

Design and Analysis of Multifidelity Finite Element Simulations

Henry Shaowu Yuchi

H. Milton Stewart School of Industrial and Systems Engineering,
Georgia Institute of Technology,
Atlanta, GA 30332
e-mail: shaowu.yuchi@gatech.edu

V. Roshan Joseph¹

A. Russell Chandler III Chair and Professor
H. Milton Stewart School of Industrial and Systems Engineering,
Georgia Institute of Technology,
Atlanta, GA 30332
e-mail: roshan@gatech.edu

C. F. Jeff Wu

Coca-Cola Chair in Engineering Statistics and Professor
H. Milton Stewart School of Industrial and Systems Engineering,
Georgia Institute of Technology,
Atlanta, GA 30332
e-mail: jeff.wu@isye.gatech.edu

The numerical accuracy of finite element analysis (FEA) depends on the number of finite elements used in the discretization of the space, which can be varied using the mesh size. The larger the number of elements, the more accurate the results are. However, the computational cost increases with the number of elements. In current practice, the experimenter chooses a mesh size that is expected to produce a reasonably accurate result, and for which the computer simulation can be completed in a reasonable amount of time. Improvements to this approach have been proposed using multifidelity modeling by choosing two or three mesh sizes. However, mesh size is a continuous parameter, and therefore, multifidelity simulations can be performed easily by choosing a different value for the mesh size for each of the simulations. In this article, we develop a method to optimally find the mesh sizes for each simulation and satisfy the same time constraints as a single or a double mesh size experiment. A range of different mesh sizes used in the proposed method allows one to fit multifidelity models more reliably and predict the outcome when meshes approach infinitesimally small, which is impossible to achieve in actual simulations. We illustrate our approach using an analytical function and a cantilever beam finite element analysis experiment. [DOI: 10.1115/1.4056874]

Keywords: approximation-based optimal design, design of experiments, metamodeling, space-filling design

1 Introduction

Computer experiments are widely used to simulate engineering applications, where physical experiments are prohibitively expensive or challenging to carry out. Statistical design and analysis of computer experiments came into prominence through the seminal works of Sacks et al. [1] and Currin et al. [2]. See the book by Santner et al. [3] for details. More recently, multifidelity modeling has become an important topic in computer experiments. For example, see the applications in satellite systems [4], laser melting [5], and polymers [6]. Multifidelity simulations are motivated by the fact that combinations of simulations with different levels of fidelity can be utilized to improve the system estimation overall, making the predictions more accurate and computationally efficient. Kennedy and O'Hagan [7] introduced a Gaussian process framework for multifidelity modeling, which is extended by other researchers [8,9], to name a few.

Space-filling designs [10] are commonly used for computer experiments, but the introduction of multiple fidelity calls for new approaches. Qian [11] proposed the idea of nested space-filling designs where the higher fidelity simulations are nested within the lower fidelity simulations. Sequential design methods that incorporate multifidelity have also been proposed in the literature [12–14]. However, they restrict the selection of fidelities to only a few discrete levels, which are insufficient when the fidelity can be changed continuously such as in finite element analysis (FEA).

FEA typically divides a larger geometric domain into smaller and simpler cells and then aims to solve partial differential equations on them instead. These cells are called mesh elements, and this representation is called meshing [15]. It is known that more mesh elements, i.e., finer meshes, lead to more accurate simulation results, but it will inevitably consume more computational power. Simulations with fewer mesh elements, though less accurate, are often

cheaper to conduct. In practice, we can adjust the number of mesh elements in FEA, achieving a trade-off between accuracy and cost.

Thus, multifidelity simulations in FEA differs from the usual problems because the fidelity can be changed almost continuously using the mesh size or the number of elements. There has been some research in multifidelity modeling in FEA [16,17], where the density of finite elements is connected to fidelity. However, the challenge of constructing experimental design under these circumstances remains. Moreover, the computational expenditure as a constraint for the designs has not been well investigated. We aim to address them both in this work.

We propose a new method to construct the experimental design where each simulation has a different mesh number/size. The proposed method targets finite element simulations with uniform meshes, where mesh density can be directly calculated from the dimensions of the uniform mesh elements. The method is novel in that it integrates computational costs for simulations into the design strategy, which is a practical matter for engineers. We will describe in detail how the experimenter can choose different mesh sizes/mesh element numbers to complete the full set of simulations within the given computational budget. We would also demonstrate how simulations performed at various mesh sizes can be integrated to produce a predictive model over the entire experimental region that is more accurate than others acquired from simulations with one or two levels the of mesh size.

This article is organized as follows. We will briefly review the commonly used space-filling experimental design methods and multifidelity models in Sec. 2. We then develop our new design in Sec. 3. Finally, we demonstrate the performance of the new design in two applications in Sec. 4: a simulation study on a function with a scalar response and an FEA on beam deflection with a functional response. Some concluding remarks are made in Sec. 5.

¹Corresponding author.

Contributed by the Design Automation Committee of ASME for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received April 2, 2022; final manuscript received January 30, 2023; published online March 15, 2023. Assoc. Editor: Julián Norato.

2 Related Work

In this section, we will review a few existing works in space-filling designs and multifidelity modeling, which are pertinent to this work.

2.1 Experimental Design. In the design of deterministic computer experiments, space-filling designs that spread out the design points in the experimental region are commonly used [3]. Johnson et al. [18] propose two strategies based on the distance between the design points: minimax designs minimize the maximum gap in the experimental region, whereas maximin designs maximize the minimum distance between the points. Since our work is closely related to maximin designs, we will explain it in a bit more detail. See Joseph [10] for a recent review on space-filling designs.

Suppose we are interested in exploring the relationship between the output y and input variables x_1, \dots, x_p . We can scale the input variables so that the experimental region is a unit hypercube $[0, 1]^p$. Let $\mathbf{D} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ be the experimental design, where $\mathbf{x}_i \in [0, 1]^p$ for $i = 1, \dots, n$. Let $\|\mathbf{x}_i - \mathbf{x}_j\|$ be the Euclidean distance between points \mathbf{x}_i and \mathbf{x}_j . Then the maximin design is obtained as the solution to the following optimization problem:

$$\max_D \min_{i \neq j} \|\mathbf{x}_i - \mathbf{x}_j\| \quad (1)$$

An issue with the maximin design is that some of the design points may project onto the same value for some input variables. This is undesirable because only a few inputs may be important, and therefore, replicated values in them are not useful in a deterministic computer experiment that has no random errors. Latin hypercube design (LHD) proposed by Ref. [19] is a solution to this problem. It ensures that the design points project onto n different values in each input variable. However, there can be many LHDs. Therefore, an optimum LHD can be obtained by maximizing the minimum distance among the points, which is called a maximin LHD [20]. Joseph et al. [21] proposed the maximum projection (MaxPro) design that tends to maximize the minimum distance among projections to all possible subsets of inputs.

