AIAA JOURNAL
Vol. 50, No. 5, May 2012

Provably Convergent Multifidelity Optimization Algorithm Not Requiring High-Fidelity Derivatives

Andrew March* and Karen Willcox†
Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

DOI: 10.2514/1.J051125

This paper presents a provably convergent multifidelity optimization algorithm for unconstrained problems that does not require high-fidelity gradients. The method uses a radial basis function interpolation to capture the error between a high-fidelity function and a low-fidelity function. The error interpolation is added to the low-fidelity function to create a surrogate model of the high-fidelity function in the neighborhood of a trust region. When appropriately distributed spatial calibration points are used, the low-fidelity function and radial basis function interpolation generate a fully linear model. This condition is sufficient to prove convergence in a trust region framework. In the case when there are multiple lower-fidelity models, the predictions of all calibrated lower-fidelity models can be combined with a maximum likelihood estimator constructed using kriging variance estimates from the radial basis function models. This procedure allows for flexibility in sampling lower-fidelity functions, does not alter the convergence proof of the optimization algorithm, and is shown to be robust to poor low-fidelity information. The algorithm is compared with a single-fidelity quasi-Newton algorithm and two first-order consistent multifidelity trust region algorithms. For simple functions the quasi-Newton algorithm uses slightly fewer high-fidelity function evaluations; however, for more complex supersonic airfoil design problems it uses significantly more. In all cases tested, our radial basis function calibration approach uses fewer high-fidelity function evaluations when compared with first-order consistent trust region schemes.

Nomenclature

B	=	a closed and bounded set in $\mathbb{R}^n$
$\mathcal{B}$	=	trust region
C	=	differentiability class
c	=	constant ratio to the l_2 norm
$\mathbf{d}$	=	high-fidelity function sample point
$e(\mathbf{x})$	=	error model
$f(\mathbf{x})$	=	objective function
H	=	Hessian of the surrogate model
$\mathcal{L}$	=	expanded level-set in $\mathbb{R}^n$
L	=	level-set in $\mathbb{R}^n$
$\mathcal{M}$	=	space of all fully linear models
$m(\mathbf{x})$	=	surrogate model of the high-fidelity function
$\mathcal{N}(\cdot,\cdot)$	=	normal distribution
n	=	number of design variables
$p_{\max}$	=	maximum number of points to be used in the interpolation
r	=	radial distance between two points
$\mathbf{s}$	=	trust region step
$\mathbf{x}$	=	design vector
x_i	=	component i of the design vector
$\mathcal{Y}$	=	set of calibration vectors used in an error model
$\mathbf{y}$	=	vector from current iterate to a high-fidelity sample point
α	=	trust region contraction ratio used in convergence check
γ_0	=	trust region contraction ratio
γ_1	=	trust region expansion ratio
Δ	=	trust region radius
δx	=	finite difference step size
ϵ	=	termination tolerance for gradient norm
ϵ_2	=	termination tolerance for trust region size
η	=	trust region update criterion
θ	=	positive constant, used to build fully linear models
κ	=	bound related to function smoothness
λ	=	coefficient of the radial basis function interpolants
ξ	=	radial basis function correlation length
Π	=	polynomial basis for $\mathbb{R}^n$
π	=	component of the polynomial basis
ρ	=	ratio of actual to predicted improvement in high-fidelity function value
σ^2	=	variance of a Gaussian process
ν	=	coefficients of the polynomial interpolant
ϕ	=	radial basis function

Superscript

$*$	=	optimal

Subscripts

0	=	initial iterate
bhm	=	bound for Hessian of the surrogate model
blg	=	upper bound on the Lipschitz constant
est	=	maximum likelihood estimate
f	=	relating to function
FCD	=	fraction of Cauchy decrease
g	=	relating to gradient
high	=	relating to the high-fidelity function
i	=	index of a point included in the current error model basis
j	=	index of any high-fidelity sample point
k	=	index of trust region iteration number
low	=	relating to a lower-fidelity function
max	=	user-set maximum value of that parameter
min	=	user-set minimum value of that parameter
med	=	relating to an intermediate-fidelity function

Presented at the 6th AIAA Multidisciplinary Design Optimization Specialist Conference, Orlando, FL, 12–15 April 2010; received 13 December 2010; revision received 26 October 2011; accepted for publication 6 December 2011.

*Graduate Student, Department of Aeronautics and Astronautics; amarch@mit.edu. Student Member AIAA.

†Associate Professor, Department of Aeronautics and Astronautics; kwillcox@mit.edu. Associate Fellow AIAA.

I. Introduction

THE expense of either building or testing a complex system drives their designers to use computational algorithms to model their systems with as much accuracy as possible. The computational requirements of these high-fidelity analyses can be immense and designing systems using formal optimization methods with them is difficult, if not impractical. However, most system designers have other lower-fidelity models available that provide estimates of system performance with considerably lower computational requirements. Even if such models are not available, other approaches such as response surface methodology [1–3], reduced-order modeling [4], or using coarser discretizations [5], can generate surrogates of high-fidelity analyses that may be treated as lower-fidelity models. This paper presents a multifidelity optimization approach that employs low-fidelity information to systematically reduce use of the high-fidelity analysis during the optimization, but guarantees convergence to a high-fidelity optimal design.

There are several different multifidelity optimization strategies that optimize a high-fidelity function using a lower-fidelity surrogate. One class of approaches uses trust regions. These methods are provably convergent to a local optimum of the high-fidelity function, if at the center of the trust region the low-fidelity function value and derivative are scaled or shifted to be equal to the high-fidelity function and gradient [5–7]. However, often in engineering design the objectives are nonsmooth, or the high-fidelity analyses are experimental results, black box codes, or numerical methods that contain noise or that may fail to return a solution. In all of these situations, the gradient of the high-fidelity function is unavailable and estimating the gradient may be unreliable. In such cases, multifidelity methods such as sequential approximate optimization (SAO) [8], efficient global optimization (EGO) [9], or surrogate management framework (SMF) [10] can be used as they do not require high-fidelity gradient estimates. Though rigorous convergence guarantees of such derivative-free methods only holds for certain types of noise/nonsmoothness (for example, that discussed in [[11], section 9.3]) the motivation for these derivative-free methods over derivative-based methods is the robustness often observed in practice.

SAO is a term often used to describe engineering design optimization that uses low-fidelity models in place of high-fidelity simulation [5]. SAO techniques generally fit response surfaces to simulation results, perform optimization on the response surfaces, and then update the response surfaces until some stopping criterion is satisfied. A gradient-free class of SAO algorithms offers good performance for problems with multiple local minima and noisy simulations; however, these techniques are heuristic and convergence is problem dependent [12,13]. In EGO a Gaussian process regression model is fit to the high-fidelity objective function. The mean of the Gaussian process interpolates the value of the high-fidelity function, while the mean square error of the Gaussian process models the uncertainty in the high-fidelity function value. This error is zero at all locations where the value of the high-fidelity function is known and increases with distance away from sample points. Optimization is then performed on the Gaussian process model, and the high-fidelity function is sampled at locations likely to reduce to the value of the function over the current observed minimum. This technique works well in practice, can be proven to find a globally optimal solution (with modifications to the common definition of the method) [14,15], can be used in a multifidelity setting using Bayesian model calibration methods [16–18], and does not require a high-fidelity derivative estimate. However, the method may be globally biased and attempt to explore the entire design space as opposed to simply reducing the objective function. In addition, the method has been shown to be sensitive to both the initial high-fidelity samples [15] and to the exact metric of selecting points likely to improve the high-fidelity function value [19]. SMF is a derivative-free pattern-search method augmented with a prediction of a locally optimal design from a surrogate model. The underlying pattern-search method ensures convergence, so a broad range of surrogate models are allowable. Of specific interest is conformal space mapping where a low-fidelity function is calibrated to the high-fidelity function at locations where the function value is known [20].

