2}, y_{1,3}, y_{3,1}]^\top$, $\boldsymbol{\xi}_3 = [y_{1,1}, y_{1,2}, y_{1,4}, y_{2,1}, y_{2,2}]^\top$. Component 1 has no incoming coupling variables $\boldsymbol{\xi}_1 = \emptyset$. A subset $[y_{1,2}]$ of the outputs of the first component are 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. Finally, this example system has $Q^{\text{sys}} = 3$ system-level QoI, which are outputs of the second and third component, specifically $\boldsymbol{q} = [y_{2,3}, y_{2,4}, y_{3,2}]^\top$.

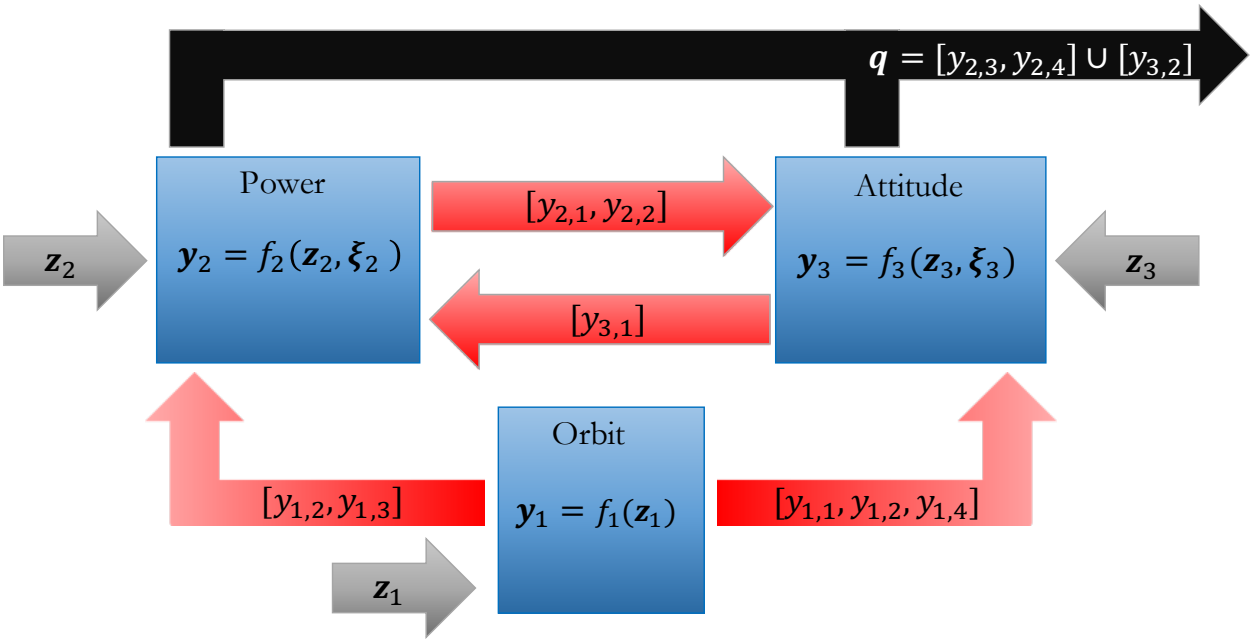


Figure 1: A fire detection satellite system consisting of three components. Coupling variables are depicted in red, external inputs in gray and system-level QoI in black. Here $\boldsymbol{z}_1 \cap \boldsymbol{z}_2 = \emptyset, \boldsymbol{z}_2 \cap \boldsymbol{z}_3 = [z_4]$, $\boldsymbol{z}_1 \cap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

## 2.2 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. In the remainder of this paper we will denote the process evaluating the system outputs $\boldsymbol{y}$, at a realization (sample) of the system random variables $\boldsymbol{z}$ with the functional

$$G_{\boldsymbol{z}} : (f_1, \ldots, f_K) \mapsto f(\boldsymbol{z}).$$

Evaluations of all component outputs are thus denoted $\boldsymbol{y} = G_{\boldsymbol{z}}[f_1, \ldots, f_K]$. Similarly, evaluations of system-level QoI are denoted $\boldsymbol{q} = \boldsymbol{R}^y G_{\boldsymbol{z}}[f_1, \ldots, f_K]$.

### 2.2.1 Feed-forward coupling

Feed-forward coupling refers to the situation when the output(s) of a component are inputs 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:

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

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 $f = f_1 \circ f_2 \circ \cdots \circ f_K$. It is common for multiple components to be inputs to another component. For these general situations information can be passed through the system of surrogates by traversing a directed acyclic graph.

### 2.2.2 Feedback coupling

Without loss of generality, consider two components with feedback coupling

$$\begin{cases} \boldsymbol{y}_j = f_j(\boldsymbol{z}_j, \boldsymbol{\xi}_j), & \boldsymbol{\xi}_j = \boldsymbol{y}_k \\ \boldsymbol{y}_k = f_k(\boldsymbol{z}_k, \boldsymbol{\xi}_k), & \boldsymbol{\xi}_j = \boldsymbol{y}_j \end{cases} \tag{4}$$

The coupling between components 2 and 3 in Figure 1 are an example of feedback coupling, where in (4) 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 for $\boldsymbol{z}_j$ and $\boldsymbol{z}_k$. Using the iteration function

$$F(\boldsymbol{\xi}) = \begin{bmatrix} f_j(\boldsymbol{z}_j, \boldsymbol{\xi}_j) \\ f_k(\boldsymbol{z}_k, \boldsymbol{\xi}_k) \end{bmatrix} \qquad \boldsymbol{\xi} = [\boldsymbol{\xi}_j, \boldsymbol{\xi}_k]^\top \tag{5}$$

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

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

until $\|\boldsymbol{\xi}^t - \boldsymbol{\xi}^{t-1}\| < \nu$, for some accuracy tolerance $\nu \geq 0$. In this paper we assume that the iterating function $F$ in (5) is a contraction which guarantees convergence of FPI [21].

When a system consists of both feed-forward and feedback coupling, we proceed by partitioning the network into groups of components that, when considered together, transform the model into a feed-forward network; 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) [43]. 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 Component surrogates for integrated systems

The goal of this paper is to present a method for constructing surrogates for the coupled system model in (1), building upon the methods in [3, 10, 11]. With this goal, we seek an approximation (surrogate model) of each component of the system, with explicit functional dependence on the coupling variables, i.e., on the $(D_k + S_k)$-variate maps in (2). 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.

Gaussian processes [47, 30, 33] and polynomial chaos expansions [8] have been used to generate decoupled surrogates in the past. In this work, we choose instead to use an approach based upon adaptive versions [28, 39] of MISC [4, 22, 23] because it provides the important features necessary to develop an experimental design strategy for allocating resources to components of integrated systems. The details of our specific algorithm is presented in Section 4, but first, in this section we discuss important considerations that impact the accuracy of predictions made using surrogates of components constructed using any approximation strategy.

## 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* $\boldsymbol{z}$. We also assume 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)*. 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 an integrated 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 $f_{k,\boldsymbol{\alpha}}(\boldsymbol{z})$. The entries $\alpha_i \in \{1, \ldots, l_{k,\alpha_i}\}$, where $l_{k,\alpha_i} \in \mathbb{N}$, dictate the value of each hyper-parameter. 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 $\boldsymbol{\alpha}^\star \geq \boldsymbol{\alpha}$, that is if $\exists j$ s.t. $\boldsymbol{\alpha}^\star_j > \boldsymbol{\alpha}_j$ and $\boldsymbol{\alpha}^\star_i = \boldsymbol{\alpha}_i$ for $i \neq j$.

To provide 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 an integrated system 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 $l_{1,a_1} = 3$ and $\alpha_1 \in \{1, 2, 3\}$. The FEM solution on the coarsest mesh is plotted in Figure 2. Here the subscript 1 of $R_1$ and $l_{a_1}$ is used because the heat-transfer model is the first component of the system.

The 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.

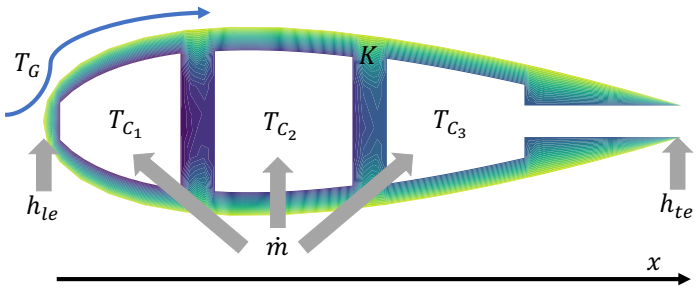


Figure 2: Finite element solution and parameterization of the turbine component model.

Table 1: Costs $W_{1,\alpha_1}$ of solving the heat transfer model for varying discretizations.

| $\alpha_1$ | 1 | 2 | 3 |
|---|---|---|---|
| Cost | 0.26388454 | 1.1500591 | 4.41993904 |
| DOF | 4998 | 17435 | 66549 |

## 3.2 Surrogate modeling

Typically, the fidelity of a numerical model is determined a priori, however this approach is inefficient. The fidelity of each component model should be determined based upon the goals of the modeling exercise. For example, when building a surrogate the numerical model error, which we call deterministic error, should be commensurate with the approximation error of the surrogate, which we call parametric error.