To accommodate the idea of multifidelity in experiment design, we need to account for the situation where some of the design points will be more valuable than others due to different levels of accuracy. Nested LHDs are specifically tailored to such scenarios [11,22,23]. For example, for two fidelity levels, the set of design points as a whole is an LHD. Meanwhile, it contains a subset that is also an LHD (of smaller size). The whole set of points is used for the low-fidelity experiment, while the subset LHD is for the high-fidelity experiment.

Sequential design strategies for multifidelity simulations are also proposed in the literature [24–26]. In contrast, this article focuses on developing a fixed design strategy that is tailored for multifidelity finite element simulations. Interestingly, almost all of the sequential strategies need an initial set, and therefore, even if one is interested in sequential simulations, the fixed design developed in this work can be utilized for constructing the initial design.

2.2 Multifidelity Modeling. In the seminal work of Kennedy and O'Hagan [7], the authors model the correlation between simulation outputs of high- and low-fidelity levels with an autoregressive function.

$$y_k(\mathbf{x}) = \rho_{k-1} y_{k-1}(\mathbf{x}) + \delta_k(\mathbf{x}) \quad (2)$$

where $y_k(\mathbf{x})$ is the output at fidelity level k (smaller k indicates lower fidelity), ρ_{k-1} is a scale factor, and $\delta_k(\mathbf{x})$ is the bias for $k = 1, \dots, K$. The bias term $\delta_k(\mathbf{x})$ is modeled by a stationary Gaussian process

with a Gaussian covariance function [3]:

$$\text{Cov}[\delta_k(\mathbf{x}), \delta_k(\mathbf{x}')] = \sigma_k^2 \exp \left\{ - \sum_{i=1}^p \theta_{k,i} (\mathbf{x}_i - \mathbf{x}'_i)^2 \right\} \quad (3)$$

where $\theta_{k,i}$ denotes the correlation parameter in the i th dimension at the k th fidelity level. The foregoing model can be used for integrating data from all the fidelity levels and use it for predicting the response at the highest fidelity level. In this article, we refer to this modeling approach as KOH. See Fernández-Godino et al. [27] for a recent review of other multifidelity modeling methods.

However, the KOH model gets overly complex when there are numerous fidelity levels (large K), which is especially true in FEA because the fidelity can be easily changed by varying the mesh density. The work of Ref. [16] addresses it by regarding the mesh density/size as a tuning parameter for the system. Denote the mesh density tuning variable by t . It is assumed that $t > 0$, and a smaller t indicates a higher mesh density, implying higher simulation accuracy. Tuo et al. [16] proposed the following model:

$$y(\mathbf{x}, t) = y(\mathbf{x}, 0) + \delta(\mathbf{x}, t) = \phi(\mathbf{x}) + \delta(\mathbf{x}, t) \quad (4)$$

where $\phi(\mathbf{x})$ is the true response and $\delta(\mathbf{x}, t)$ denotes the bias in the simulation at input \mathbf{x} under mesh density t . The true response function $\phi(\mathbf{x})$ is unattainable in FEA because it is impossible to run the simulation with $t = 0$. The beauty of Tuo et al.'s approach is that by modeling data from different fidelity levels, we can extrapolate and predict the response at $t = 0$. Similar to the model used in Ref. [7], the two terms $\phi(\mathbf{x})$ and $\delta(\mathbf{x}, t)$ can be modeled by two independent Gaussian processes, where $\phi(\mathbf{x})$ has a stationary covariance function:

$$\text{Cov}[\phi(\mathbf{x}_1), \phi(\mathbf{x}_2)] = K_0(\mathbf{x}_1, \mathbf{x}_2) = \sigma_1^2 \exp \left\{ - \sum_{s=1}^p \theta_s^{(0)} (x_{1s} - x_{2s})^2 \right\} \quad (5)$$

However, $\delta(\mathbf{x}, t)$ cannot be modeled by a stationary Gaussian process because $\delta(\mathbf{x}, t) \rightarrow 0$ as $t \rightarrow 0$ regardless of \mathbf{x} , breaking the necessary condition for a stationary process. Therefore, Tuo et al. [16] use a nonstationary covariance function involving Brownian motion:

$$\begin{aligned} \text{Cov}[\delta(\mathbf{x}_1, t_1), \delta(\mathbf{x}_2, t_2)] &= \sigma_2^2 K_1(\mathbf{x}_1, \mathbf{x}_2) \min(t_1, t_2)^l \\ &= \sigma_2^2 \exp \left\{ - \sum_{s=1}^p \theta_s^{(1)} (x_{1s} - x_{2s})^2 \right\} \min(t_1, t_2)^l \end{aligned} \quad (6)$$

where l is a covariance parameter. There are a few choices for l but typically it is set to 4 motivated by the error analysis of a finite element simulation. Note that $\text{Var}[\delta(\mathbf{x}, t)] = \sigma_2^2 t^l \rightarrow 0$ as $t \rightarrow 0$, and therefore, the bias disappears from the model.

Tuo et al.'s model is more flexible than the KOH model because each simulation is free to select a different mesh number/density parameter. Moreover, this model contains only $2p$ unknown correlation parameters as opposed to Kp unknown parameters in KOH, making the estimation easier when $K > 2$. However, it poses a challenge to the experimental design strategy. While space-filling design for system inputs is relatively well studied, determining the corresponding fidelity parameters is not. To meet this challenge, we propose a more flexible and versatile experimental design method that is accommodative to finite element simulations with various mesh densities and mesh element numbers. We call it *multi-mesh experimental design* (MMED), which is discussed in Sec. 3.

3 Multi-Mesh Experimental Design

The aim is to create an n -point experimental design $\mathbf{D} = \{(\mathbf{x}_1, t_1), \dots, (\mathbf{x}_n, t_n)\}$, where each simulation can have a different mesh density. Since the geometry of the meshing element can differ depending on the type of the meshing system used in the FEA, it will be convenient to use the number of mesh elements (M) as the tuning parameter instead of the mesh density. They are related by the relation

$$t \propto \frac{1}{M^{1/k}} \quad (7)$$

where k is the dimension of the mesh (usually 2 or 3). Thus, $t \rightarrow 0$ when $M \rightarrow \infty$. In other words, the larger the M is, the more accurate the results will be. However, the computational cost increases with M . Assume that the computational time (T) of a single simulation is related to the number of mesh elements by

$$T = \beta M^\alpha \quad (8)$$

where β is a proportionality constant that depends on the meshing characteristics such as geometry and adaptation, and α is a positive constant. For ease of exposition, the rest of this article considers the case of $\alpha = 1$, which seems to be a reasonable assumption [28]. The case of $\alpha \neq 1$ is given in the Appendix. We will also restrict our attention to uniform meshes, although there is evidence that the linearity holds with certain mesh adaptation as well [29].