In this paper, we present a provably convergent multifidelity optimization algorithm based on model calibration that does not require high-fidelity derivative estimates. The first-order-consistent trust region methods mentioned above can be thought of as employing model calibration; however, the calibration is only local and temporary, since sample points from previous iterations are not reused. The challenge we address here is twofold, first we must produce a surrogate model that captures local function behavior sufficiently well to prove convergence without requiring a high-fidelity gradient estimate, and secondly we must ensure the surrogate captures some global function behavior to speed convergence to a stationary point of the high-fidelity function. A trust region algorithm is convergent provided either the error between the gradient of the function and the gradient of surrogate model is bounded by a constant times the gradient of the function [21] or provided the accuracy of the surrogate can be improved dynamically within the trust region framework [[22], section 10.6]. Oeuvray [23] showed that a radial basis function interpolation satisfies these criteria, provided the interpolation points satisfy certain conditions. Conn et al. [24] then showed that both the error between a function and a smooth interpolation model as well as the error between the function's derivative and the interpolation model's derivative can be bounded by appropriately selecting interpolation points. Conn et al. [25] also proved that any interpolation model that can locally be made fully linear (defined in the next section) can be used in a provably convergent derivative-free trust region framework. Wild et al. [26], Wild [27], and Wild and Shoemaker [28] then developed an algorithm to produce fully linear radial basis function interpolation models and showed that their method could be used within Conn's provably convergent optimization framework.

This paper combines the provably convergent optimization frameworks of Wild et al. [26] and Conn et al. [25] with Bayesian model calibration ideas to result in a provably convergent multifidelity optimization approach that does not require high-fidelity gradient information. Section II provides an overview of the derivative-free trust region algorithm using fully linear models proposed by Conn et al. [25]. Section III discusses the approach of Wild et al. to build a fully linear model using RBF functions, and presents our extension to the case of multifidelity model calibration. Section IV provides an overview of the computational implementation of the method and suggests a way to incorporate the method of generating fully linear models from Wild and Shoemaker [28] with flexible Bayesian model calibration techniques. Section V demonstrates the multifidelity optimization algorithm on an analytical example and a supersonic airfoil design problem. Section VI then develops the extension of our approach to the case when there are multiple lower-fidelity models. Finally, Sec. VII concludes the paper.

II. Trust-Region-Based Multifidelity Optimization

We consider a setting where we have two (or more) models that represent the physical system of interest: a high-fidelity function that accurately estimates system metrics of interest but is expensive to evaluate, and a low-fidelity function with lower accuracy but cheaper evaluation cost. We define our high-fidelity function as $f_{\text{high}}(\mathbf{x})$ and our low-fidelity function as $f_{\text{low}}(\mathbf{x})$, where $\mathbf{x} \in \mathbb{R}^n$ is the vector of n design variables. Our goal is to solve the unconstrained optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} f_{\text{high}}(\mathbf{x}) \tag{1}$$

using information from evaluations of $f_{\text{low}}(\mathbf{x})$ to reduce the required number of evaluations of $f_{\text{high}}(\mathbf{x})$.

We use the derivative-free trust region algorithm of Conn et al. [25] to solve Eq. (1). From an initial design vector $\mathbf{x}_0$, the trust region method generates a sequence of design vectors that each reduce the high-fidelity function value, where we denote $\mathbf{x}_k$ to be this design vector on the kth trust region iteration. Following the general

Bayesian calibration approach in [17], we define $e_k(\mathbf{x})$ to be a model of the error between the high- and low-fidelity functions on the kth trust region iteration, and we construct a surrogate model $m_k(\mathbf{x})$ for $f_{\text{high}}(\mathbf{x})$ as

$$m_k(\mathbf{x}) = f_{\text{low}}(\mathbf{x}) + e_k(\mathbf{x}) \tag{2}$$

We define the trust region at iteration k, $\mathcal{B}_k$, to be the region centered at $\mathbf{x}_k$ with size Δ_k

$$\mathcal{B}_k = \{\mathbf{x}\colon \|\mathbf{x} - \mathbf{x}_k\| \leq \Delta_k\} \tag{3}$$

where any norm can be used, provided there exist constants c_1 and c_2 such that $\|\cdot\|_2 \leq c_1\|\cdot\|$ and $\|\cdot\| \leq c_2\|\cdot\|_2$. The trust region is a portion of the design space where the surrogate model is considered an accurate representation of the high-fidelity function. The accuracy is ensured with a local calibration, in this case updating the error model, $e_k(\mathbf{x})$. To improve the high-fidelity function value, the surrogate model is minimized, or approximately minimized, within the trust region. The size of the trust region is updated dynamically based on the quality of the surrogate model estimate of the high-fidelity function value; good estimates cause the size of the trust region to increase and poor estimates cause the size of the trust region to decrease. Therefore, the trust region serves as both a means to control the size of the allowable optimization step and as a means to control the accuracy of the surrogate model predictions. For a detailed discussion of trust region algorithms see [22] or [29].

If the high-fidelity function $f_{\text{high}}(\mathbf{x})$ and the surrogate models $m_k(\mathbf{x})$ satisfy certain conditions, this framework provides a guarantee of convergence to a stationary point of the high-fidelity function $f_{\text{high}}(\mathbf{x})$. Specifically, the convergence proof requires that the high-fidelity function $f_{\text{high}}(\mathbf{x})$ be 1) continuously differentiable, 2) have a Lipschitz continuous derivative, and 3) be bounded from below within a region of a relaxed level-set $\mathcal{L}(\mathbf{x}_0)$, defined as

$$L(\mathbf{x}_0) = \{\mathbf{x} \in \mathbb{R}^n\colon f_{\text{high}}(\mathbf{x}) \leq f_{\text{high}}(\mathbf{x}_0)\} \tag{4}$$

$$B(\mathbf{x}_k) = \{\mathbf{x} \in \mathbb{R}^n\colon \|\mathbf{x} - \mathbf{x}_k\| \leq \Delta_{\max}\} \tag{5}$$

$$\mathcal{L}(\mathbf{x}_0) = L(\mathbf{x}_0) \bigcup_{\mathbf{x_k} \in L(\mathbf{x}_0)} B(\mathbf{x}_k) \tag{6}$$

where $\Delta_{\max}$ is the maximum allowable trust region size. The relaxed level-set is required because the trust region algorithm may attempt to evaluate the high-fidelity function at points outside of the level set at $\mathbf{x}_0$. The convergence proof further requires that the surrogate models $m_k(\mathbf{x})$ are *fully linear*, where the following definition of a fully linear model is from Conn et al. [25]:

Definition 1 *Let a function* $f_{\mathbf{high}}(\mathbf{x})\colon \mathbb{R}^n \to \mathbb{R}$ *that satisfies the conditions 1–3 above, be given. A set of model functions* $\boldsymbol{\mathcal{M}} = \{m\colon \mathbb{R}^n \to \mathbb{R}, m \in C^1\}$ *is called a fully linear class of models if the following occur:*

There exist positive constants κ_f, κ_g *and* κ_{blg} *such that for any* $\mathbf{x} \in L(\mathbf{x}_0)$ *and* $\Delta_k \in (0, \Delta_{\max}]$ *there exists a model function* $m_k(\mathbf{x})$ *in* $\mathcal{M}$ *with Lipschitz continuous gradient and corresponding Lipschitz constant bounded by* k_{blg}, *and such that the error between the gradient of the model and the gradient of the function satisfies*

$$\|\nabla f_{\text{high}}(\mathbf{x}) - \nabla m_k(\mathbf{x})\| \leq \kappa_g \Delta_k \quad \forall\ \mathbf{x} \in \mathcal{B}_k \tag{7}$$

and the error between the model and the function satisfies

$$|f_{\text{high}}(\mathbf{x}) - m_k(\mathbf{x})| \leq \kappa_f \Delta_k^2 \quad \forall\ \mathbf{x} \in \mathcal{B}_k \tag{8}$$

Such a model $m_k(\mathbf{x})$ *is called fully linear on* $\mathcal{B}_k$ [25].