In this paper we construct surrogates of each component of an integrated system so as to reduce the cost of repeated interrogation of the system. With this goal, we denote the surrogate of the $k$-th component by

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

using a set $\mathcal{U}_{k,\boldsymbol{\beta}} = \{\boldsymbol{u}_k^{(m)}\}_{m=1}^M$ of $M$ samples of $\boldsymbol{u}_k := [\boldsymbol{z}_k^\top, \boldsymbol{\xi}_k^\top]^\top$ and evaluations $\{f_{k,\boldsymbol{\alpha}}(\boldsymbol{u}_k^{(m)})\}_{m=1}^M$ at those samples.[2] Here $\boldsymbol{\beta} \in \mathbb{R}^B$ is a multi-index that controls the computational cost of constructing a surrogate and

[2] $[\boldsymbol{\alpha}, \boldsymbol{\beta}]$ denotes the concatenation of the two multi-indices $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.

its parametric error. The number of entries $B$ is dictated by the approximation strategy used to construct each surrogate $f_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$. For example, $\boldsymbol{\beta}$ can be a scalar dictating the number of random training data used to build the surrogate or the level of refinement in each direction of $\boldsymbol{u}_k$.

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

$$\|f_k - f_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}\| \leq \|f_k - f_{k,\boldsymbol{\alpha}}\| + \|f_{k,\boldsymbol{\alpha}} - f_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}\|. \tag{6}$$

Here, the first term on the right-hand side represents the deterministic error and the second term represents the parametric error. In Figure 3 we use a simple example to depict these two sources of error. 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 the high-fidelity model $f_{k,[2]}$ (both dashed blue). As $\beta$ increases, we add two additional training data which allows the surrogate $f_{k,[\alpha,\beta]}$ to more accurately approximate $f_{k,\alpha}$, that is the parametric error decreases. 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.

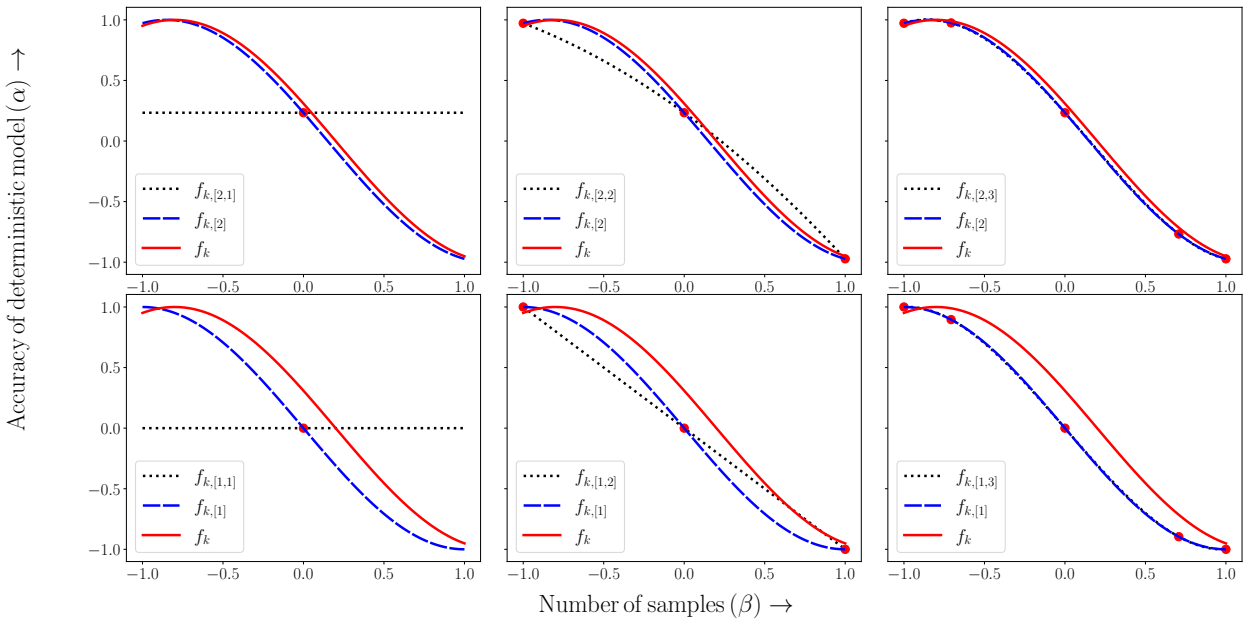


Figure 3: Approximations $f_{k,[\alpha,\beta]}$ (dotted black) of the one-dimensional function $f_k^{\epsilon}(\boldsymbol{u}) = \cos(\frac{1}{2}\pi(u_1 + \epsilon) + \frac{2}{5}\pi)$. The model fidelity is controlled by $\epsilon$. The low- $f_1$ and high-fidelity $f_2$ models (dashed blue) use $\epsilon = 0.2$ and $\epsilon = 0.05$ respectively. The true model (solid red) $f_k$ sets $\epsilon = 0$. Red dots depict samples used to build the interpolants. Cost of constructing $f_{k,[\alpha,\beta]}$ increases with $\alpha$ and $\beta$. MISC chooses the combination of $f_{k,[\alpha,\beta]}$ that minimize error for a given budget.

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 needed to balance these two sources of error.

In Section 4 we propose a cost-effective experimental design strategy that combines surrogates of each component, built using evaluations of multiple fidelities $f_{k,\boldsymbol{\alpha}}$. 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 relative to the cost of evaluating the model. Typically, less samples are assigned to higher model fidelities, that is $\boldsymbol{\alpha}$ with larger entries. To account for such approaches, moving forward we will denote a 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 random system 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. Following [27, 9], we construct an approximation of a component over an estimated range $\hat{\Xi}_k$ of the input variables with respect to a simpler probability measure $\nu$.

The following lemma characterizes the accuracy in a $\omega$-weighted norm given an approximation that is accurate in the $\nu$-weighted norm.

**Lemma 3.1** (Strong convergence [9]). *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 $f_\nu$ of $f$ with approximation error $\epsilon$, i.e.,*

$$\epsilon := \|f - f_\nu\|_{L^p_\nu(\Xi)}, \quad p \geq 1, \tag{7}$$

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

$$\|f - f_\nu\|_{L^p_\omega(\Xi)} \leq C_r^{1/p}\epsilon + C_f \delta^{1/p}, \quad \textit{provided} \quad C_r := \max_{\boldsymbol{u} \in \Xi \cup \hat{\Xi}} \frac{\omega(\boldsymbol{u})}{\nu(\boldsymbol{u})} < \infty. \tag{8}$$

This lemma suggests that treating the coupling variables as variables with an unknown distribution does not affect the rate at which the error converges in a component surrogate. The second term in (8) 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 $\hat{\Xi}_k$.

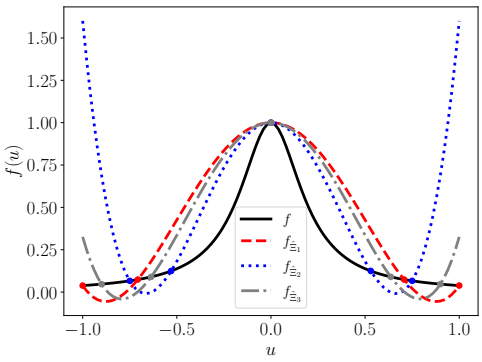


Figure 4: Lagrange polynomial interpolants of the Runge function $f(u) = (1 + 25u^2)^{-1}$ constructed using five function evaluations on different intervals $\hat{\Xi}_i$, $i = 1, 2, 3$.

In the following, we set $\nu$ 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 can lead to large approximation errors outside $\hat{\Xi}$.

In this paper, we use an adaptive algorithm, presented in Section 4.3, 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 Evaluating system-level QoI using component 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 use the functional

$$G_{\boldsymbol{z},\mathcal{J}} : (f_{1,\mathcal{I}_1}, \ldots, f_{K,\mathcal{I}_K}) \mapsto f(\boldsymbol{z})$$

to represent the evaluation of the integrated system at a sample of $\boldsymbol{z}$. Here $\mathcal{J} = \{\mathcal{I}_1, \ldots, \mathcal{I}_K\}$ are the index sets associated with each component surrogates $f_{k,\mathcal{I}_k}$ $k = 1, \ldots, K$. The accuracy of the system-level QoI, $\boldsymbol{q}_\mathcal{J} = \boldsymbol{R}^y G_{\boldsymbol{z},\mathcal{J}}[f_{1,\mathcal{I}_1}, \ldots, f_{K,\mathcal{I}_K}]$, obtained 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.