To perform simulations given the number of mesh elements M , we need to make sure that the mesh arrangement is reasonable and the simulation is able to converge. Suppose FEA simulations within a range of \tilde{M} and \bar{M} converge and produce reasonable results. Then if we were to perform the whole set of simulations at a single fidelity level, we can choose the upper bound \bar{M} as the mesh element number. If we were to pursue a multifidelity scenario, then choosing two different fidelity levels is a common approach. We can designate meshing with \tilde{M} elements as the low-fidelity level, and that with \bar{M} elements as the high-fidelity level. In this scenario, we will be able to perform more simulations at \tilde{M} compared to \bar{M} and incur the same computational time. That is, for each simulation performed at \bar{M} , we can do (\bar{M}/\tilde{M}) simulations at \tilde{M} . For a multi-mesh experimental design, we will choose $M_1, \dots, M_n \in [\tilde{M}, \bar{M}]$ while at the same time ensure the total computational time is still within the budget.

Suppose we have a budget to perform \bar{n} simulations at \bar{M} . Thus, the total time budget is $\bar{n} \times \beta \bar{M}$. We would like to divide this budget for simulations at n distinct levels M_1, \dots, M_n . Thus, $\sum_{i=1}^n \beta M_i = \bar{n} \beta \bar{M}$. Therefore, the levels need to satisfy the constraint

$$\sum_{i=1}^n M_i = \bar{n} \bar{M} \quad (9)$$

How should the n levels be chosen so that Eq. (9) is satisfied? One possibility is to find the levels so that it minimizes the integrated mean squared error criterion [3] under Tuo et al.'s [16] multifidelity model. However, such a criterion would involve the correlation parameters of the Gaussian process models, which are unknown before doing the simulations. Therefore, a more robust approach is to use a space-filling design. If we were to use a maximin design in $[\tilde{M}, \bar{M}]$, we would choose the levels using

$$\max_{\mathbf{d}} \min_{i \neq j} |M_i - M_j| \quad (10)$$

where $\mathbf{d} = \{M_1, \dots, M_n\}$. Since this is a one-dimensional case, the maximin design is an equidistant point set given by Ref. [18]

$$M_i = \tilde{M} + \frac{i-1}{n-1}(\bar{M} - \tilde{M}), \quad i = 1, \dots, n$$

where n can be chosen to approximately satisfy Eq. (9). However, this solution spreads the levels uniformly within $[\tilde{M}, \bar{M}]$, which

does not make sense because we expect to have more simulations near \tilde{M} than near \bar{M} . This can be achieved by using a weighted maximin design

$$\max_{\mathbf{d}} \min_{i \neq j} w_i w_j |M_i - M_j| \quad (11)$$

where the weights w_i 's should be inversely proportional to M_i . The weighted maximin design is also known by the name minimum energy design [30,31], which has the property that the optimal design points will converge to a distribution with density proportional to w^2 . Thus, if we take $w_i = 1/\sqrt{M_i}$, then the design points will have a distribution proportional to $1/M$. This makes sense because under the assumption in Eq. (8), the number of simulations that can be carried out at level M for a given time is inversely proportional to M . Therefore, the n levels can be obtained by solving the optimization problem:

$$\max_{\mathbf{d}} \min_{i \neq j} \frac{1}{\sqrt{M_i M_j}} |M_i - M_j| \quad (12)$$

There is no explicit solution to the foregoing optimization problem. However, an approximate solution can be obtained using transformations as follows. As shown in Refs. [30,31], the solution to Eq. (12) will asymptotically converge to a distribution with density function:

$$\begin{aligned} f(M) &= \frac{1/M}{\int_{\tilde{M}}^{\bar{M}} 1/M \, dM} \\ &= \frac{1/M}{\log(\bar{M}/\tilde{M})} \end{aligned}$$

for $M \in [\tilde{M}, \bar{M}]$. The cumulative distribution function is given by

$$F(M) = \frac{\log(M/\tilde{M})}{\log(\bar{M}/\tilde{M})}, \quad \text{for } M \in [\tilde{M}, \bar{M}]$$

Thus, $u = F(M)$ has a uniform distribution in $[0, 1]$. Therefore, under this transformation, Eq. (12) becomes

$$\max_{\mathbf{d}} \min_{i \neq j} |u_i - u_j|$$

The solution to this problem is given by Ref. [18]

$$u_i = \frac{i-1}{n-1}, \quad i = 1, \dots, n$$

Thus, the solution to (12) is given by

$$\begin{aligned} M_i &= F^{-1}(u_i) \\ &= \tilde{M}(\bar{M}/\tilde{M})^{u_i}, \quad i = 1, \dots, n \end{aligned} \quad (13)$$

Substituting Eq. (13) into Eq. (9), we obtain

$$\sum_{i=1}^n (\bar{M}/\tilde{M})^{u_i} = \bar{n}(\bar{M}/\tilde{M}) \quad (14)$$

Let $\lambda = \bar{M}/\tilde{M}$. Then, Eq. (14) becomes

$$\sum_{i=1}^n \lambda^{i-1} = \bar{n} \lambda$$

which gives

$$\frac{\lambda^{n/(n-1)} - 1}{\lambda^{1/(n-1)} - 1} = \bar{n} \lambda$$

By solving for n and choosing the nearest integer solution, we

obtain

$$n = 1 + \left\lceil \frac{\log \lambda}{\log \kappa} \right\rceil \quad (15)$$

where $\kappa = (\bar{n}\lambda - 1)/(\lambda(\bar{n} - 1))$ and $\lceil x \rceil$ denotes the nearest integer of x . Once n is obtained, $\{M_1, \dots, M_n\}$ can be obtained from Eq. (13).

Now to get the MMED, $\mathbf{D} = \{(\mathbf{x}_1, M_1), \dots, (\mathbf{x}_n, M_n)\}$, we can first find an n -point MaxPro LHD [21] in $(p + 1)$ dimensions and replace the n levels of the last column with $\{M_1, \dots, M_n\}$. The design procedure is summarized in Algorithm 1.

Algorithm 1 MMED($p, \tilde{M}, \bar{M}, \bar{n}$): Multi-Mesh Experiment Design

Data: p : System input dimension,
 $[\tilde{M}, \bar{M}]$: Range of mesh element numbers,
 \bar{n} : No. of simulations at \bar{M} affordable by budget.

Find suitable n :
 Obtain the number of design points n from Eq. (15).