At iteration k, the trust region algorithm solves the subproblem

$$\min_{\mathbf{s}_k} m_k(\mathbf{x}_k + \mathbf{s}_k) \qquad \text{s.t.} \|\mathbf{s}_k\| \leq \Delta_k \tag{9}$$

to determine the trust region step $\mathbf{s}_k$. If the subproblem is not solved exactly, the minimum requirement is that the steps found in the trust region subproblem must satisfy a sufficient decrease condition. At iteration k, we require that the model $m_k(\mathbf{x})$ have a finite upper bound on the norm of its Hessian matrix evaluated at $\mathbf{x}_k$: $\|H_k(\mathbf{x}_k)\| \leq \kappa_{bhm} < \infty$. This bound on the Hessian may be viewed as a bound on the Lipschitz constant of the gradient of $m_k(\mathbf{x}_k)$ [25]. The sufficient decrease condition requires the step to satisfy the fraction of Cauchy decrease. As given in [25,27], this requires that for some constant, $\kappa_{\text{FCD}} \in (0, 1)$, the step $\mathbf{s}_k$ satisfies

$$m_k(\mathbf{x}_k) - m_k(\mathbf{x}_k + \mathbf{s}_k) \geq \frac{\kappa_{\text{FCD}}}{2} \|\nabla m_k(\mathbf{x}_k)\| \min\left[\frac{\|\nabla m_k(\mathbf{x}_k)\|}{\kappa_{bhm}}, \Delta_k\right] \tag{10}$$

The high-fidelity function f_{high} is then evaluated at the new point, $\mathbf{x}_k + \mathbf{s}_k$. We compare the actual improvement in the function value with the improvement predicted by the model by defining

$$\rho_k = \frac{f_{\text{high}}(\mathbf{x}_k) - f_{\text{high}}(\mathbf{x}_k + \mathbf{s}_k)}{m_k(\mathbf{x}_k) - m_k(\mathbf{x}_k + \mathbf{s}_k)} \tag{11}$$

The trial point is accepted or rejected according to

$$\mathbf{x}_{k+1} = \begin{cases} \mathbf{x}_k + \mathbf{s}_k & \text{if } \rho_k > 0 \\ \mathbf{x}_k & \text{otherwise} \end{cases} \tag{12}$$

If the step is accepted, then the trust region is updated to be centered on the new iterate $\mathbf{x}_{k+1}$. The size of the trust region Δ_k must now be updated based on the quality of the surrogate model prediction. The size of the trust region is increased if the surrogate model predicts the change in the function value well and the trust region is contracted if the model predicts the function change poorly. Specifically, we update the trust region size using

$$\Delta_{k+1} = \begin{cases} \min\{\gamma_1 \Delta_k, \Delta_{\max}\} & \text{if } \rho_k \geq \eta \\ \gamma_0 \Delta_k & \text{if } \rho_k < \eta \end{cases} \tag{13}$$

where $0 < \eta < 1$, $0 < \gamma_0 < 1$, and $\gamma_1 > 1$.

A new fully linear model $m_{k+1}(\mathbf{x})$ is then built using the radial basis function interpolation approach described in the next section. That surrogate model will be fully linear on a region $\mathcal{B}_{k+1}$ having center $\mathbf{x}_{k+1}$ and size Δ_{k+1}.

To check for algorithm termination, the gradient of the model is computed at $\mathbf{x}_{k+1}$. If $\|\nabla m_{k+1}(\mathbf{x}_{k+1})\| > \epsilon$ for a small ϵ, the trust region algorithm will continue to iterate, solving the next subproblem on the new trust region, $\mathcal{B}_{k+1}$, with the updated model, $m_{k+1}(\mathbf{x})$. However, if $\|\nabla m_{k+1}(\mathbf{x}_{k+1})\| \leq \epsilon$, we need to confirm that the algorithm has reached a stationary point of $f_{\text{high}}(\mathbf{x})$. If gradients of the high-fidelity function are available, one could evaluate if $\|\nabla f_{\text{high}}(\mathbf{x}_{k+1})\| \leq \epsilon$ directly. In the general derivative-free case, we use the condition in Eq. (7), and show that if $\Delta_{k+1} \to 0$ then $\|\nabla f_{\text{high}}(\mathbf{x}_{k+1}) - \nabla m_{k+1}(\mathbf{x}_{k+1})\| \to 0$. In practice we achieve this by updating the model to be fully linear on a trust region with size some fraction, $0 < \alpha < 1$, of Δ_{k+1}. This process continues until either $\|\nabla m_{k+1}(\mathbf{x}_{k+1})\| > \epsilon$, in which case the trust region algorithm will continue with the updated model and updated Δ_{k+1}, or $\Delta_{k+1} \leq \epsilon_2$, for a small ϵ_2, which terminates the algorithm. This process of checking for convergence is referred to as the criticality check in Conn et al. [25].

III. Interpolation-Based Multifidelity Models

In this section we discuss a method of creating surrogate models that satisfy the conditions for provable convergence presented in Sec. II. This section first presents an overview of the radial basis function (RBF) interpolation approach of Wild et al. [26], where the interpolation points are chosen so that the resulting model is fully linear. Next, we present an extension of this approach to the case of multifidelity models.

Define $\mathbf{d}_j$ to be the jth point in the set of designs at which the high-fidelity and low-fidelity functions have been sampled. Define $\mathbf{y}_i$ to be the vector from the current iterate (i.e., center of the current trust region), $\mathbf{x}_k$, to any sample point inside or within the vicinity of the current trust region, $\mathbf{d}_i$, that is selected to be an interpolation point. Also define $\mathcal{Y}$ to be the set of the zero vector (i.e., $\mathbf{y}=\mathbf{0}$ corresponds to $\mathbf{x}_k$) and all of the vectors $\mathbf{y}_i$. This notation is shown graphically in Fig. 1.

The RBF interpolation is defined so that by construction the surrogate model is equal to the high-fidelity function at all interpolation points. That is, the error between the high- and low-fidelity functions is interpolated exactly for all points defined by the vectors within $\mathcal{Y}$:

$$e_k(\mathbf{x}_k+\mathbf{y}_i)=f_{\text{high}}(\mathbf{x}_k+\mathbf{y}_i)-f_{\text{low}}(\mathbf{x}_k+\mathbf{y}_i)\quad\forall\ \mathbf{y}_i\in\mathcal{Y}\tag{14}$$

The RBF interpolation has the form

$$e_k(\mathbf{x})=\sum_{i=1}^{|\mathcal{Y}|}\lambda_i\phi(\|\mathbf{x}-\mathbf{x}_k-\mathbf{y}_i\|)+\sum_{i=1}^{n+1}\nu_i\pi_i(\mathbf{x}-\mathbf{x}_k)\tag{15}$$

where ϕ is any positive definite, twice continuously differentiable RBF with $\phi'(0)=0$, and the second term in Eq. (15) represents a linear tail, where π_i denotes the ith component of the vector $\Pi(\mathbf{x}-\mathbf{x}_k)=[1\quad(\mathbf{x}-\mathbf{x}_k)]^T$. The coefficients λ_i and ν_i represent the RBF interpolation, and are found by the QR-factorization technique of Wild et al. [26]. For the model to be fully linear, the RBF coefficients λ_i and ν_i must be bounded in magnitude. This is achieved by using the interpolation point selection method in Wild et al. The process can be summarized as follows. First, the existing high-fidelity sample points $\mathbf{d}_j$ in the vicinity of the trust region are tested to see if there are $n+1$ affinely independent vectors. (This test is carried out using singular value decomposition of a matrix containing as columns the vectors $\mathbf{y}_j$ and is computationally inexpensive compared with the typical cost of a high-fidelity solve.) If fewer than $n+1$ affinely independent points are found, additional high-fidelity function evaluations are required to generate additional interpolation points. Second, we test all other points $\mathbf{d}_j$ at which the high-fidelity function value is known, by measuring the impact of their addition as interpolation points on the RBF coefficients λ_i and ν_i. Those points that ensure the RBF coefficients remain bounded are used as additional interpolation points to update the model. Wild proved that this RBF interpolation model construction algorithm produces a fully linear model for a function satisfying conditions 1 and 2 above [27]. An illustration of the calibrated models resulting from this process is presented as Fig. 2.