### 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 3.1** (Feed-forward surrogate error). *Assume that each component $f_k(\boldsymbol{z}, \boldsymbol{\xi})$ is Lipschitz continuous with respect to the coupling variables $\boldsymbol{\xi}_k$ with uniform Lipschitz constant $L_k$ for all $\boldsymbol{\xi} \in \Xi_k$. Furthermore, recall $y_{k,q}$ is the $q$-th QoI of the $k$-th component and let $y_{k,\mathcal{I}_k,q}$ denote the associated surrogate evaluation. If the trained surrogates satisfy $\|y_{k,q} - y_{k,\mathcal{I}_k,q}\|_{L^\infty(\Gamma)} \leq \epsilon_k \forall q = 1, \ldots, Q$, then we have for $f = f_1 \circ f_2 \circ \cdots \circ f_K$ and $\hat{f} = f_{1,\mathcal{I}_1} \circ f_{2,\mathcal{I}_2} \circ \cdots \circ f_{K,\mathcal{I}_K}$ that*

$$\sup_{\boldsymbol{z}\in\Gamma} \max_{q=1,\ldots,Q} |f_q(\boldsymbol{z}) - \hat{f}_q(\boldsymbol{z})| \leq \epsilon \frac{1 - L^K}{1 - L},$$

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

This result can be applied to any directed acyclic graph structure by applying the proposition to each branch of the graph and setting $\epsilon_k$ to be the largest of the errors in the surrogates for all upstream components providing inputs to the component under consideration. Lastly note that $\epsilon_k$ 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 (5) with itself, we can trivially use Proposition 3.1 to guarantee convergence and to arrive at the following result. Note that without loss of generality we can assume $\boldsymbol{\xi}_k = \boldsymbol{y}_j, \boldsymbol{\xi}_j = \boldsymbol{y}_k$ and state the result in terms of $\boldsymbol{\xi}$ only:

**Proposition 3.2** (Feedback surrogate error). *Under the assumptions of Proposition 3.1, the error in the coupling variables after $T$ fixed-point iterations satisfies*

$$\sup_{\boldsymbol{z}\in\Gamma} \max_{q=1,\ldots,Q} |\xi_q^T - \hat{\xi}_q^T| \leq \epsilon \frac{1 - L^T}{1 - L},$$

*where $\hat{\boldsymbol{\xi}} \in \mathbb{R}^Q$ is the output of the coupled surrogates, $L = \max_{t=1,\ldots,T} L_T$ and $\epsilon = \max_{k=1,\ldots,T} \epsilon_t$.*

# 4 Greedy experimental design for integrated systems

As shown in the previous section, the accuracy of the system-level QoI obtained by integrating component surrogates depends on the accuracy of each surrogate. This section presents a greedy algorithm, based on Adaptive MISC [28], that greedily allocates resources to components and their varying fidelities in a manner that is commensurate with their impact on the system-level QoI $\boldsymbol{q}$. We first discuss the use of MISC to construct a surrogate of a single component. We then extend MISC to formulate our adaptive resource allocation method and show how MISC can be used to accurately learn the appropriate ranges of the coupling variables.

## 4.1 Multi-index stochastic collocation

Multi-index stochastic collocation provides an effective mechanism to combine data from models $f_{\boldsymbol{\alpha}}$ of varying fidelity [28]. Here and in the remainder of this section, we drop the dependence on the component index $k$ for simplicity. MISC approximates a component model as a linear combination of surrogates $f_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$ built using evaluations of the component using different model fidelities $f_{k,\boldsymbol{\alpha}}$

$$f(\boldsymbol{u}) \approx f_{\mathcal{I}}(\boldsymbol{u}) = \sum_{[\boldsymbol{\alpha},\boldsymbol{\beta}]\in\mathcal{I}} c_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}\, f_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}(\boldsymbol{u}). \tag{9}$$

where $N$ denotes the number of variables in $\boldsymbol{u} := [\boldsymbol{z}^\top, \boldsymbol{\xi}^\top]^\top$.

MISC utilizes multiple $f_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}$ to balances physical and parametric errors and to reduce the computational cost of achieving a specified level of accuracy. The accuracy of the MISC approximation (9) is dictated by the set $\mathcal{I} \subset \mathbb{N}^{R+N}$. Provided the index set is *downward-closed*, that is

$$\text{If } [\boldsymbol{\gamma},\boldsymbol{\delta}] \leq [\boldsymbol{\alpha},\boldsymbol{\beta}] \text{ and } [\boldsymbol{\alpha},\boldsymbol{\beta}] \in \mathcal{I} \Rightarrow [\boldsymbol{\gamma},\boldsymbol{\delta}] \in \mathcal{I}, \tag{10}$$

where $\boldsymbol{\gamma} \leq \boldsymbol{\alpha}$ if $\gamma_i \leq \alpha_i$, $\forall i$, then the coefficients $c_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}$ of (9) are given by the so-called *combination technique formula*[3]

$$c_{[\boldsymbol{\alpha},\boldsymbol{\beta}]} = \sum_{\substack{[\boldsymbol{i},\boldsymbol{j}]\in\{0,1\}^{R+N} \\ [\boldsymbol{\alpha}+\boldsymbol{i},\boldsymbol{\beta}+\boldsymbol{j}]\in\mathcal{I}_k}} (-1)^{\|[\boldsymbol{i},\boldsymbol{j}]\|_1}, \tag{11}$$

We represent $f_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}(\boldsymbol{u})$ using tensor-product Lagrangian interpolants constructed using evaluations of the $\boldsymbol{\alpha}$-fidelity model $f_{\boldsymbol{\alpha}}(\boldsymbol{u})$ associated with a Cartesian grid of univariate points. The resolution of each univariate set is prescribed by the multi-index $\boldsymbol{\beta}$. Letting $m : \mathbb{N} \to \mathbb{N}$ be an increasing function and for each univariate variable $u_n, n = 1, \ldots, N$ of the variable $\mathbf{u}$, we construct a set of $m(\beta_n)$ interpolation points

$$\mathcal{U}_{n,\beta_n} = \{u_{n,\beta_n}^{(j)}\}_{j=1}^{m(\beta_n)}, \tag{12}$$

and build $m(\beta_n)$ Lagrange polynomials

$$\mathcal{L}_{n,\beta_n}^{(j)}(u_n) = \prod_{l=1,l\neq j}^{m(\beta_n)} \frac{u_n - u_{n,\beta_n}^{(l)}}{u_{n,\beta_n}^{(j)} - u_{n,\beta_n}^{(l)}}, \quad j = 1, \ldots, m(\beta_n).$$

In the following we use univariate weighted Leja sequences [36, 16] tailored to the probability distribution function of $u_n$ for $\mathcal{U}_{n,\beta_n}$. Leja sequences are nested, that is, $\mathcal{U}_{n,\beta_n} \subset \mathcal{U}_{n,\beta_n^\star}$ if $\beta_n^\star > \beta_n$. We define $m(\beta) = 2\beta+1$ in Equation (12) and set the maximum level of the univariate sequence to $\beta = 15$. For more details on the construction of Leja sequences, refer to Section 4.3.

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

$$\mathcal{U}_{\boldsymbol{\beta}} = \times_{n=1}^{N} \mathcal{U}_{n,\beta_n} := \{\boldsymbol{u}_{\boldsymbol{\beta}}^{(\boldsymbol{j})}\}_{\boldsymbol{j}\leq m(\boldsymbol{\beta})}$$

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

$$\mathcal{L}_{\boldsymbol{\beta}}^{(\boldsymbol{j})}(\boldsymbol{u}) = \prod_{n=1}^{N} \mathcal{L}_{n,\beta_n}^{(j_n)}(u_n), \quad \boldsymbol{j} \leq m(\boldsymbol{\beta}),$$

and finally define the tensor-product interpolant as

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

Again, consider the approximations depicted in Figure 3. As stated previously, ideally we would use the approximation $f_{k,[2,3]}$ in the top right panel, but this can be infeasible due to the cost of evaluating high-fidelity models. Instead MISC provides a mechanism for using a combination of $f_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$, defined via the index set $\mathcal{I}_k$, to produce an accurate approximation (9) for a given budget. The accuracy and cost of the resulting approximation is dictated by this set $\mathcal{I}_k$. An adaptive algorithm for determining $\mathcal{I}_k$ for black-box models was presented in [28]. In the following section we extend this algorithm to not only balance parametric and deterministic errors of a single component surrogate but also balance the errors of each component surrogate on predictions of system-level QoI.

[3] $\{0,1\}^s$ is the set of s-dimensional vectors containing all combinations of zero and one and $\|\boldsymbol{\delta}\|_1 = \sum_i |\delta_i|$

## 4.2 Adaptively allocating resources to components

Proposition 3.1 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 presents a greedy algorithm that adaptively allocates resources to each component in a manner that is commensurate with their impact on the system-level QoI $\boldsymbol{q}$. The procedure is summarized in Algorithm 1. The algorithm is an extension of the adaptive MISC algorithm used to construct multi-fidelity approximations of black-box models. Consequently, the method described here balances the parametric and deterministic errors of each component inaddition to balancing errors from each component surrogate.