Space-filling design:

1. Obtain n levels for mesh elements from Eq. (13).
2. Obtain an n -run design in $(p + 1)$ dimensions using a space-filling design such as MaxPro LHD [21].
3. Assign the p input variables from the first p columns and the mesh elements from the remaining column of the design.

Output: Experimental design $\mathbf{D} = \{(\mathbf{x}_i, M_i)\}_{i=1}^n$.

4 Applications

In this section, we apply MMED to two applications: an analytical function and a cantilever beam deflection. To evaluate MMED, we compare it to two other design methods and their corresponding modeling practices. The first method is a set of space-filling points with simulations executed with the same number of mesh elements. We call it the single-mesh design as only one meshing arrangement is used throughout the finite element simulations. Under this setting, fidelity is not taken into account at all. The second method is a two-level nested Latin hypercube design [11] (briefly described in Sec. 2.1), where all simulations on level 1 are carried out with a mesh arrangement with fewer elements \tilde{M} , while simulations on level 2 are conducted with more mesh elements \bar{M} . The computational budget is equally split between the two fidelity levels because running simulations in parallel from two separate solvers makes the whole experiment finish at the same time. We call it the double-mesh design.

For modeling the data from the single-mesh simulations, we fit a standard Gaussian process with a Gaussian covariance function. For

double-mesh simulations, we use the KOH model [7] described at the beginning of Sec. 2.2. For MMED simulations, we use the non-stationary model of Ref. [16] described at the end of Sec. 2.2. The hyperparameters for all three models are estimated using maximum likelihood.

4.1 Analytical Function. We first evaluate the performance of MMED using a 2D analytical function [2]:

$$y(\mathbf{x}) = \left[1 - \exp\left(-\frac{1}{2x_2}\right) \right] \frac{2300x_1^3 + 1900x_1^2 + 2092x_1 + 60}{100x_1^3 + 500x_1^2 + 4x_1 + 20} \quad (16)$$

where $\mathbf{x} = [x_1, x_2] \in [0, 1]^2$. Although this is an analytical function, for illustration purposes, we predict the output from a mesh grid via a grid-based interpolation. The meshing process divides the input domain into many square elements in 2D. The response surface is shown in Fig. 1 with 25 mesh elements. We obtain the surface plot via the following procedure: (i) generate a uniform 5×5 square mesh grid in the input plane, (ii) evaluate the function outputs $y(\mathbf{x})$ by Eq. (16) at all these grid points, and (iii) carry out piecewise linear interpolation and form a continuous surface within each grid box from the discrete grid evaluations. Since there are two input variables, we use bilinear interpolation. To estimate the system response given input (p, q) , we first find the four corner points of the square mesh box in which the input point falls denoted by (p_1, q_1) , (p_1, q_2) , (p_2, q_1) , (p_2, q_2) . (By convention, $p_1 < p_2$ and $q_1 < q_2$.) The interpolated function response is then:

$$\hat{y}(p, q) = \frac{1}{(p_2 - p_1)(q_2 - q_1)} \begin{bmatrix} p_2 - p & p - p_1 \\ g(p_1, q_1) & g(p_1, q_2) \\ g(p_1, q_2) & g(p_2, q_2) \end{bmatrix} \begin{bmatrix} q_2 - q \\ q - q_1 \end{bmatrix} \quad (17)$$

This procedure is effectively the same as finite element meshing, splitting the input plane into many smaller squares. The number of mesh elements used controls the scale of these squares. For example, $M=25$ means the mesh grid contains 5×5 discrete points. The more mesh cells there are, the more accurate the surface becomes. Another plot of Eq. (16) is generated using $M=400$ mesh cells, as shown in Fig. 1. The interpolated surface becomes much more accurate because there are more mesh elements.

To evaluate the performance of MMED, we first designate a reasonable range for the number of mesh elements N , with $\tilde{M} = 16$ and $\bar{M} = 144$. Then we specify the computational time constraint to be equivalent to the total time of running ten simulations at $\bar{M} = 144$. Therefore, the single-mesh method would consist of ten simulations with \bar{M} . We use MaxPro LHD to find the ten design points in $[0, 1]^2$. For the double-mesh method, we propose a two-layer Latin

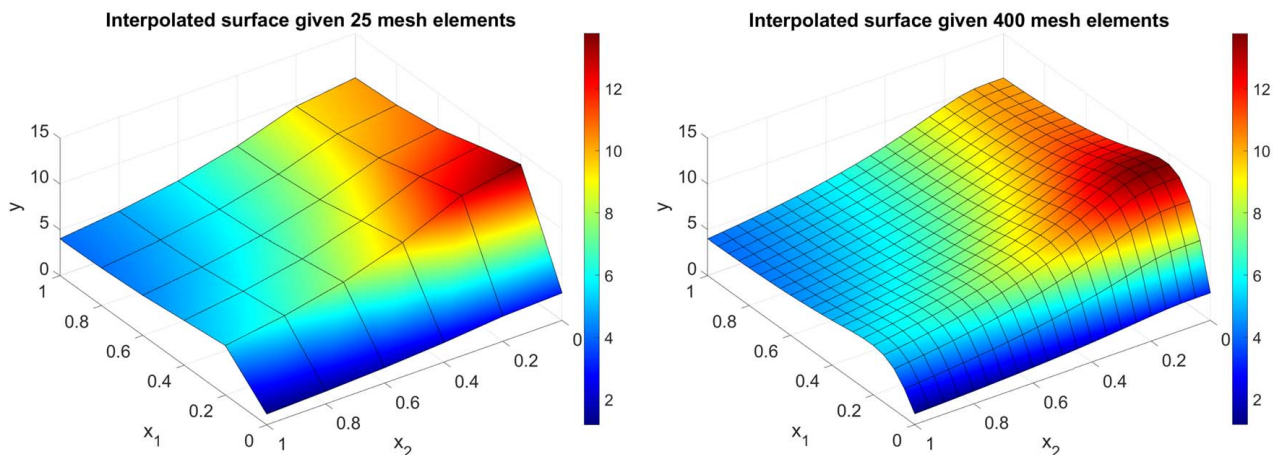


Fig. 1 Visualization of Eq. (16) given different mesh settings: $M = 25$ (left) and $M = 400$ (right)

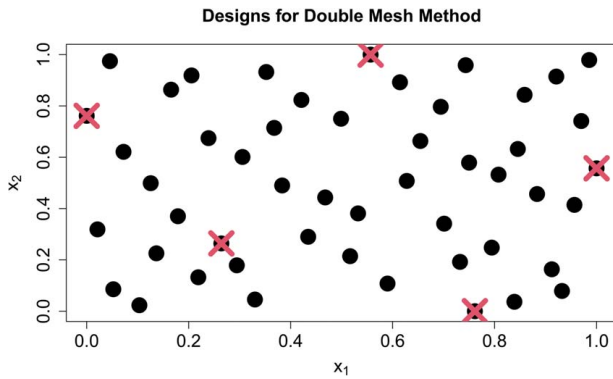


Fig. 2 The 50 nested design points generated by double-mesh method in $[0, 1]^2$. (The black circles represent points in the first layer (less accurate), while the red crosses represent points in the second layer (more accurate).)