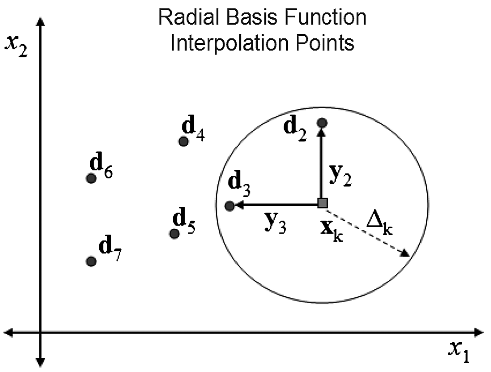


Fig. 1 Graphical representation of the notation used to define points and vectors in and around the trust region.

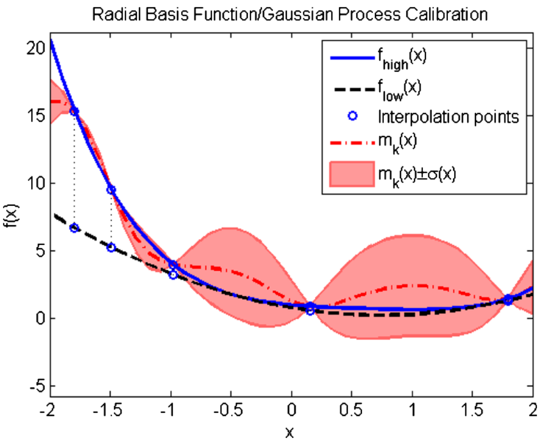


Fig. 2 An illustration of the surrogate model of a high-fidelity function created using a radial basis function interpolation of the error between a high- and low-fidelity function. The high-fidelity function is in blue, the low-fidelity function is in black, the fully linear surrogate model is in red, the uncertainty estimate of the surrogate model is in pink, and the calibration points are circled.

For Wild's interpolation approach to be applicable in our Bayesian calibration setting, we require that the error function defined by $f_{\text{high}}(\mathbf{x})-f_{\text{low}}(\mathbf{x})$ satisfies conditions 1 and 2 above. Condition 1, that the function is continuously differentiable, is satisfied if both $f_{\text{high}}(\mathbf{x})$ and $f_{\text{low}}(\mathbf{x})$ are continuously differentiable. To establish condition 2, that the derivative of $f_{\text{high}}(\mathbf{x})-f_{\text{low}}(\mathbf{x})$ is Lipschitz continuous, we require that both $\nabla f_{\text{high}}(\mathbf{x})$ and $\nabla f_{\text{low}}(\mathbf{x})$ be Lipschitz continuous in the relaxed level set defined in Eq. (6). For the high-fidelity function we require

$$\frac{\|\nabla f_{\text{high}}(\mathbf{x}_1)-\nabla f_{\text{high}}(\mathbf{x}_2)\|}{\|\mathbf{x}_1-\mathbf{x}_2\|}\leq\kappa_{\text{high}}\quad\forall\ \mathbf{x}_1,\mathbf{x}_2\in\mathcal{L}(\mathbf{x}_0)\tag{16}$$

and for the low-fidelity function

$$\frac{\|\nabla f_{\text{low}}(\mathbf{x}_1)-\nabla f_{\text{low}}(\mathbf{x}_2)\|}{\|\mathbf{x}_1-\mathbf{x}_2\|}\leq\kappa_{\text{low}}\quad\forall\ \mathbf{x}_1,\mathbf{x}_2\in\mathcal{L}(\mathbf{x}_0)\tag{17}$$

with Lipschitz constants κ_{high} and κ_{low}, respectively. Therefore, by the triangle inequality, we obtain

$$\frac{\|[\nabla f_{\text{high}}(\mathbf{x}_1)-\nabla f_{\text{low}}(\mathbf{x}_1)]-[\nabla f_{\text{high}}(\mathbf{x}_2)-\nabla f_{\text{low}}(\mathbf{x}_2)]\|}{\|\mathbf{x}_1-\mathbf{x}_2\|}\leq\kappa_{\text{high}}+\kappa_{\text{low}}\quad\forall\ \mathbf{x}_1,\mathbf{x}_2\in\mathcal{L}(\mathbf{x}_0)\tag{18}$$

where the Lipschitz constant of the difference is bounded by $\kappa_{\text{high}}+\kappa_{\text{low}}$. Accordingly, the convergence proof for the trust region algorithm used in Conn et al. [25] holds.

IV. Numerical Implementation of Algorithms

This section represents an overview of the numerical implementation of the multifidelity optimization algorithm and suggests a manner in which the method of Wild and Shoemaker [28] to generate fully linear models can be used in a flexible Bayesian calibration setting. The first subsection, Sec. IV.A, implements the trust region-based optimization algorithm presented in Sec. II. Whenever creation of a new fully linear model is needed, the method discussed in Sec. III is implemented using the algorithm presented in Sec. IV.B.

A. Trust Region Implementation

Algorithm 1 provides an overview of the numerical implementation of the trust region optimization method presented in Sec. II. For each trust region iteration, the algorithm guarantees that a step is found that satisfies the fraction of Cauchy decrease, Eq. (10). The algorithm only samples the high-fidelity function when necessary for convergence, and it stores all high-fidelity function evaluations in a database so that design points are never reevaluated. Whenever an updated surrogate model is needed, the model generation method described in the following subsection creates a surrogate model using this database of high-fidelity function evaluations together with new high-fidelity evaluations when necessary. The parameters of the trust region optimization algorithm were defined in Sec. II, while recommended values and sensitivity of results to those values will be presented in Sec. V.

Algorithm 1 Trust-Region Algorithm for Iteration k

```
1: Compute a step, $\mathbf{s}_k$, that satisfies the fraction of Cauchy decrease
     requirement, Eq. (10), for the trust region subproblem, by solving
       $\min_{\mathbf{s}_k} m_k(\mathbf{x}_k + \mathbf{s}_k) \qquad \text{s.t. } \|\mathbf{s}_k\| \leq \Delta_k$
2: If $f_{\text{high}}(\mathbf{x}_k + \mathbf{s}_k)$ has not been evaluated previously, evaluate
     the high-fidelity function at that point.
     2a: Store $f_{\text{high}}(\mathbf{x}_k + \mathbf{s}_k)$ in database.
3: Compute the ratio of actual improvement to predicted improvement
       $\rho_k = \dfrac{f_{\text{high}}(\mathbf{x}_k) - f_{\text{high}}(\mathbf{x}_k + \mathbf{s}_k)}{m_k(\mathbf{x}_k) - m_k(\mathbf{x}_k + \mathbf{s}_k)}$
4: Accept or reject the trial point according to $\rho_k$
       $\mathbf{x}_{k+1} = \begin{cases} \mathbf{x}_k + \mathbf{s}_k & \text{if } \rho_k > 0 \\ \mathbf{x}_k & \text{otherwise} \end{cases}$
5: Update the trust region size according to $\rho_k$
       $\Delta_{k+1} = \begin{cases} \min\{\gamma_1 \Delta_k, \Delta_{\max}\} & \text{if } \rho_k \geq \eta \\ \gamma_0 \Delta_k & \text{if } \rho_k < \eta \end{cases}$
6: Create a new model $m_{k+1}(\mathbf{x})$ that is fully linear on
     $\{\mathbf{x}\colon \|\mathbf{x} - \mathbf{x}_{k+1}\| \leq \Delta_{k+1}\}$ using Algorithm 2.
7: Check for convergence: if $\|\nabla m_{k+1}(\mathbf{x}_{k+1})\| > \epsilon$, algorithm has not
     converged—go to step 1. Otherwise,
     7a: While $\|\nabla m_{k+1}(\mathbf{x}_{k+1})\| \leq \epsilon$ and $\Delta_{k+1} > \epsilon_2$,
     7b: Reduce the trust region size, $\alpha \Delta_{k+1} \to \Delta_{k+1}$.
     7c: Update model $m_{k+1}(\mathbf{x})$ to be fully linear on
         $\{\mathbf{x}\colon \|\mathbf{x} - \mathbf{x}_{k+1}\| \leq \Delta_{k+1}\}$ using Algorithm 2.
```