**Algorithm 1** CONSTRUCT_COMPONENT_SURROGATES$[\{f_{k,\boldsymbol{\alpha}}\}_{k=1}^K, \tau, W_{\max}] \rightarrow f_{\mathcal{J}}$

**for** $k = 1, \ldots, K$ **do**
  $\mathcal{I}_k, \mathcal{R}_k, \mathcal{E}_{\mathcal{R}_k} :=$ INITIALIZE_SURROGATE$(f_{k,\boldsymbol{\alpha}})$ ▷ Initialize $k$-th component surrogates
**end for**
**while not** TERMINATE$[\{\mathcal{I}_k\}_{k=1}^K, \tau, W_{\max}]$ **do**
  $\{l, [\boldsymbol{\alpha}^\star, \boldsymbol{\beta}^\star]\} := \text{argmax}_{k\in[1,K]} \mathcal{E}_k$ ▷ Get highest priority $\gamma_{[\boldsymbol{\alpha},\boldsymbol{\beta}]} \in \mathcal{E}_k$ index over all $K$ components
  $\boldsymbol{y}_{k,[\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star]} := f_{k,\boldsymbol{\alpha}}(\mathcal{U}^{\Delta}_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]})$ ▷ Evaluate new training data
  $\mathcal{R}_l, \mathcal{I}_l :=$ REFINE$[[\boldsymbol{\alpha}^\star, \boldsymbol{\beta}^\star], \mathcal{R}_l, \mathcal{I}_l]$ ▷ Update surrogate $\mathcal{I}_l$ and candidate $\mathcal{R}_l$ index sets
  **for** $k = 1, \ldots, K$ **do** ▷ Recompute indicators of candidate indices of all $K$ components
    **for** $[\boldsymbol{\alpha}, \boldsymbol{\beta}] \in \mathcal{R}_k$ **do**
      $\gamma^k_{[\boldsymbol{\alpha},\boldsymbol{\beta}]} :=$ ERROR_INDICATOR$[\mathcal{I}_k, [\boldsymbol{\alpha}, \boldsymbol{\beta}]]$
    **end for**
  **end for**
**end while**

The proposed algorithm begins by initializing a set of surrogates for each component in the system $f_{k,\mathcal{I}_k}$ $k = 1, \ldots, K$. This initialization returns the initial index set $\mathcal{I}_k$, a set $\mathcal{R}_k$ of possible indices $[\boldsymbol{\alpha}, \boldsymbol{\beta}]$ to add to the surrogate, and the error indicators $\gamma_{[\boldsymbol{\alpha},\boldsymbol{\beta}]} \in \mathcal{E}_k$ associated with each of these candidates.[4] The initialization routine is summarized in Algorithm 2.

**Algorithm 2** INITIALIZE_SURROGATE$[f_{k,\boldsymbol{\alpha}}] \rightarrow \mathcal{I}_k, \mathcal{R}_k, \mathcal{E}_k$

**for** $k = 1, \ldots, K$ **do**
  $[\boldsymbol{\alpha}^\star, \boldsymbol{\beta}^\star] := [1, 1, \cdots] \in \mathbb{R}^{R_k + D_k + S_k}$ ▷ Initialize each component surrogate
  $\mathcal{I}_k = \{[\boldsymbol{\alpha}^\star, \boldsymbol{\beta}^\star]\}$
  $\boldsymbol{y}_{k,[\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star]} := f_{k,\boldsymbol{\alpha}}(\mathcal{U}^{\Delta}_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]})$ ▷ Evaluate new training data
  $\mathcal{R}_k, \mathcal{I}_k :=$ REFINE$[[\boldsymbol{\alpha}^\star, \boldsymbol{\beta}^\star], \emptyset, \mathcal{I}_k]$ ▷ Initialize surrogate $\mathcal{I}_l$ and candidate $\mathcal{R}_l$ index sets
  **for** $k = 1, \ldots, K$ **do** ▷ Compute indicators of candidate indices of all $K$ components
    **for** $[\boldsymbol{\alpha}, \boldsymbol{\beta}] \in \mathcal{R}_k$ **do**
      $\gamma^k_{[\boldsymbol{\alpha},\boldsymbol{\beta}]} :=$ ERROR_INDICATOR$[\mathcal{I}_k, [\boldsymbol{\alpha}, \boldsymbol{\beta}]]$
    **end for**
  **end for**
**end for**

Once initialized, Algorithm 1 chooses the best individual component surrogate to refine, which we denote with the index $l$, which is predicted to lead to the greatest reduction in error. These choices are made using the error indicators $\gamma^k_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}$, contained in $\mathcal{E}_k$ for $k = 1, \ldots, K$, which predict the reduction in error in system-level QoI obtained from refinement of each surrogate. The calculation of the error indicators is summarized in Section 4.2.1.

When a surrogate is selected for refinement, Algorithm 1 generates training samples $\mathcal{U}_{k,[\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star]}$ and associated evaluations of $f_{l,\boldsymbol{\alpha}^\star}$; the exact number of new samples is dictated by $\boldsymbol{\beta}^\star$. This data are then used to improve the accuracy of the chosen surrogate. When using nested Leja sequences many points in $\mathcal{U}_{k,[\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star]}$,

[4] 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.

needed to construct $f_{k,[\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star]}$, may have already been evaluated and can simply be reused. Consequently the function $f_{k,\boldsymbol{\alpha}^\star}$ is evaluated on the set of new points $\mathcal{U}^{\Delta}_{k,[\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star]}$ needed to construct $f_{k,[\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star]}$.

Once additional training data has been generated, the `REFINE` procedure summarized in Algorithm 3 then removes $[\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star]$ from the set of possible candidates $\mathcal{R}_l = \mathcal{R}_l \setminus [\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star]$ and adds it to the surrogate index set $\mathcal{I}_l = \mathcal{I}_l \cup [\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star]$. Step 5 generates new candidates and steps 6 and 7 add all candidates that satisfy the downward closed condition in (10) to $\mathcal{R}_l$. See [28] for a visual depiction of the refinement step.

**Algorithm 3** `REFINE`$[[\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star],\mathcal{R}_k,\mathcal{I}_k]\rightarrow \mathcal{R}_k,\mathcal{I}_k$

1: $N_k = R_k + D_k + S_k$
2: $\mathcal{R}_k = \mathcal{R}_k \setminus [\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star]$
3: $\mathcal{I}_k = \mathcal{I}_k \cup [\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star]$
4: **for** $i = 1,\dots,N_k$ **do**
5: $\quad\boldsymbol{\delta} := [\boldsymbol{\alpha}^\star,\boldsymbol{\beta}^\star] + e_{N_k,i}$
6: $\quad$**if** $\boldsymbol{\delta} - e_{N_k,j} \in \mathcal{I}_k \;\forall j = 1,\dots,N_k$ **then** ▷ Check downwards closed condition (10)
7: $\quad\quad\mathcal{R}_l = \mathcal{R}_l \cup \delta$
8: $\quad$**end if**
9: **end for**

The coupled nature of the integrated system causes changes in one surrogate to impact some or all other components. Consequently, the final step of the adaptive algorithm involves updating 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.

The aforementioned steps are repeated until a computational budget $W_{\max}$ is exceeded or a pre-specified accuracy tolerance $\tau$ is met. The final output of the algorithm 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 accurately used to predict the system-level QoI. 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 the algorithm uses a-posteriori estimates of error, i.e., it evaluates the priority 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. Note that when $k = 1$ this algorithm is simply the black-box algorithm presented in [28].

### 4.2.1 Error indicators

The function for computing the error indicators employed in Algorithm 1 must be tailored to promote accuracy in system-level QoI. With this goal we make the, often reasonable, assumption that the cost of a simulation for a given fidelity $\boldsymbol{\alpha}$ is fixed, i.e. does not change with $\boldsymbol{u}$; hereafter, we denote this quantity $W_\alpha$. Now, given an approximation $f_{k,\mathcal{I}_k}$, let $[\boldsymbol{\alpha},\boldsymbol{\beta}]$ be a candidate index from of $\mathcal{R}_k$ and $f_{\mathcal{J}^k_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}}$ denote the approximation obtained by adding the new approximation $f_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$ to (9), where

$$\mathcal{J}^k_{[\boldsymbol{\alpha},\boldsymbol{\beta}]} = \mathcal{I}_1 \cup \mathcal{I}_2 \cup \cdots \cup \Big\{\mathcal{I}_k \cup \{[\boldsymbol{\alpha},\boldsymbol{\beta}]\}\Big\} \cup \mathcal{I}_{k+1} \cup \cdots$$

To construct our error indicator, we generate a set $\mathcal{Z}^{\text{refine}} = \{\boldsymbol{z}^{(l)}\}_{l=1}^L$ of $L$ realizations of the input random variables $\boldsymbol{z}$. We then evaluate the system-level QoI, using $f_{\mathcal{J}^k_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}}$ and $f_{\mathcal{J}^k}$, at each sample to obtain the corresponding surrogate estimates of the QoI, $\boldsymbol{q}^{(l)} = \boldsymbol{R}^y G_{z^{(l)},\mathcal{J}^k_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}}$ and $\boldsymbol{q}^{(l)} = \boldsymbol{R}^y G_{z^{(l)},\mathcal{J}^k}$ respectively. Finally denote the $i$-th entry of QoI vector as $q_i$ and let

$$\Delta E^k_{[\boldsymbol{\alpha},\boldsymbol{\beta}]} = \max_{i=1,\dots,Q^{\text{sys}}} \left(\sum_{l=1}^{L} (q_i^{(l)} - q_i^{(l)})^2\right)^{\frac{1}{2}} \qquad \Delta W^k_{[\boldsymbol{\alpha},\boldsymbol{\beta}]} = |\text{Work}[f_{\mathcal{J}^k_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}}] - \text{Work}[f_{\mathcal{J}}]| \tag{13}$$

respectively denote the worst case (over QoI) root mean squared error between two successive approximations and the work needed to update the approximation. Then we use

$$\gamma^k_{\boldsymbol{\alpha},\boldsymbol{\beta}} = \frac{\Delta E^k_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}}{\Delta W^k_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}}\,. \tag{14}$$

as the error indicator in Algorithm 1. The quantity $\Delta W^k_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}$ has the closed form expression $\Delta W^k_{[\boldsymbol{\alpha},\boldsymbol{\beta}]} = W_{\boldsymbol{\alpha}}\text{card}(\mathcal{U}^{\Delta}_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]})$. The quantity $\Delta E^k_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}$ cannot be computed a priori and instead must be evaluated *a-posteriori*, i.e. after having actually added $[\boldsymbol{\alpha},\boldsymbol{\beta}]$ to $\mathcal{I}_k$ and evaluating the component at the samples $\mathcal{U}^{\Delta}_{k,[\boldsymbol{\alpha},\boldsymbol{\beta}]}$.