hypercube nested design, where the simulations in the first layer use \bar{M} mesh elements and those in the second layer use \bar{M} mesh elements. We split the budget into half for simulations in each layer. It leads to 45 simulations in the first layer and five in the second layer (more precise). The nested design is illustrated in Fig. 2, which is constructed using the `MaxProAugment` function in the R package `MaxPro`. For our multi-mesh method, we follow Algorithm 1 to construct the design. We can obtain $n = 24$ design points with different values of $M \in [\bar{M}, \bar{M}]$ such that the total simulation time does not exceed the budget constraint. The scatter plots of

the design points in inputs and mesh element numbers are shown in Fig. 3. We can see the generated designs are space filling in each of the 2D projections plotted. The histogram plots on the diagonal show that the points fill the (x_1, x_2) space uniformly, whereas more points are allocated to the region with a smaller number of mesh elements.

To evaluate the performance of the three methods, we randomly draw 1000 system inputs \mathbf{x} as the testing dataset. For each of the three methods, we train the model using their design points and corresponding interpolation response. Subsequently, we obtain their predictions at the testing points, denoted by $\hat{y}_1, \dots, \hat{y}_{1000}$. The true response is calculated using Eq. (16) directly, denoted by y_1, \dots, y_{1000} . We use the root-mean-squared error (RMSE) as the performance metric:

$$\text{RMSE} = \sqrt{\frac{1}{1000} \sum_{i=1}^{1000} (y_i - \hat{y}_i)^2} \quad (18)$$

Repeating this procedure 30 times, we can obtain 30 RMSE values for each of the three methods. The results are plotted as a box plot in Fig. 4. The average RMSE over the 30 runs for the single-mesh, double-mesh, and multi-mesh methods are 2.00, 1.06, and 0.74, respectively. This clearly shows that MMED significantly outperforms the two other methods.

4.2 Cantilever Beam Deflection Under Stress. For the second application, we conduct static stress analysis on a cantilever beam using FEA simulations with cubical cells. The simulations are carried out using the `ABAQUS` software [32].

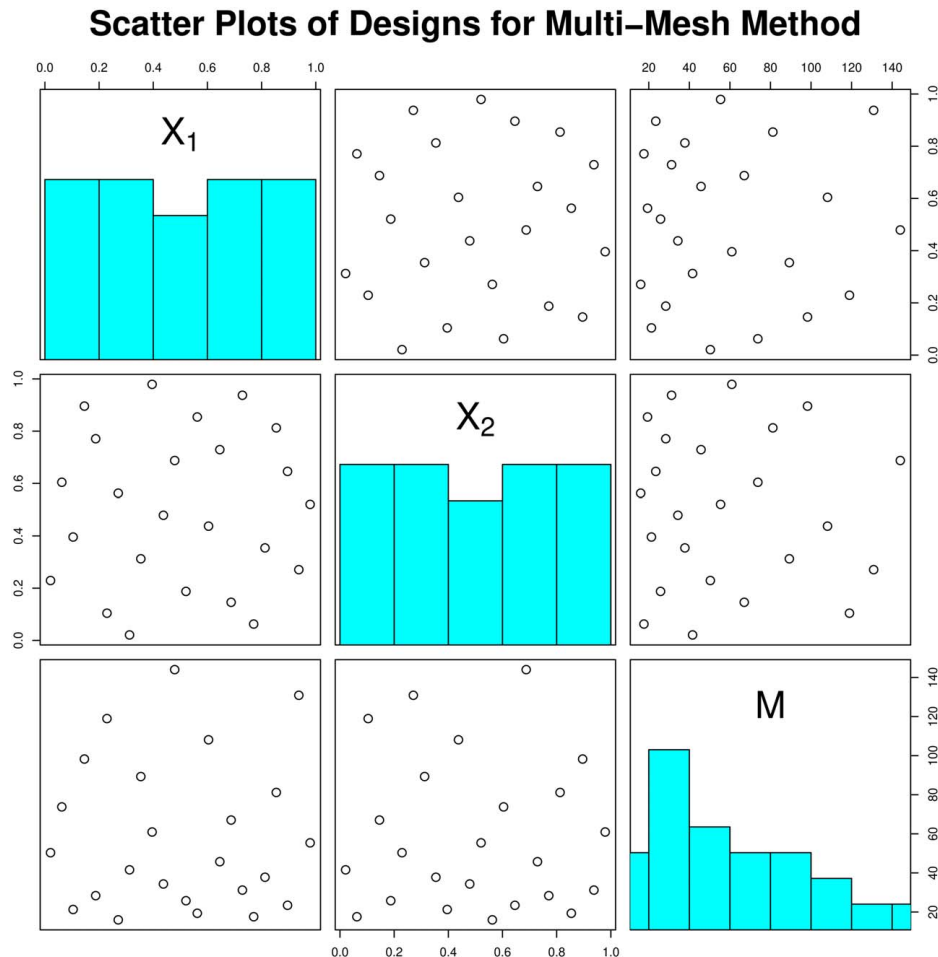


Fig. 3 The scatter plots for the 24 nested design points generated by the multi-mesh method

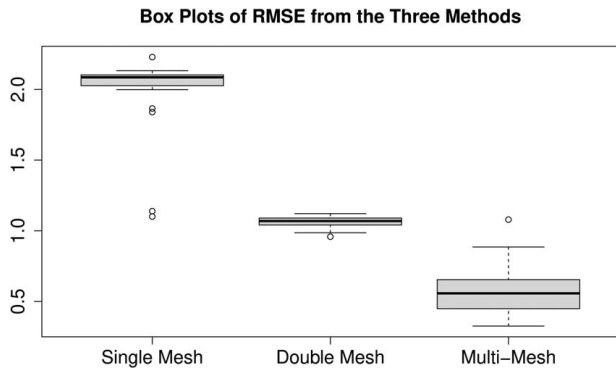


Fig. 4 Box plot of RMSE for the three methods over 30 runs in the analytical function example