Algorithm 2 Create Fully Linear Models Allowing Maximum Likelihood Correlation Lengths

```
1: Compute the likelihood for all RBF correlation lengths,
     $\xi_i \in \{\xi_1, \ldots, \xi_n\}$ with steps 2–5.
2: Generate a set of $n + 1$ affinely independent points in the vicinity of
     the trust region:
     2a: Set $\mathbf{y}_1 = \mathbf{0}$, and add $\mathbf{y}_1$ to the set of calibration vectors $\mathcal{Y}$.
     2b: Randomly select any high-fidelity sample point, $\mathbf{d}_2$, within the
         current trust region and add the vector $\mathbf{y}_2 = \mathbf{d}_2 - \mathbf{x}_k$ to $\mathcal{Y}$.
     2c: For all unused high-fidelity sample points within the current
         trust region, add the vector $\mathbf{y} = \mathbf{d}_j - \mathbf{x}_k$ to $\mathcal{Y}$ if the projection of $\mathbf{y}$
         onto the nullspace of the span of the vectors in the current $\mathcal{Y}$ is
         greater than $\theta_1 \Delta_k$, $0 < \theta_1 < 1$.
     2d: If fewer than $n + 1$ vectors are in calibration set, repeat step 2c
         allowing a larger search region of size $\theta_3 \Delta_k$, $\theta_3 > 1$.
     2e: While fewer than $n + 1$ vectors are in $\mathcal{Y}$,
     2f: Evaluate the high-fidelity function at a point within the nullspace
         of the span of the vectors in $\mathcal{Y}$ and add $\mathbf{y} = \mathbf{d} - \mathbf{x}_k$ to $\mathcal{Y}$.
     2g: Store the results of all high-fidelity function evaluations in the
         database.
3: Consider the remaining unused high-fidelity sample points within a
     region centered at the current iterate with size $\theta_4 \Delta_k$, $\theta_4 > 1$. Add
     points so that the total number of interpolation points does not
     exceed $p_{\max}$, the RBF coefficients remain bounded, and the
     surrogate model is fully linear (using, for example, the AddPoints
     algorithm of Wild et al. [28]).
4: Compute the RBF coefficients using the QR factorization technique of
     Wild et al. [26].
5: If only $n + 1$ vectors are in the calibration set, $\mathcal{Y}$, assign the likelihood
     of the current correlation length, $\xi_i$, to $-\infty$. Otherwise compute the
     likelihood of the RBF interpolation using standard methods [30,31].
6: Select the $\xi_i$ with the maximum likelihood.
     6a: If the maximum likelihood is $-\infty$ choose the largest $\xi_i$. This
         model typically occurs if exactly $n + 1$ points are in the
         neighborhood of the trust region and corresponds to a linear
         regression of the high-fidelity function at the calibration points
         included in $\mathcal{Y}$, but it still satisfies conditions for convergence.
7: Return the set of calibration vectors $\mathcal{Y}$, RBF coefficients, and updated
     database of high-fidelity function evaluations.
```

B. Fully Linear Bayesian Calibration Models

Algorithm 2 presents the numerical implementation of the method to generate fully linear surrogate models, allowing for a Bayesian maximum likelihood estimate of the RBF correlation length. The RBF models used in Bayesian model calibration have a length scale parameter that provides flexibility. For instance, in the Gaussian RBF model $\phi(r) = e^{-r^2/\xi^2}$, the parameter ξ is a variable length scale that can alter the shape of the correlation structure. If the interpolation errors are assumed to have a Gaussian distribution, then a maximum likelihood estimate can be used to estimate the value of ξ that best represents the data [30,31]. Therefore, our process to generate a fully linear surrogate model uses the method of Wild et al. [26] on a set of candidate length scales, $\xi_i \in \{\xi_1, \ldots, \xi_n\}$. A fully linear model is constructed for each candidate length scale, and the likelihood of each length scale is computed. The trust region algorithm then uses the surrogate model constructed with ξ^*, where ξ^* is chosen as the value of ξ corresponding to the maximum likelihood. This maximum likelihood approach can improve the model calibration, and also provides flexibility in selecting sample points in the extension to the case when there are multiple lower-fidelity models (as will be discussed in Sec. VI).

V. Multifidelity Optimization Examples

This section demonstrates the multifidelity optimization scheme for two examples. The first is an analytical example considering the Rosenbrock function and the second is a supersonic airfoil design problem.

A. Rosenbrock Function

The first example multifidelity optimization example is the Rosenbrock function

$$\min_{\mathbf{x} \in \mathbb{R}^2} f_{\text{high}}(\mathbf{x}) = (x_2 - x_1^2)^2 + (1 - x_1)^2 \tag{19}$$

The minimum of the Rosenbrock function is at $x^* = (1, 1)$ and $f(x^*) = 0$. For this simple analytical function with only two design variables, we do not expect the multifidelity method to significantly outperform a quasi-Newton method, however, the example is useful to illustrate the multifidelity approach and to demonstrate effects of algorithm parameters. Table 1 presents the number of high-fidelity function evaluations required to optimize the Rosenbrock function using a variety of low-fidelity functions. All of the low-fidelity functions have a different minimum than the Rosenbrock function, with the exception of the case when the low-fidelity function is set equal to the Rosenbrock function, corresponding to a perfect low-fidelity function. For all of the examples in this section the optimization parameters used are given in Table 2 and are discussed in the remainder of this subsection.

We use a Gaussian RBF, $\phi(r) = e^{-r^2/\xi^2}$, to build the RBF error interpolation and two methods of selecting the spatial correlation length, ξ. The first method is to fix a value of ξ, and the second approach is based on kriging methods, which assume interpolation errors are normally distributed and maximize the likelihood that the RBF surface predicts the function [30,31]. To save computation time, the maximum likelihood correlation length is estimated by examining 10 correlation lengths between 0.1 and 5.1, and the correlation length that has the maximum likelihood is chosen. If all correlation lengths have the same likelihood, the maximum correlation length is used. The results in Table 1 show that the correlation length has a moderate impact on the convergence rate of the method. For this problem, using either $\xi = 2$ or ξ^*, the correlation length that maximizes the likelihood at each trust region iteration, leads to the best result.