The refinement indicator (14) refines component surrogates based upon their contribution to the error in the approximation of the system-level QoI. This requires evaluating and integrating the output of each component via the functional $\mathcal{G}_z$, that is using the feed-forward and fixed-point iterations methods outlined in Section 2.2. Unfortunately, using (14) will give misleading results when the surrogate of an upstream component is a constant function. When this occurs, the indicator will predict no system improvement when improving the downstream component surrogate, i.e. $\Delta E^{k,\text{sys}}_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}$. To avoid this situation, we initialize the priorities in Algorithm 1 using the following indicator, which measures the local component error:

$$\gamma^k_{[\boldsymbol{\alpha},\boldsymbol{\beta}]} = \frac{1}{\Delta W_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}}\left(\omega\Delta E^{k,\mu}_{[\boldsymbol{\alpha},\boldsymbol{\beta}]} + (1-\omega)\Delta E^{k,\sigma}_{[\boldsymbol{\alpha},\boldsymbol{\beta}]}\right), \quad \omega \in [0,1] \tag{15}$$

$$\Delta E^{k,\mu}_{[\boldsymbol{\alpha},\boldsymbol{\beta}]} = \frac{1}{|f_{k,\mathbf{0},\mathbf{0}}|}\left|\mathbb{E}\left[f_{k,\mathcal{I}_k\cup\{[\boldsymbol{\alpha},\boldsymbol{\beta}]\}}\right] - \mathbb{E}\left[f_{k,\mathcal{I}_k}\right]\right|,$$

$$\Delta E^{k,\sigma}_{[\boldsymbol{\alpha},\boldsymbol{\beta}]} = \frac{1}{|f_{k,\mathbf{0},\mathbf{0}}|}\sqrt{\left|\mathbb{V}\left[f_{k,\mathcal{I}_k\cup\{[\boldsymbol{\alpha},\boldsymbol{\beta}]\}}\right] - \mathbb{V}\left[f_{k,\mathcal{I}_k}\right]\right|}.$$

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 $|f_{k,\mathbf{0},\mathbf{0}}|$ is the value of the coarsest fidelity at the mid of the parametric domain $\Gamma_k \times \Xi_k$ 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 $\omega = 0.5$.

## 4.3 Estimating the range of the coupling variables

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.

A univariate weighted Leja sequence is defined as a sequence of samples $z^{(l)} \in I$ for $l \in 1, 2, \ldots$ such that

$$u^{(L+1)} = \underset{u\in I}{\text{argmax}}\, v(u)\prod_{l=1}^{L}|u-u^{(l)}| \tag{16}$$

for some weight function $v(u)$. In this paper, we follow [36] and set $v(u) = \sqrt{\rho(u)}$ where $\rho(u)$ is the PDF of the variable $u$. For uniform variables used to represent coupling variables, the PDF is a constant and does not affect the Leja sequence.

By construction, Leja sequences are nested. That is, $\mathcal{U}_L \subset \mathcal{U}_{L+1}$ where $\mathcal{U}_L = \{u^{(1)}, \ldots, u^{(L)}\}$. Moreover, the initial point $u^{(1)}$ may be arbitrarily chosen. Indeed we can extend any set of initial points $\mathcal{U}_L = \{u^{(1)}, \ldots, u^{(L)}\}$. Thus, given a Leja sequence $\mathcal{U}^{I^t}_L$ constructed on a range $I^t$, we can generate the next point of Leja sequence $\mathcal{U}^{I^{t+1}}_L$ over a larger range $I^{t+1}$, satisfying $I^t \subset I^{t+1}$ by simply searching for the next point in the interval $I^{t+1}$ instead of $I^t$. 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 modifying Algorithm 1 slightly. The modification entails estimating the ranges of $\boldsymbol{\xi}$ using the values of $\boldsymbol{y}$ obtained from evaluating the various $f_{\mathcal{J}^k_{\boldsymbol{\alpha},\boldsymbol{\beta}},q}(\mathcal{U}^{\text{refine}}), k = 1, \ldots K$ needed when using Equation (14). More 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 $\Xi^t_{k,i} = [a^t_{k,i}, b^t_{k,i}]$ to denote the current range of $\xi_{k,i}$, the new range $\Xi^{t+1}_{k,i} = [a^{t+1}_{k,i}, b^{t+1}_{k,i}]$ is updated as follows:

$$a^{t+1}_{k,i} = \min\left(a^t_{k,i}, \min_{\boldsymbol{z}\in\mathcal{U}^{\text{refine}}} f_{m,\mathcal{I}^{t+1}_m,q}(\boldsymbol{z})\right) \qquad b^{t+1}_{k,i} = \max\left(b^t_{k,i}, \max_{\boldsymbol{z}\in\mathcal{U}^{\text{refine}}} f_{m,\mathcal{I}^{t+1}_m,q}(\boldsymbol{z})\right)$$

where $\mathcal{I}_m^{t+1}$ denotes the new multi-index set being considered at the current iteration for building the surrogate model of the $m$-th component, i.e., $\mathcal{I}_m^{t+1} = \mathcal{I}_m^t \cup \{[\boldsymbol{\alpha}, \boldsymbol{\beta}]\}$ for some multi-index $[\boldsymbol{\alpha}, \boldsymbol{\beta}]$.

In the numerical examples to follow, we evaluate the entire system at 10 random realizations of the input $\boldsymbol{z}$, and set the initial ranges $\Xi_{k,i}^0$ of each coupling variable to be the minimum and maximum values over the 10 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 a priori-fixed bounds.

# 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 measure error in the predictions made from combining those surrogates. Error is measured by drawing 1000 random samples from the PDF of the random variables $\boldsymbol{z}$ and computing the relative $\ell^2$ (root mean squared) error between the exact system output and the surrogate approximation, normalized by the $\ell^2$ norm of the validation data. All the numerical results were produced using the PyApprox software package [26].

## 5.1 Algebraic single-fidelity feed-forward system

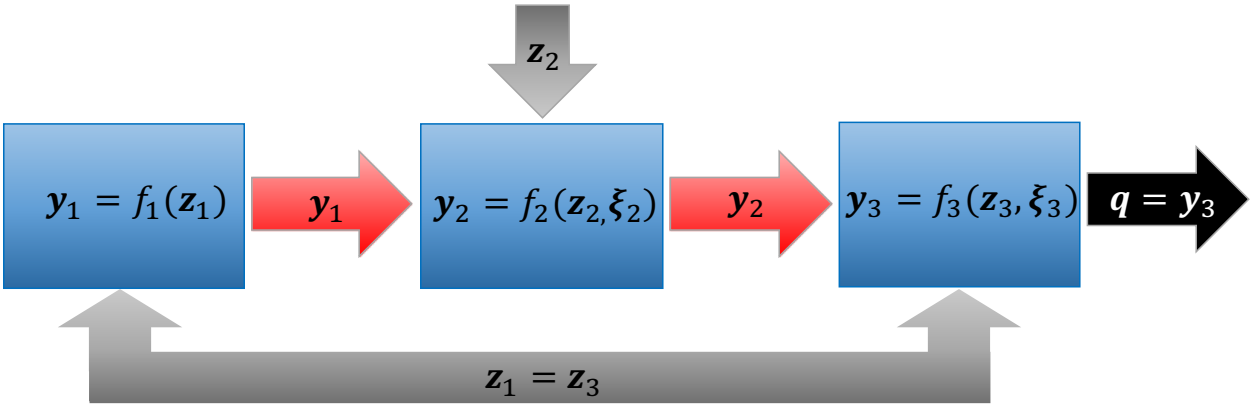


Figure 5: A feed-forward system consisting of three components. Coupling variables are depicted in red, external inputs in gray and system-level QoI in black. The first and third components share the same random variables, i.e. $\boldsymbol{z}_1 = \boldsymbol{z}_3$, and the inputs to the second component are unique to that component i.e. $\boldsymbol{z}_2 \cap \boldsymbol{z}_k = \emptyset$, $k = 1, 3$.