For the beam structure, we set the dimensions as (d_1, d_2, d_3) corresponding to its breadth, height, and length, respectively. We set $d_1 = d_2$, so the cross section of the beam is square shaped. An illustration of the beam can be seen in Fig. 5. We set its Young's modulus to be 200 MPa, and Poisson ratio to be 0.28, which corresponds to steel. For the static stress analysis, one end surface along the length of the beam is fixed with no degree-of-freedom allowed, whereas the other surfaces are free to move. A continuous static half-sine pressure field is then applied vertically downward onto the top surface of the beam as shown in Fig. 5. For the deflection analysis due to this pressure field, we set three system inputs

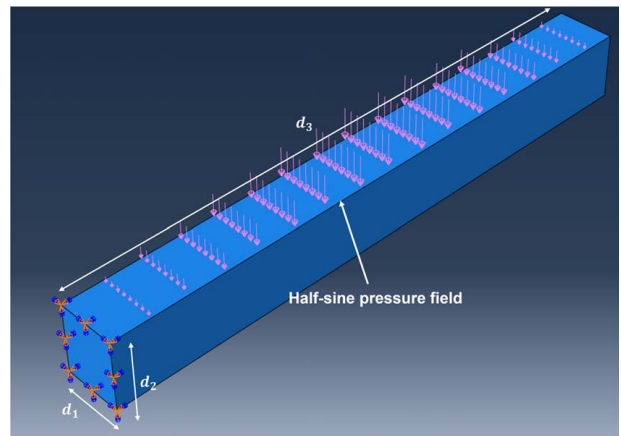


Fig. 5 Beam cantilever with fixed surface and pressure field (shown in ABAQUS interface)

variables $\mathbf{x} = [x_1, x_2, x_3] \in [0, 1]^3$ for this problem. The pressure field is described by a function depending on the span along d_3 from the fixed surface, denoted by $z \in [0, 200x_2]$:

$$\text{pressure} = 2000x_1 \sin(\pi z / 200x_2) \quad (19)$$

where input variable $x_1 \in [0, 1]$ is the weighting parameter, and $x_2 \in [0, 1]$ controls the length of the beam by $d_3 = 200x_2$. x_3 denotes the breadth and the height of the beam, corresponding to both d_1 and d_2 ,

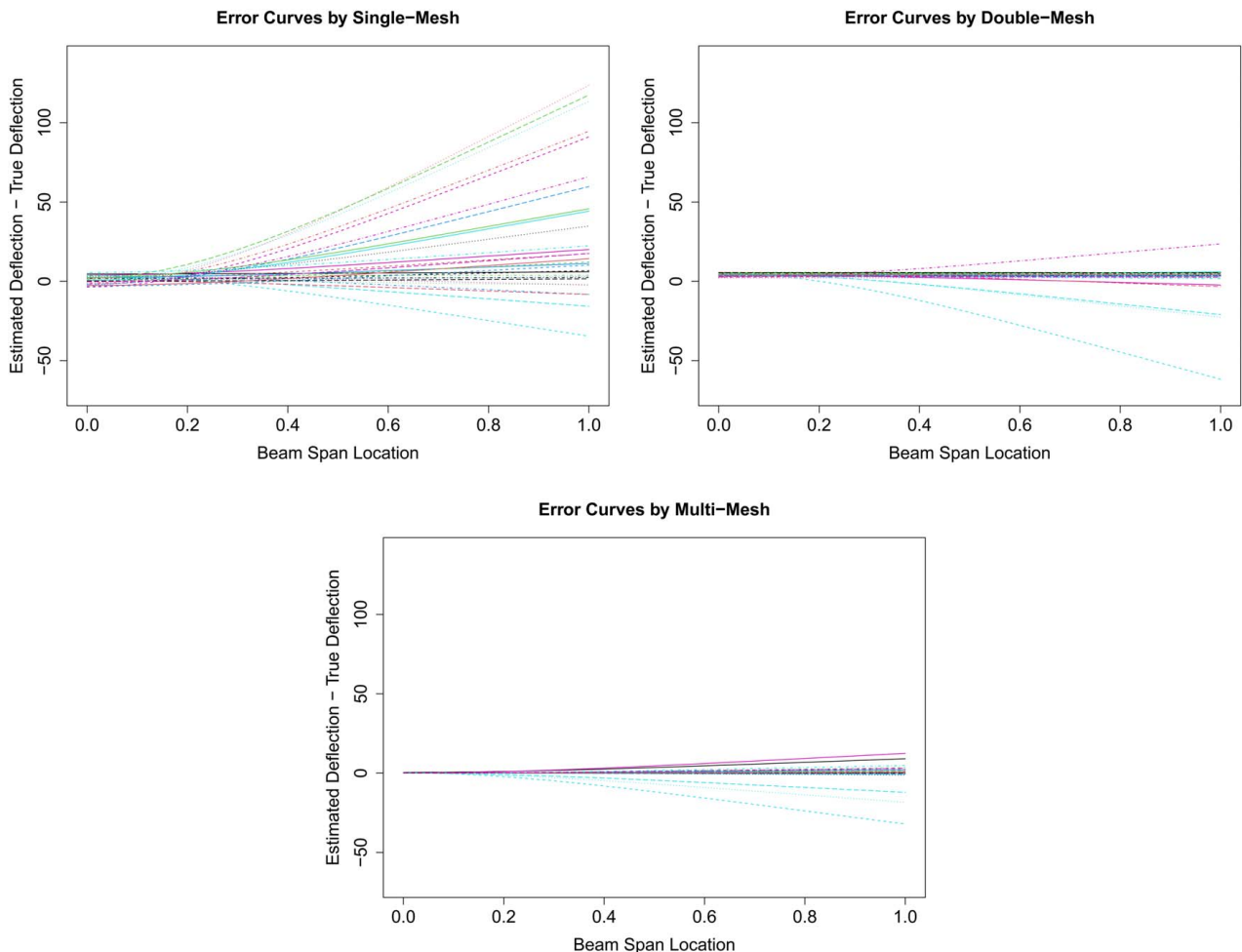


Fig. 6 Deflection profile estimation error curves by the three methods

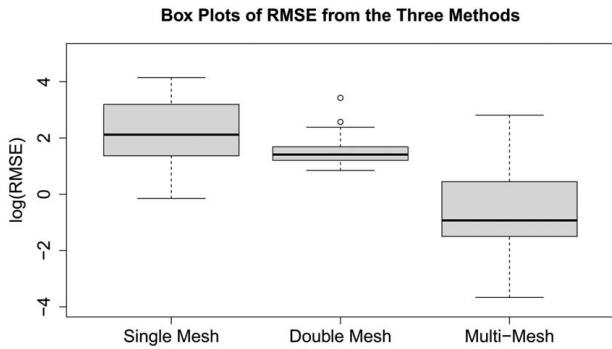


Fig. 7 Box plot of log-RMSE for the three methods in the cantilever beam example

respectively.