Table 1 demonstrates that the quality of the low-fidelity function can significantly impact the number of required high-fidelity function evaluations. As a baseline, the average number of function calls for a quasi-Newton method [32] directly optimizing the Rosenbrock function is 69 and for the global optimization method, DIRECT [33], is 565. The Bayesian calibration approach uses between 5 and 250 high-fidelity function evaluations depending on the quality of the low-fidelity model. The worst case, 250 high-fidelity function

Table 1 Table of average number of function evaluations required to minimize the Rosenbrock function, Eq. (19), from a random initial point on $x_1, x_2 \in [-5, 5]$. Results for a selection of Gaussian radial bases function spatial parameters, ξ, are shown. ξ^* corresponds to optimizing the spatial parameter according to a maximum likelihood criteria [30]. Also included are the number of function evaluations required using Efficient Global Optimization (EGO) and first-order consistent trust region methods with a multiplicative correction and an additive correction. The gradient-free calibration uses only additive corrections. For a standard quasi-Newton method the average number of function evaluations is 69 and using DIRECT requires 565 function evaluations

	Gradient-Free (calibration)						Gradient-Based	
Low-Fidelity function	$\xi = 1$	$\xi = 2$	$\xi = 3$	$\xi = 5$	ξ^*	EGO	Add-Corr.	Mult.-Corr.
$f_{\text{low}}(\mathbf{x}) = 0$	148	107	177	223	178	27	503	289[a]
$f_{\text{low}}(\mathbf{x}) = x_1^2 + x_2^2$	129	77	106	203	76	28	401	312
$f_{\text{low}}(\mathbf{x}) = x_1^4 + x_2^2$	74	74	73	87	65	27	289	171
$f_{\text{low}}(\mathbf{x}) = f_{\text{high}}(\mathbf{x})$	5	5	5	5	7	fail	6	7
$f_{\text{low}}(\mathbf{x}) = -x_1^2 - x_2^2$	195	130	132	250	100	27	fail	352

[a] Indicates $f_{\text{low}}(\mathbf{x}) = 1$ had to be used.

Table 2 Optimization parameters used in the Rosenbrock function demonstration

Parameter	Description	Value
$\phi(r)$	RBF correlation	e^{-r^2/ξ^2}
ξ	RBF spatial correlation length	See Table 1
Δ_0	Initial trust region size	$\max[10, \|\mathbf{x}_0\|_\infty]$
$\Delta_{\max}$	Maximum trust region size	$10^3\Delta_0$
ϵ, ϵ_2	Termination tolerances	5×10^{-4}
γ_0	Trust region contraction ratio	0.5
γ_1	Trust region expansion ratio	2
η	Trust region expansion criterion	0.2
α	Trust region contraction ratio used in convergence check	0.9
κ_{FCD}	Fraction of Cauchy decrease requirement	10^{-4}
$p_{\max}$	Maximum number of calibration points	50
θ_1	Minimum projection into nullspace of calibration vectors	10^{-3}
θ_2	RBF coefficient conditioning parameter	10^{-4}
θ_3	Expanded trust region size to find basis, $\theta_3\Delta_k$	10
θ_4	Maximum calibration region size, $\theta_4\Delta_k$	10
δx	Finite difference step size	10^{-6}

evaluations, corresponds to having a poor low-fidelity model, one with opposite trends in most of the design space. The best case, 5 high-fidelity evaluations, corresponds to the case when the low-fidelity function exactly models the high-fidelity function. With a rather good low-fidelity function, for example a fourth degree polynomial, the multifidelity method performs similarly to the quasi-Newton method. Clearly the performance of this method compared with conventional optimization methods depends considerably on the quality of the low-fidelity function used. Results for two first-order consistent multifidelity trust region methods and multifidelity EGO are presented in Table 1 along with the results of the Bayesian calibration method. The multifidelity EGO method creates a kriging model of the error between the high- and low-fidelity functions and maximizes the expected improvement [9] of the low-fidelity function plus the kriging model to select additional high-fidelity evaluations and to update the kriging model. The kriging model is initialized using a Latin hypercube sampling of the design space with four function evaluations. The first of the gradient-based multifidelity methods creates a trust region surrogate by multiplying the low-fidelity function by a correction such that the surrogate matches both the high-fidelity function value and gradient at the current iterate [5]. The second creates the trust region surrogate by adding a linear function with the error between the high- and low-fidelity function value and gradient at the current iterate to the low-fidelity function. Though an analytical gradient is available for this simple case, when comparing gradient-based multifidelity methods to gradient-free multifidelity methods, we estimate the high-fidelity gradient at each $\mathbf{x}_k$ using a finite difference approximation (requiring n additional high-fidelity function evaluations). This gradient estimate is inexact; however, it is necessary when using a gradient-based optimization method for a function that does not have an available derivative.

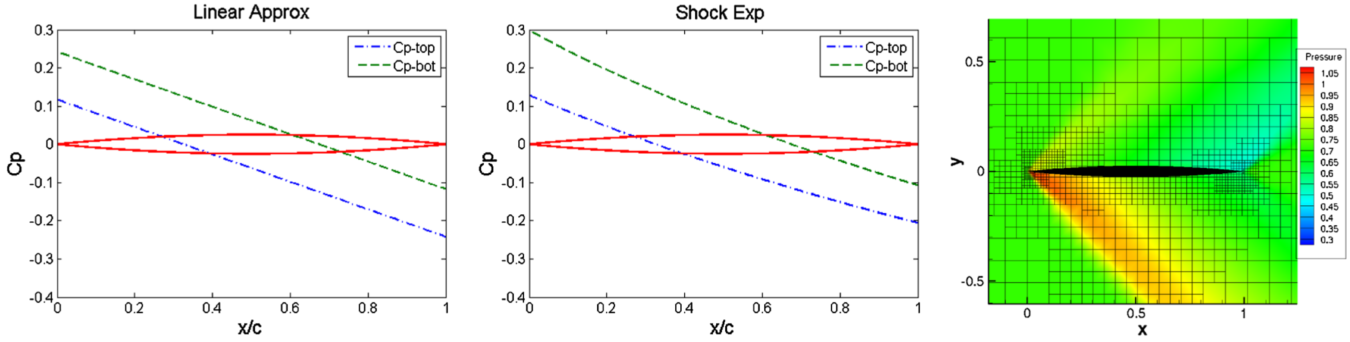
The general result for this test problem is that the Bayesian calibration approach uses fewer high-fidelity function evaluations than the first-order consistent trust region approaches of Alexandrov et al. [5] but more than multifidelity EGO.

For this simple high-fidelity function, the first-order consistent trust region methods and the quasi-Newton method require less than half the wall-clock time that the Bayesian calibration method requires (EGO takes roughly twice as long as the Bayesian calibration method). Building the RBF models requires multiple matrix inversions, each of which requires $\mathcal{O}(p_{\max}(p_{\max} + n + 1)^3)$ operations, where n is the number of design variables and $p_{\max}$ is the user-set maximum number of calibration points allowed in a model. Accordingly, the Bayesian calibration method is only recommended for high-fidelity functions that are expensive compared with the cost of repeatedly solving for RBF coefficients, which is the case of interest in this paper.

As with any optimization algorithm, tuning parameters can affect performance significantly; however, the best choices for these tuning parameters can be highly problem dependent. A sensitivity study measured the impact of algorithm parameters on the number of high-fidelity function evaluations for the Rosenbrock example using $f_{\text{low}}(\mathbf{x}) = x_1^2 + x_2^2$ as the low-fidelity function. For all of these tests, one parameter is varied and the remainder are all set to the values in Table 2. The conclusions drawn are based on the average of at least ten runs with random initial conditions on the interval $x_1, x_2 \in [-5, 5]$. While these conclusions may provide general useful guidance for setting algorithm parameters, similar sensitivity studies are recommended for application to other problems.