Consider the coupled system depicted in Figure 5. The system consists of three components in a chain with vector-valued feed-forward coupling, where $f(\boldsymbol{z}_1) = (y_{1,1}, \ldots, y_{1,Q_1})^\top$, $f(\boldsymbol{z}_2, \boldsymbol{\xi}_2) = (y_{2,1}, \ldots, y_{2,Q_2})^\top$ and $f(\boldsymbol{z}_3, \boldsymbol{\xi}_3) = [y_{3,1}]$, and the expressions of the outputs $y_{i,j}$ are given by

$$
\begin{aligned}
y_{1,\alpha_1,q} &= f_{1,\alpha_1,q}(\boldsymbol{z}_1) = z_1^q \sin\left(\sum_{d=1}^{D_1} z_{1,d} + \epsilon_{\alpha_1}\right), \quad q = 1, \ldots, Q_1 \\
y_{2,\alpha_2,q} &= f_{2,\alpha_2,q}(\boldsymbol{z}_2, \boldsymbol{\xi}_2) = \left(\prod_{s=1}^{S_2} (\xi_s^{q+1} - \epsilon_{\alpha_2})\right)\left(\prod_{d=1}^{D_2} z_{2,d}\right), \quad q = 1, \ldots, Q_2 \\
y_{3,\alpha_3,q} &= f_{3,\alpha_3}(\boldsymbol{z}_3, \boldsymbol{\xi}_3) = \exp\left[-\sum_{s=1}^{S_3} (\xi_s - \epsilon_{\alpha_3})^2\right] \frac{1}{1 + \frac{25}{16}\left(\sum_{d=1}^{D_3} z_{3,d}\right)^2},
\end{aligned}
\tag{17}
$$

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 $\epsilon_{\alpha_k}$ and thus the accuracy of $f_{k,\alpha_k}$.

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 $\boldsymbol{z}_1 = \boldsymbol{z}_3$. 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, whereas Case 3 utilizes models of varying accuracy cost for each component. We assume that cost increases with increasing $\alpha$, but we assume, for a given value of $\alpha$, evaluating each component has the same computational cost. This means evaluating the entire system costs three times as much as evaluating a single component. In Sections 5.1.1 and 5.1.2, we will also assume that we only have access to a single-fidelity model for each component; that is, we do not have access to a set of approximations $f_{1,\alpha}, f_{2,\alpha}, f_{3,\alpha}$. Specifically, we set $\alpha_k \to \infty$ such that $\epsilon_k = 0$, $k = 1, \ldots K$. In Section 5.1.3, we demonstrate the utility of including models of varying fidelity for each component.

### 5.1.1 Range estimation

The true range of the coupling variables can be determined from inspection of the component functions. For Case 1, both coupling variables can assume values in the interval $[0, 1]$. The true PDF of the coupling variables are shown in Figure 6 (right). In Figure 6 (left), we plot the error in the predictions of $f_3$ for Case 1 using different ranges for the coupling variables. The intervals in the legend denote the a priori fixed range of the coupling variables. The legend element "Estimated" refers to the approximation obtained when using the adaptive range estimation procedure outlined in Section 4.3. We also compare against the case of a single surrogate that treats the system as a black-box.

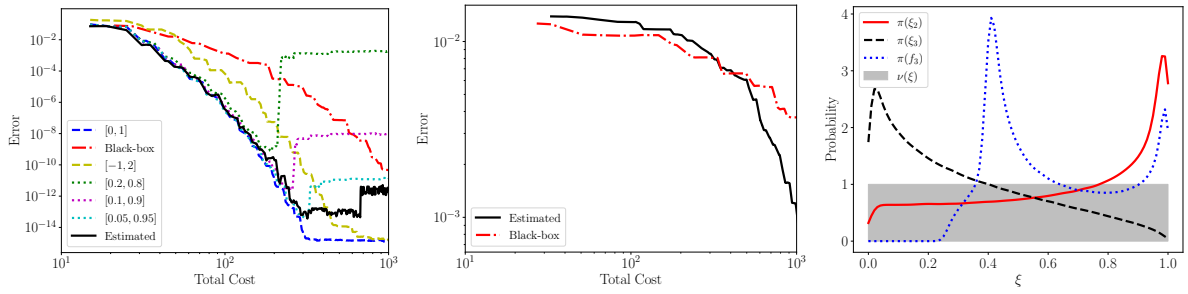


Figure 6: Error vs cost for the feed-forward system (17) for (left) Case 1 with ($Q_1 = Q_2 = 1$, $D_1 = D_2 = 1$) and (middle) Case 2 with ($Q_1 = Q_2 = 4$, $D_1 = D_2 = 2$). True PDFs (right) of the coupling variables and output of the integrated system for Case 1. The dominating measure ($\nu$ from Lemma 3.1), used to compute the Leja sequences for the coupling variables on $[0, 1]$, is depicted in gray.

All component surrogate cases 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 $\delta$ in Lemma 3.1, where $\delta$ 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 in that the curve shifts right relative to the black curve based on the true ranges.

The dynamic estimation of the coupling ranges can be seen to be effective. It identifies the coupling ranges of $\xi_1$ and $\xi_2$ to be $[-0.02, 1.00]$ and $[-0.23, 0.92]$, respectively. The error saturates because we only use $L = 100$ samples to estimate the ranges of the coupling variables and thus under-estimate the upper bound of $\xi_2$. Although not shown, the saturation of error can be removed by increasing the number of samples $L$. Again, we emphasize that this does not increase the number of evaluation of the true component functions. We choose such a small value of $L$ 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 6, 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 6 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 Dimension reduction and non-linearity

The middle plot of Figure 6 compares the accuracy of integrated component 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 component 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 $D_k < D$. 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[5] interpolation of a function, which requires $O\left(\epsilon^{-D}\right)$ black-box evaluations of the entire system to achieve an error $\epsilon$. The cost of using this method to build component surrogates over both the coupling and random variables satisfies

$$O\left(\sum_{k=1}^{K} \epsilon^{-(D_k+S_k)}\right) < O\left(\epsilon^{-D}\right) \qquad \text{if} \qquad \max_{k=1,\ldots,K} D_k + S_k < D + \frac{\log(K)}{\log(\epsilon)} \tag{18}$$

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, coupled systems of 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. Consider a composition of K linear functions. The system-level QoI will have degree K and so will be much more difficult to evaluate than any component. Note, 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.

We conclude this single-fidelity example by remarking that it was 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]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.

### 5.1.3 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 (17) with $Q_1 = Q_2 = 2$ and $D_1 = D_2 = 2$. By varying $\alpha_k$, 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 $f_{k,\alpha_k}$ 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 7 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 by using a multi-fidelity approach.

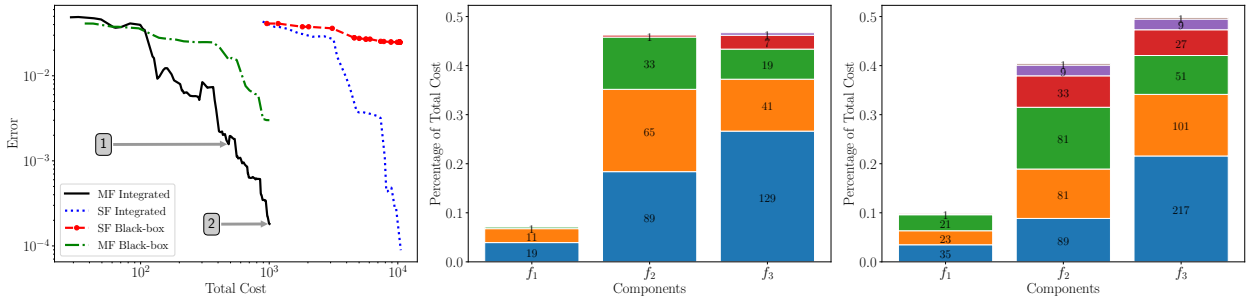


Figure 7: (Left) Error vs cost for the feed-forward multi-fidelity system (17). Evaluations allocated by the MF Integrated sampling procedure to the varying fidelity models of each component when the total cost is approximately 484 (middle) and 1005 (right). The two numbered boxes in the left plot indicate the points on the convergence curve that are associated respectively with the middle and right plots.

The middle and right plots of Figure 7 depict the percentage of the computational work allocated to the various model discretizations of each component when the total work is $\approx 484$ and $\approx 1005$, 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 use of these models alone cannot further reduce the error, and so evaluations of the higher-fidelity models are required. For example, the middle plot shows only four fidelities have been used for component 2. In the right plot, six fidelities have been evaluated. Higher-fidelity (larger $\alpha_k$) evaluations are needed when the interpolation error $\|f_{k,\alpha} - f_{k,\mathcal{I}_k}\|$ of a component surrogate becomes commensurate with the model error $\|f_k - f_{k,\alpha}\|$ induced by using the model approximation $f_{k,\alpha}$.