The deflection profile is measured across the span of the beam. We take the maximum deflection measurements along the beam at $z = 2x_2, 4x_2, \dots, 200x_2$, which gives us 100 uniformly placed discrete points to form the deflection profile as the functional response. Since the beam is a cuboid and the mesh elements are set to be cubic, we use $t = M^{-1/3}$ as the mesh density parameter. Overall, each simulation requires three input variables and one mesh density parameter.

Since the deflection profile is a functional response, it requires an additional variable, the measuring location on the span, to be put into the response model Eq. (6). We assume a Gaussian correlation function for this additional variable as well. The computations in functional Gaussian process modeling can be simplified using Kronecker products as described in Ref. [33].

For this application, suppose a reasonable range for mesh cell numbers is $\bar{M} = 1,000$ and $\bar{M} = 8,000$. Let the time budget be equivalent to the total computational time of running eight simulations with \bar{M} . Similar to the previous application, the single-mesh design with eight points in the input space $[0, 1]^3$ is obtained using MaxPro. The double-mesh design uses a two-layer nested design with half of the time budget allocated to each layer. The first layer consists of 32 simulations at \bar{M} , and the second layer contains four simulations at \bar{M} . Finally, for the multi-mesh method, $n = 18$ simulations with different values of $M \in [\bar{M}, \bar{M}]$ are obtained using Algorithm 1, which maintain the same total time constraint.

We randomly draw 30 sets of system inputs \mathbf{x} as the testing dataset. To obtain the corresponding deflection profile, we simulate each of these 30 sets with a corresponding finely meshed FEA model with $M = 320,000$. Under this setting, we mesh the beam using a $40 \times 40 \times 200$ grid. With such a large number of mesh elements, the size of each mesh element is small. As a result, we presume the deflection measurements from these simulations to be sufficiently accurate and can serve as the “true” response. To evaluate the performance of the three methods, we compare the estimated deflection profiles $\{\hat{y}_s\}_{s=1}^{100}$ by each of the methods at these testing points against the true profile $\{y_s\}_{s=1}^{100}$, where s denotes the index of the testing point.

The error curves $\{\hat{y}_s - y_s\}_{s=1}^{100}$ of the deflection profiles for the 30 testing cases under each of the three methods are plotted in Fig. 6. We can see that the errors in estimated deflection profiles by MMED are smaller than those by double-mesh and significantly smaller than those using single-mesh. The RMSE:

$$\text{RMSE} = \sqrt{\frac{1}{100} \sum_{s=1}^{100} (y_s - \hat{y}_s)^2} \quad (20)$$

is plotted as a box plot in Fig. 7. The p-values for the two-sample t-tests with unequal variances are 2.4×10^{-10} and 7.5×10^{-9} for multi-mesh versus single-mesh and multi-mesh versus double

mesh, respectively. Thus, the MMED significantly outperforms the other two methods in this application.

5 Conclusions

In this work, we have proposed an experimental design method that enables the experimenter to choose optimal mesh sizes for finite element simulations given a fixed time budget. We have shown that it outperforms the single-mesh and double-mesh approaches because the design is well coupled with the modeling method. The single-mesh approach does not take the concept of multifidelity into account, hampering its ability to take the effects of meshing into account. Kennedy and O’Hagan’s model [7] used in double mesh utilizes multifidelity, but it can predict the response only at the highest fidelity level used in the simulations. On the other hand, MMED naturally fits with the model proposed by Ref. [16], which helps to perform extrapolation and predict the true response that is impossible to achieve through simulations.

For future work, MMED can be refined and tailored to accommodate more complex finite element simulations where the meshing is no longer uniform, or the geometry is difficult to be easily described by the number of mesh elements generated alone. In FEA computer simulators such as ABAQUS and LS-DYNA, nonuniform meshes can often be generated by specifying mesh properties such as average or maximum/minimum sizes. Therefore, we expect MMED to remain effective for these scenarios, although this needs to be validated with complex finite element simulations. For nonuniform and adaptive mesh assignments, other variables will impact the computational time and simulation accuracy on top of mesh density. For instance, chordal errors referring to the quality of mesh approximation to true geometry can be considered as one such parameter. This would lead to multiple tuning parameters controlling the computational cost jointly, which goes beyond the scope of the current work where only one parameter is involved.

Acknowledgment

This work was supported by an LDRD grant from Sandia National Laboratories. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the US Department of Energy’s National Nuclear Security Administration (Contract No. DE-NA0003525). This article describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the article do not necessarily represent the views of the US Department of Energy or the United States Government. Wu was also supported by NSF DMS-1914632.

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The datasets generated and supporting the findings of this article are obtainable from the corresponding author upon reasonable request.

Appendix: MMED for $\alpha \neq 1$

Suppose $\alpha \neq 1$ in Eq. (8). Then, we should take the weights in Eq. (11) to be $w_i = 1/M_i^{\alpha/2}$ for $i = 1, \dots, n$ so that the optimal design points will have an asymptotic distribution proportional to $1/M^\alpha$ [30,31]. As mentioned earlier, let $\lambda = \bar{M}/\bar{M}$. Now following the same steps in Sec. 3, we obtain the optimal number of mesh

elements as follows:

$$M_i = \tilde{M} \left\{ 1 - \left(\frac{i-1}{n-1} \right) \left(1 - \frac{1}{\lambda^{\alpha-1}} \right) \right\}^{-\frac{1}{\alpha-1}}$$

for $i = 1, \dots, n$, where n needs to be solved from

$$\sum_{i=1}^n \left\{ 1 - \left(\frac{i-1}{n-1} \right) \left(1 - \frac{1}{\lambda^{\alpha-1}} \right) \right\}^{-\frac{1}{\alpha-1}} = \bar{n}\lambda$$

There is no explicit solution for n when $\alpha \neq 1$, and it needs to be solved numerically.