The parameter η is the trust region expansion criterion, where the trust region expands if $\rho_k \geq \eta$ and contracts otherwise. The sensitivity results show that lower values of η have the fewest high-fidelity function calls, and any value $0 \leq \eta \leq 0.2$ performs well. For the trust region expansion ratio γ_1 the best results are at $\gamma_1 \approx 2$, and high-fidelity function evaluations increase substantially for other values. Similarly, for the contraction ratio γ_0 the best results are observed at $\gamma_0 \approx 0.5$, with a large increase in high-fidelity function evaluations otherwise. For the fraction of Cauchy decrease κ_{FCD} the results show the number of high-fidelity evaluations is fairly insensitive to any value $0 < \kappa_{\text{FCD}} < 10^{-2}$. Similarly, for the trust region contraction ratio used in the algorithm convergence check α the number of high-fidelity function evaluations is insensitive to any value $0.5 < \alpha < 0.95$.

The method of Wild et al. [26,28] to generate fully linear models requires four tuning parameters, θ_1, θ_2, θ_3, and θ_4. The parameter θ_1 $(0 < \theta_1 < 1)$ determines the acceptable points when finding the affinely independent basis in the vicinity of the trust region in Algorithm 2. As θ_1 increases, the calibration points added to the basis must have a larger projection onto the nullspace of the current basis, and therefore fewer points are admitted to the basis. We find for the Rosenbrock example that the fewest function evaluations occurs with $\theta_1 \approx 10^{-3}$; however, for any value of θ_1 within 2 orders of magnitude of this value, the number of function evaluations increases by less than 50%. The second parameter θ_2 $(0 < \theta_2 < 1)$ is used in the AddPoints algorithm of Wild and Shoemaker [28] to ensure that the RBF coefficients remain bounded when adding additional



a) Panel method **b) Shock-expansion method** **c) Cart3D**

Fig. 3 Supersonic airfoil model comparisons at Mach 1.5 and 2 deg angle of attack.

calibration points. The number of allowable calibration points increases as θ_2 decreases to zero; however, the matrix used to compute the RBF coefficients also becomes more ill-conditioned. For our problem, we find that $\theta_2 \approx 10^{-4}$ enables a large number of calibration points while providing acceptable matrix conditioning. The two other parameters, θ_3 and θ_4, used in the calibration point selection algorithm, are significant to the algorithm's performance. The parameter θ_3 ($\theta_3 > 1$) is used if $n + 1$ affinely independent previous high-fidelity sample points do not exist within the current trust region. If fewer than $n + 1$ points are found, the calibration algorithm allows a search region of increased size $\{\mathbf{x}\colon \|\mathbf{x} - \mathbf{x}_k\| \leq \theta_3\Delta_k\}$ to find $n + 1$ affinely independent points before evaluating the high-fidelity function in additional locations. The results show that the number of function calls is insensitive to θ_3 for $1 < \theta_3 \leq 10$, with $\theta_3 \approx 3$ yielding the best results. The parameter θ_4 ($\theta_4 > 1$) represents the balance between global and local model calibration, as it determines how far points can be from the current iterate, $\mathbf{x}_k$, and still be included in the RBF interpolation. Points that lie within a region $\{\mathbf{x}\colon \|\mathbf{x} - \mathbf{x}_k\| \leq \theta_4\Delta_k\}$ are all candidates to be added to the interpolation. Calibration points outside of the trust region will affect the shape of the model within the trust region, but the solution to the subproblem must lie within the current trust region. The results of our analysis show that $\theta_4 \approx 10$ offers the best performance, with the number of high-fidelity function calls increasing substantially if $\theta_4 < 5$ or $\theta_4 > 15$. We also note that the parameter values suggested in this section are similar to the values recommended by Wild et al. [26].

B. Supersonic Airfoil Optimization

As an engineering example, a supersonic airfoil is optimized for minimum drag at Mach 1.5. Three analysis tools are available: a supersonic linear panel method, a shock-expansion theory panel method, and a computational fluid dynamics model Cart3D [34]. The linear panel method and shock-expansion theory method only compute flow quantities on the surface of the airfoil. The panel method assumes that all changes in geometry are small, so that the local surface pressure varies linearly with the slope changes in the airfoil surface. Shock-Expansion theory computes the change in the local surface pressure by solving nonlinear equations associated with compression and expansion waves related to the surface slope changes. Cart3D uses a finite volume method to approximate the Euler equations and solve for the flowfield in a domain around the airfoil. Cart3D uses an adjoint-based mesh refinement approach. The refinement is managed so that the error in the flow solution due to spatial approximation on the computational mesh is less than a set tolerance. (The smallest tolerance we were able to consistently achieve was about $\mathcal{O}(10^{-5})$ for drag.)

Figure 3 shows the approximate level of detail used in the models, and Table 3 compares the lift and drag estimates from each of the models for a 5% thick biconvex airfoil at Mach 1.5 and 2 deg angle of attack. The linear panel method and shock-expansion theory both require sharp leading and trailing edges on the airfoil, so the airfoils are parameterized by a set of spline points on the upper and lower surfaces and the angle of attack. The leading- and trailing-edge points of both surfaces are constrained to be coincident to maintain the sharp leading and trailing edges. Accordingly, an airfoil with 11 variables is parameterized by the angle of attack, and has seven spline points on the upper and lower surfaces, but only five points on each surface can be varied.

To demonstrate the RBF calibration approach to optimization, the linear supersonic panel method is used as the low-fidelity function and shock-expansion theory is used as the high-fidelity function. For supersonic flow, a zero thickness airfoil will have the minimum drag, so the airfoil must be constrained to have a thickness to chord ratio greater than 5%. This is accomplished by adding a penalty function, so that if the maximum thickness of the airfoil is less than 5%, the penalty term $1000(t/c - 0.05)^2$ is added to the drag. A similar penalty is added if the thickness anywhere on the airfoil is less than zero.

The optimization parameters used by this method are the same as in Table 2, with the exception that the RBF correlation length is either $\xi = 2$ or optimized at each iteration. A consecutive step size of less than 5×10^{-6} is an additional termination criteria for all of the multifidelity methods compared. The number of high-fidelity function evaluations required to optimize the airfoil for each of the methods using a different number of design variables is presented in Fig. 4. The airfoil optimization shows that both the first-order consistent methods and the RBF calibration method perform significantly better than the quasi-Newton method. This is largely because the multifidelity methods have a significant advantage over the single-fidelity methods in that the physics-based low-fidelity model is a reasonable representation of the high-fidelity model. However, the RBF calibration approach uses less than half the number of function evaluations as the multiplicative-correction approach. In addition, the additive correction outperforms the multiplicative correction for this problem, but the RBF calibration outperformed both. The method of maximizing the likelihood of the RBF calibration performs slightly better than just using a fixed correlation length. The results in Fig. 4 are also better than two global optimization methods, DIRECT [33] and a multifidelity formulation of EGO [9]. DIRECT requires significantly more high-fidelity evaluations than the quasi-Newton method and all of the trust region approaches. For example to get within 1% of the optimal function value with 11 design variables, DIRECT requires over 13,000 evaluations. We are unable to get the multifidelity formulation of

Table 3 Five-percent-thick biconvex airfoil results comparison at Mach 1.5 and 2 deg angle of attack. Percent difference is taken with respect to the Cart3D results. These results are typical for "well-designed" airfoils, however, in other parts of the feasible design space a discrepancy of over 33% error has been observed for drag and of at least 0.058 (47%) for lift between the panel method and shock-expansion theory

	Panel	Shock-Expansion	Cart3D
C_L	0.1244	0.1278	0.1250
Percent difference	0.46%	2.26%	0.00%
C_D	0.0164	0.0167	0.01666
Percent difference	1.56%	0.24%	0.00%

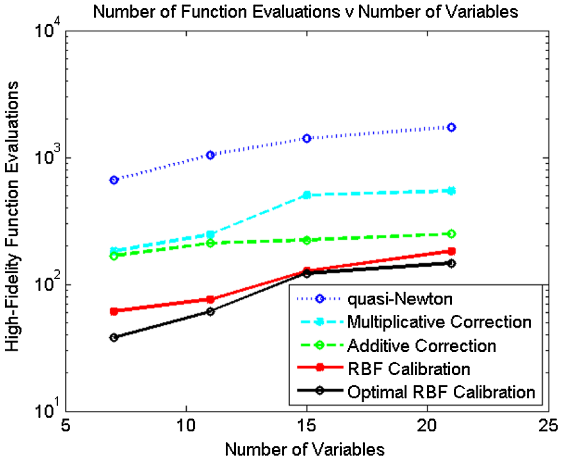


Fig. 4 Number of shock-expansion theory evaluations required to minimize the drag of a supersonic airfoil verse the number of parameters. The low-fidelity model is the supersonic panel method.