## 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 [52]. The system has eight random variables (see Table 2) and seven coupling variables (see Table 3). Model constants

are reported in [6]. In the following, we use integrated surrogates to accurately approximate three outputs of the system: the total torque $\tau_{\text{tot}}$ ($y_{3,2}$) coming from the attitude control component, the total power output $P_{\text{tot}}$ ($y_{2,3}$), and the area of the solar array $A_{\text{sa}}$ ($y_{2,4}$) 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 | $H$ | $\boldsymbol{z}_1, \boldsymbol{z}_3$ | $\mathcal{N}\left(18 \times 10^6, 1 \times 10^6\right)$ |
| 2 | Target diameter | $\phi$ | $\boldsymbol{z}_1$ | $\mathcal{N}\left(235, 10\right)$ |
| 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 | $C_d$ | $\boldsymbol{z}_3$ | $\mathcal{N}\left(1, 0.2\right))$ |

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 | $y_{3,2}$ | — |

The left graphic of Figure 8 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 three fixed-point iterations to estimate the coupling variables. Assuming the cost of each component is one unit, the evaluation of the black-box requires three evaluations of each component, that is $3 \times 3 = 9$ 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 8 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 allocates more computational resources to approximating the second and third component. Note, 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. The adaptive sampling algorithm allocates many more evaluations to the attitude and power components relative to the orbit component.

Note that the saturation of the errors, present for both approaches depicted in the left plot of Figure 8, 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.

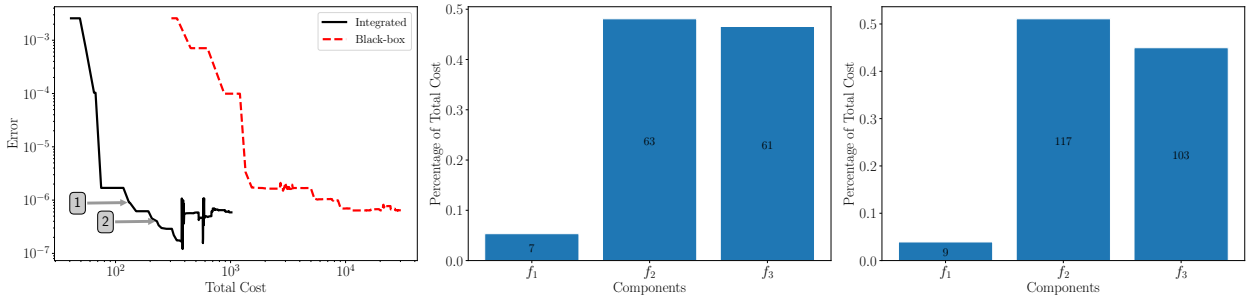


Figure 8: (Left) Error vs cost for the QoI $P_{tot}$ obtained from the fire detection satellite system model. (Middle) The percentage of the computational work allocated to evaluating each component when building the integrated surrogates when total cost is 131 and (Right) 229. The two numbered boxes 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

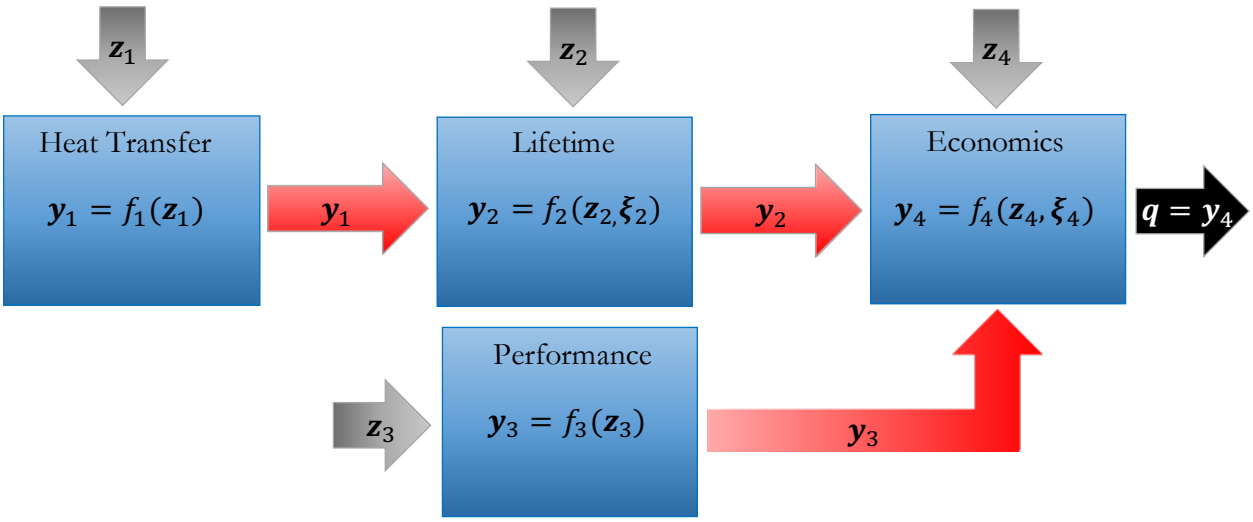


Figure 9: A multi-fidelity economics model of a turbine consisting of four components. Coupling variables are depicted in red, external inputs in gray and system-level QoI in black. The random variables are $\boldsymbol{z}_1 = [T_{c_1}, T_{c_2}, T_{c_3}, K, h_{le}, h_{te}]^\top, \boldsymbol{z}_2 = [P_{lm}]^\top, \boldsymbol{z}_3 = [\dot{m}, T_g, F_{perf}]^\top$ 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 $\boldsymbol{z}_j \cap \boldsymbol{z}_k = \emptyset \ \forall j, k$.

In this section we investigate the performance of our methodology on a coupled financial model for a gas turbine as depicted in Figure 9 [1, 31]. 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.

**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

$$\begin{aligned}\nabla(k\cdot\nabla h(x)) &= 0 \quad x\in\Omega\\ h(x) &= T_{c_j} \quad x\in\partial\Omega_j,\ j=1,2,3\\ h(x) &= h_{te}+(h_{le}-h_{te})\exp\left(-4\frac{x_1^2}{4\times10^{-6}}\right) \quad x\in\partial\Omega_4\end{aligned}$$

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 [42] 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.

The left of Figure 10 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

Table 4: Random variables of the economics-turbine system depicted in Figure 9. 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 | $F_{econ}$ | $\boldsymbol{z}_4$ | $\mathcal{U}[0.9, 1.1]$ |

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, these investments can be limited to only those necessary relative to other concerns. Moreover, the required precision can be automatically determined.

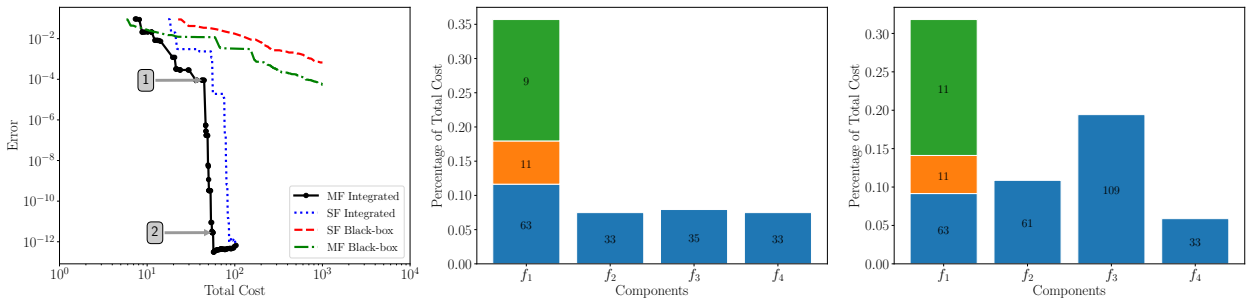


Figure 10: (Left) Error vs cost for the economics-turbine system model. (Middle) Allocation of work for the MF Integrated surrogates when total cost is approximately 44 and (Right) 51. The two numbered boxes in the left plot indicate the points on the convergence curve that are associated respectively 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. Extensive numerical examples demonstrated that building approxima-

tions 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 Proposition

The following outlines the proof of Proposition 3.1.

**Proof**

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

$$\begin{aligned}\|f_k(\boldsymbol{z}_k,\boldsymbol{\xi}_k) - f_{k,\mathcal{I}_k}(\boldsymbol{z}_k,\boldsymbol{\xi}_k^\star)\| &\le \|f_k(\boldsymbol{z}_k,\boldsymbol{\xi}_k) - f_k(\boldsymbol{z}_k,\boldsymbol{\xi}_k^\star)\| + \|f_k(\boldsymbol{z}_k,\boldsymbol{\xi}_k^\star) - f_{k,\mathcal{I}_k}(\boldsymbol{z}_k,\boldsymbol{\xi}_k^\star)\| \\ &\le L_k\|\boldsymbol{\xi}_k - \boldsymbol{\xi}_k^\star\| + \epsilon \end{aligned} \tag{19}$$

where in the last step we have used the assumption of Lipschitz continuity to bound the first term and the definition of $\epsilon$ to bound the second one. Now without loss of generality set $\boldsymbol{z}_k = \boldsymbol{z}$ and $\boldsymbol{\xi}_k = \boldsymbol{\xi}$, $\forall k$, then by

repeated use of (19)

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

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. □

## References


[1] Sergio Amaral, Douglas Allaire, and Karen Willcox. A decomposition-based approach to uncertainty analysis of feed-forward multicomponent systems. *International Journal for Numerical Methods in Engineering*, 100(13):982–1005, 2014.

[2] M. Arnst, R. Ghanem, E. Phipps, and J. Red-Horse. Dimension reduction in stochastic modeling of coupled problems. *International Journal for Numerical Methods in Engineering*, 92(11):940–968, 2012.