References

- [1] Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P., 1989, "Design and Analysis of Computer Experiments," *Statist. Sci.*, **4**(4), pp. 409–423.
- [2] Currin, C., Mitchell, T., Morris, M., and Ylvisaker, D., 1991, "Bayesian Prediction of Deterministic Functions, With Applications to the Design and Analysis of Computer Experiments," *J. Am. Stat. Assoc.*, **86**(416), pp. 953–963.
- [3] Santner, T. J., Williams, B. J., Notz, W. I., and Williams, B. J., 2003, *The Design and Analysis of Computer Experiments*, Vol. 1, Springer, New York.
- [4] Shi, R., Liu, L., Long, T., Wu, Y., and Gary Wang, G., 2020, "Multi-Fidelity Modeling and Adaptive Co-Kriging-Based Optimization for All-Electric Geostationary Orbit Satellite Systems," *ASME J. Mech. Des.*, **142**(2), p. 021404.
- [5] Olleak, A., and Xi, Z., 2020, "Calibration and Validation Framework for Selective Laser Melting Process Based on Multi-Fidelity Models and Limited Experiment Data," *ASME J. Mech. Des.*, **142**(8), p. 081701.
- [6] Patra, A., Batra, R., Chandrasekaran, A., Kim, C., Huan, T. D., and Ramprasad, R., 2020, "A Multi-Fidelity Information-Fusion Approach to Machine Learn and Predict Polymer Bandgap," *Comput. Mater. Sci.*, **172**, p. 109286.
- [7] Kennedy, M. C., and O'Hagan, A., 2000, "Predicting the Output From a Complex Computer Code When Fast Approximations Are Available," *Biometrika*, **87**(1), pp. 1–13.
- [8] Qian, Z., Seepersad, C. C., Joseph, V. R., Allen, J. K., and Wu, C. F. J., 2006, "Building Surrogate Models Based on Detailed and Approximate Simulations," *ASME J. Mech. Des.*, **128**(4), pp. 668–677.
- [9] Goh, J., Bingham, D., Holloway, J. P., Grosskopf, M. J., Kuranz, C. C., and Rutter, E., 2013, "Prediction and Computer Model Calibration Using Outputs From Multifidelity Simulators," *Technometrics*, **55**(4), pp. 501–512.
- [10] Joseph, V. R., 2016, "Space-Filling Designs for Computer Experiments: A Review," *Q. Eng.*, **28**(1), pp. 28–35.
- [11] Qian, P. Z., 2009, "Nested Latin Hypercube Designs," *Biometrika*, **96**(4), pp. 957–970.
- [12] Sarkar, S., Mondal, S., Joly, M., Lynch, M. E., Bopardikar, S. D., Acharya, R., and Perdikaris, P., 2019, "Multifidelity and Multiscale Bayesian Framework for High-Dimensional Engineering Design and Calibration," *ASME J. Mech. Des.*, **141**(12), p. 021404.
- [13] Liu, Y., Li, K., Wang, S., Cui, P., Song, X., and Sun, W., 2021, "A Sequential Sampling Generation Method for Multi-Fidelity Model Based on Voronoi Region and Sample Density," *ASME J. Mech. Des.*, **143**(12), p. 121702.
- [14] Stroh, R., Bect, J., Demeyer, S., Fischer, N., Marquis, D., and Vazquez, E., 2021, "Sequential Design of Multi-Fidelity Computer Experiments: Maximizing the Rate of Stepwise Uncertainty Reduction," *Technometrics*, **62**(2), pp. 1–11.
- [15] Zienkiewicz, O. C., Taylor, R. L., Zhu, J. Z., 2005, *The Finite Element Method: Its Basis and Fundamentals*, Elsevier, Amsterdam.
- [16] Tuo, R., Wu, C. F. J., and Yu, D., 2014, "Surrogate Modeling of Computer Experiments With Different Mesh Densities," *Technometrics*, **56**(3), pp. 372–380.
- [17] DiazDelaO, F., and Adhikari, S., 2012, "Bayesian Assimilation of Multi-Fidelity Finite Element Models," *Comput. Struct.*, **92–93**, pp. 206–215.
- [18] Johnson, M. E., Moore, L. M., and Ylvisaker, D., 1990, "Minimax and Maximin Distance Designs," *J. Stat. Plan. Inference*, **26**(2), pp. 131–148.
- [19] McKay, M. D., Beckman, R. J., and Conover, W. J., 2000, "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output From a Computer Code," *Technometrics*, **42**(1), pp. 55–61.
- [20] Morris, M. D., and Mitchell, T. J., 1995, "Exploratory Designs for Computational Experiments," *J. Stat. Plan. Inference*, **43**(3), pp. 381–402.
- [21] Joseph, V. R., Gul, E., and Ba, S., 2015, "Maximum Projection Designs for Computer Experiments," *Biometrika*, **102**(2), pp. 371–380.
- [22] Haaland, B., and Qian, P. Z., 2010, "An Approach to Constructing Nested Space-Filling Designs for Multi-Fidelity Computer Experiments," *Stat. Sinica*, **20**(3), p. 1063.
- [23] Qian, P. Z., Tang, B., and Wu, C. J., 2009, "Nested Space-Filling Designs for Computer Experiments With Two Levels of Accuracy," *Stat. Sinica*, **19**(1), pp. 287–300.
- [24] Huang, D., Allen, T. T., Notz, W. I., and Miller, R. A., 2006, "Sequential Kriging Optimization Using Multiple-Fidelity Evaluations," *Struct. Multidiscipl. Optim.*, **32**(5), pp. 369–382.
- [25] Le Gratiot, L., and Cannamela, C., 2015, "Cokriging-Based Sequential Design Strategies Using Fast Cross-Validation Techniques for Multi-Fidelity Computer Codes," *Technometrics*, **57**(3), pp. 418–427.
- [26] Chen, S., Jiang, Z., Yang, S., and Chen, W., 2017, "Multimodel Fusion Based Sequential Optimization," *AIAA J.*, **55**(1), pp. 241–254.
- [27] Fernández-Godino, M. G., Park, C., Kim, N. -H., and Hafika, R. T., 2019, "Issues in Deciding Whether to Use Multifidelity Surrogates," *AIAA J.*, **57**(5), pp. 2039–2054.
- [28] Marin, F., Souza, A. F. d., Pabst, R. G., and Ahrens, C. H., 2019, "Influences of the Mesh in the Cae Simulation for Plastic Injection Molding," *Polímeros*, **29**(3), p. 05019.
- [29] Pain, C., Umpheley, A., De Oliveira, C., and Goddard, A., 2001, "Tetrahedral Mesh Optimisation and Adaptivity for Steady-State and Transient Finite Element Calculations," *Comput. Methods. Appl. Mech. Eng.*, **190**(29–30), pp. 3771–3796.
- [30] Joseph, V. R., Dasgupta, T., Tuo, R., and Wu, C. J., 2015, "Sequential Exploration of Complex Surfaces Using Minimum Energy Designs," *Technometrics*, **57**(1), pp. 64–74.
- [31] Joseph, V. R., Wang, D., Gu, L., Lyu, S., and Tuo, R., 2019, "Deterministic Sampling of Expensive Posteriors Using Minimum Energy Designs," *Technometrics*, **61**(3), pp. 297–308.
- [32] Smith, M., 2009, "ABAQUS/Standard User's Manual," Version 6.9, Dassault Systèmes Simulia Corp, Providence, RI.
- [33] Hung, Y., Joseph, V. R., and Melkote, S. N., 2015, "Analysis of Computer Experiments With Functional Response," *Technometrics*, **57**(1), pp. 35–44.