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

[4] J. Beck, L. Tamellini, and R. Tempone. IGA-based Multi-Index Stochastic Collocation for random PDEs on arbitrary domains. *Computer Methods in Applied Mechanics and Engineering*, 351:330 – 350, 2019.

[5] Kevin Carlberg, Sofia Guzzetti, Mohammad Khalil, and Khachik Sargsyan. The network uncertainty quantification method for propagating uncertainties in component-based systems. *arXiv:1908.11476*, 2020.

[6] Anirban Chaudhuri, Remi Lam, and Karen Willcox. Multifidelity uncertainty propagation via adaptive surrogates in coupled multidisciplinary systems. *AIAA Journal*, 56(1):235–249, 2018.

[7] P. Chen, A. Quarteroni, and G. Rozza. Comparison between reduced basis and stochastic collocation methods for elliptic problems. *Journal of Scientific Computing*, 59(1):187–216, 2014.

[8] Xiao Chen, Brenda Ng, Yunwei Sun, and Charles Tong. A flexible uncertainty quantification method for linearly coupled multi-physics systems. *Journal of Computational Physics*, 248:383 – 401, 2013.

[9] Xiaoxiao Chen, Eun-Jae Park, and Dongbin Xiu. A flexible numerical approach for quantification of epistemic uncertainty. *Journal of Computational Physics*, 240:211 – 224, 2013.

[10] Y. Chen, J.D. Jakeman, C. Gittelson, and D. Xiu. Local polynomial chaos expansion for linear differential equations with high dimensional random inputs. *SIAM Journal on Scientific Computing*, 37(1):A79–A102, 2015.

[11] P.G. Constantine, E.T. Phipps, and T.M. Wildey. Efficient uncertainty propagation for network multiphysics systems. *International Journal for Numerical Methods in Engineering*, 99(3):183–202, 2014.

[12] Andres A. Contreras, Paul Mycek, Olivier P. Le Maître, Francesco Rizzi, Bert Debusschere, and Omar M. Knio. Parallel domain decomposition strategies for stochastic elliptic equations part b: Accelerated monte carlo sampling with local pc expansions. *SIAM Journal on Scientific Computing*, 40(4):C547–C580, 2018.

[13] A. Doostan, A. Validi, and G. Iaccarino. Non-intrusive low-rank separated approximation of high-dimensional stochastic models. *Computer Methods in Applied Mechanics and Engineering*, 263:42–55, 2013.

[14] Martin Eigel and Robert Gruhlke. 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*, 122(4):1001–1030, 2021.

[15] Howard C. Elman and Qifeng Liao. Reduced basis collocation methods for partial differential equations with random coefficients. *SIAM/ASA Journal on Uncertainty Quantification*, 1(1):192–217, 2013.

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

[17] Sam Friedman, Benson Isaac, Seyede Fatemeh Ghoreishi, and Douglas L Allaire. Efficient decoupling of multiphysics systems for uncertainty propagation. *Proceedings of the AIAA SciTech Forum*, 2018.

[18] R.G. Ghanem and P.D. Spanos. *Stochastic Finite Elements: A Spectral Approach*. Springer-Verlag New York, Inc., New York, NY, USA, 1991.

[19] SF Ghoreishi and DL Allaire. Adaptive uncertainty propagation for coupled multidisciplinary systems. *AIAA journal*, 55(11):3940–3950, 2017.

[20] Alex A. Gorodetsky and John D. Jakeman. Gradient-based optimization for regression in the functional tensor-train format. *Journal of Computational Physics*, 374:1219 – 1238, 2018.

[21] Andrzej Granas and James Dugundji. Fixed point theory. *Springer Monographs in Mathematics*, 2003.

[22] A. Haji-Ali, F. Nobile, L. Tamellini, and R. Tempone. Multi-index stochastic collocation for random pdes. *Computer Methods in Applied Mechanics and Engineering*, 306:95 – 122, 2016.

[23] A.-L. Haji-Ali, F. Nobile, L. Tamellini, and R. Tempone. Multi-index Stochastic Collocation convergence rates for random PDEs with parametric regularity. *Foundations of Computational Mathematics*, 16(6):1555–1605, 2016.

[24] H. Harbrecht, J.D. Jakeman, and P. Zaspel. Cholesky-based experimental design for gaussian process and kernel-based emulation and calibration. *Communications in Computational Physics*, 2021. In press.

[25] Benson Isaac, Sam Friedman, and Douglas L Allaire. Efficient approximation of coupling variable fixed point sets for decoupling multidisciplinary systems. *Proceedings of the AIAA SciTech Forum*, 2018.

[26] J. D. Jakeman. Pyapprox: Probabilistic analysis and approximation of data and simulation. https://sandialabs.github.io/pyapprox/index.html, 2021.

[27] J.D. Jakeman, M. Eldred, and D. Xiu. Numerical approach for quantification of epistemic uncertainty. *Journal of Computational Physics*, 229(12):4648–4663, 2010.

[28] J.D. Jakeman, M.S. Eldred, G. Geraci, and A. Gorodetsky. Adaptive multi-index collocation for uncertainty quantification and sensitivity analysis. *International Journal for Numerical Methods in Engineering*, 2019.

[29] J.D. Jakeman and S.G. Roberts. Local and dimension adaptive stochastic collocation for uncertainty quantification. In Jochen Garcke and Michael Griebel, editors, *Sparse Grids and Applications*, volume 88 of *Lecture Notes in Computational Science and Engineering*, pages 181–203. Springer Berlin Heidelberg, 2013.

[30] Ksenia N. Kyzyurova, James O. Berger, and Robert L. Wolpert. Coupling computer models through linking their statistical emulators. *SIAM/ASA Journal on Uncertainty Quantification*, 6(3):1151–1171, 2018.

[31] Kaiyu Li and Douglas Allaire. 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.

[32] A. Manzoni, S. Pagani, and T. Lassila. Accurate solution of bayesian inverse uncertainty quantification problems combining reduced basis methods and reduction error models. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):380–412, 2016.

[33] Marque-Pucheu, Sophie, Perrin, Guillaume, and Garnier, Josselin. Efficient sequential experimental design for surrogate modeling of nested codes. *ESAIM: PS*, 23:245–270, 2019.

[34] A. Mittal, X. Chen, C. H. Tong, and G. Iaccarino. A flexible uncertainty propagation framework for general multiphysics systems. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):218–243, 2016.

[35] Lin Mu and Guannan Zhang. A domain decomposition model reduction method for linear convection-diffusion equations with random coefficients. *SIAM Journal on Scientific Computing*, 41(3):A1984–A2011, 2019.

[36] Akil Narayan and John D Jakeman. Adaptive Leja sparse grid constructions for stochastic collocation and high-dimensional approximation. *SIAM/J. Sci. Comput*, 36(6):2952–2983, 2014.

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

[38] I. V. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[39] C. Piazzola, L. Tamellini, R. Pellegrini, R. Broglia, A. Serani, and M. Diez. Uncertainty Quantification of Ship Resistance via Multi-Index Stochastic Collocation and Radial Basis Function Surrogates: A Comparison. *Proceedings of the AIAA Aviation Forum 2020*, 2020.

[40] Tong Qin, Zhen Chen, John D. Jakeman, and Dongbin Xiu. Deep learning of parameterized equations with applications to uncertainty quantification. *International Journal for Uncertainty Quantification*, 11(2):63–82, 2021.

[41] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[42] R.C. Reed. *The Superalloys: Fundamentals and applications*. Cambridge University Press, New York, 2006.

[43] J. Rogers. *DeMAID/GA - An enhanced design manager's aid for intelligent decomposition*. 2012.

[44] G. Rozza, D.B.P. Huynh, and A.T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Archives of Computational Methods in Engineering*, 15(3):1–47, 2007.

[45] Jerome Sacks, William J. Welch, Toby J. Mitchell, and Henry P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–423, 1989.

[46] Shankar Sankararaman and Sankaran Mahadevan. Likelihood-based approach to multidisciplinary analysis under uncertainty. *Journal of Mechanical Design*, 134(3), 2012.

[47] Francois Sanson, Olivier Le Maitre, and Pietro Marco Congedo. Systems of gaussian process models for directed chains of solvers. *Computer Methods in Applied Mechanics and Engineering*, 352:32 – 55, 2019.

[48] Kathrin Smetana and Anthony T. Patera. Optimal local approximation spaces for component-based static condensation procedures. *SIAM Journal on Scientific Computing*, 38(5):A3318–A3356, 2016.

[49] B. Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering & System Safety*, 93(7):964–979, JUL 2008.

[50] D. Xiu and J.S. Hesthaven. High-order collocation methods for differential equations with random inputs. *SIAM Journal on Scientific Computing*, 27(3):1118–1139, 2005.

[51] D. Xiu and G.E. Karniadakis. The Wiener-Askey Polynomial Chaos for stochastic differential equations. *SIAM J. Sci. Comput.*, 24(2):619–644, 2002.

[52] Kais Zaman and Sankaran Mahadevan. Robustness-based design optimization of multidisciplinary system under epistemic uncertainty. *AIAA Journal*, 51(5):1021–1031, 2013.

[53] Yinhao Zhu and Nicholas Zabaras. Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification. *Journal of Computational Physics*, 366:415 – 447, 2018.