EGO to find an airfoil design having a drag coefficient within 20% of the optimal value for a variety of low-fidelity functions and a budget of 1500 high-fidelity evaluations. Many EGO implementations have been attempted, including initializing the kriging model with a Latin hypercube sample using $3n$, $4n$, and $5n$ high-fidelity evaluations and maximizing the expected improvement using DIRECT and a genetic algorithm.

As a second test case, the panel method was used as a low-fidelity function to minimize the drag of an airfoil with Cart3D as the high-fidelity function. Cart3D has an adjoint-based mesh refinement, which ensures the error caused by the discretization is less than a tolerance. Accordingly, the drag computed by Cart3D is only to within a tolerance. The drag is therefore not Lipschitz continuous due to the finite precision. In the execution of our multifidelity optimization algorithm, the gradient of the surrogate model does not go to zero. However, no progress is made and the trust region radius converges to zero. This forces the algorithm to take small steps and the combination of a small step size and small trust region is a supplemental termination criteria. On average, the airfoil parameterized with 11 variables requires 88 high-fidelity (Cart3D) function evaluations. A comparison of the minimum drag airfoils from the panel method, shock-expansion theory, and Cart3D is presented in Fig. 5. The airfoils all resemble the expected diamond shape.

To compare this gradient-free approach on a problem with multiple local minima, we change the objective to find an airfoil with a lift coefficient of 0.3 and a drag coefficient of less than 0.05, minimizing $(C_L - 0.3)^2 + \max\{C_D - 0.05, 0\}^2$, and retain the same penalty function to ensure positive thickness and at least 5% thickness to chord ratio. There are numerous airfoil designs in the design space that have the optimal function value, zero (approximately, due to the penalty function). Considering shock-expansion theory as the high-fidelity analysis and the panel method as the low-fidelity analysis, Table 4 compares the average number of high-fidelity evaluations from random initial airfoils parametrized by the angle of attack and ten surface spline points. Two low-fidelity functions are used, $(C_L - 0.3)^2 + \max\{C_D - 0.05, 0\}^2$, which is the same as the objective function except computed with the panel method instead of shock-expansion theory, and $(C_L - 0.4)^2 + \max\{C_D - 0.05, 0\}^2$ also computed by the panel method. Both of these low-fidelity objectives are augmented with the penalty function. The first low-fidelity objective has minima near the high-fidelity minima; however, the second low-fidelity objective has multiple local minima that are all separate from the high-fidelity minima. For comparison, from random initial airfoil designs a quasi-Newton method [32] requires 643 shock-expansion evaluations, and the global optimization method, DIRECT, requires over 1000 evaluations to find an objective value of 1×10^{-5}, and over 8000 evaluations for 1×10^{-8}. The results show that on average with a "good" low-fidelity model the RBF calibration approach uses the fewest high-fidelity evaluations to find a locally optimal solution. For this problem, with a "poor" low-fidelity model the RBF calibration approach finished second to the first-order consistent trust region approach using an additive correction. We also observe that the RBF calibration approach is not significantly affected by either choosing a correlation length, $\xi = 2$, or by using the maximum likelihood correlation length, $\xi = \xi^*$. For the EGO implementation, $4n$ high-fidelity evaluations are used for the initial Latin hypercube sample and a genetic algorithm is used to maximize the expected improvement.

VI. Combining Multiple Fidelity Levels

This section addresses how the radial basis function interpolation technique can be extended to optimize a function when there are multiple lower-fidelity functions. For instance, consider the case when our goal is to find the $\mathbf{x}^*$ that minimizes $f_{\text{high}}(\mathbf{x})$, and there exists two or more lower-fidelity functions, an intermediate-fidelity, $f_{\text{med}}(\mathbf{x})$, and a low-fidelity, $f_{\text{low}}(\mathbf{x})$.

The typical approach to solve this problem is to nest the lower-fidelity function; that is, to use the intermediate-fidelity function as the low-fidelity model of the high-fidelity function, and to use the lowest-fidelity function as the low-fidelity model of the intermediate-fidelity function. To do this, two calibration models are needed

$$f_{\text{high}}(\mathbf{x}) \approx f_{\text{med}}(\mathbf{x}) + e_{\text{med}}(\mathbf{x}) \tag{20}$$

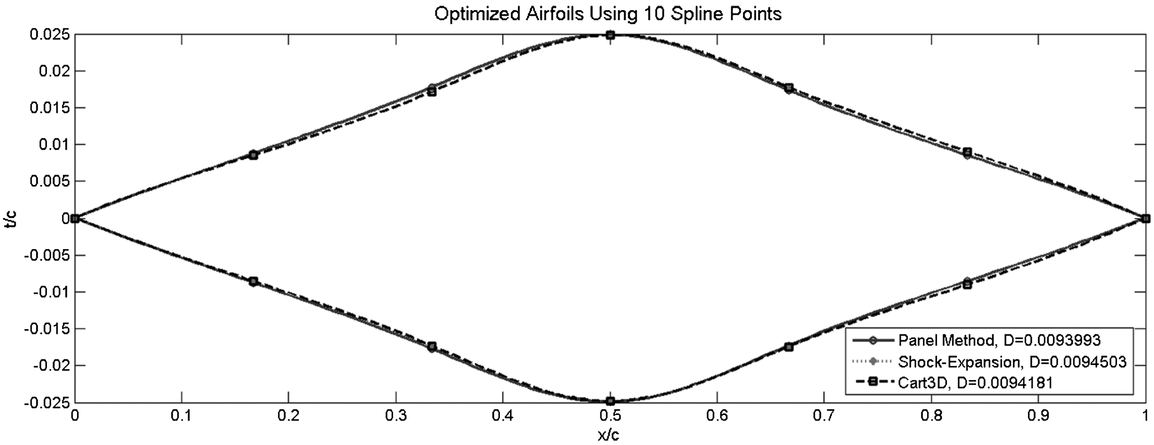


Fig. 5 Minimum drag airfoils from each of the three analysis models. The panel method airfoil is generated by solving a single-fidelity optimization problem with a quasi-Newton method. The shock-expansion method and Cart3D airfoils are generated with this multifidelity approach and spatial correlation length $\xi = 2$ with the panel method as a low-fidelity model.

Table 4 Average number of shock-expansion theory evaluations required to find an airfoil (angle of attack and ten surface spline points, 11 design parameters) with a target lift coefficient of 0.3 and maximum drag coefficient of 0.05. The low-fidelity analysis is the panel method, and two low-fidelity objective functions are used to demonstrate the influence of low-fidelity model quality. For comparison, a quasi-Newton method requires 643 evaluations and the global optimization method, DIRECT, requires 1031 evaluations to get within 1 × 10^{-5} of the optimal objective, and 8215 evaluations to get within 1 × 10^{-8} of the optimal objective

	Gradient-Free (calibration)			Gradient-Based	
Low-Fidelity function	$\xi = 2$	$\xi = \xi^*$	EGO	Add-Corr.	Mult.-Corr.
$(C_L - 0.3)^2 + \max\{C_D - 0.05, 0\}^2$	78	54	118	80	123[a]
$(C_L - 0.4)^2 + \max\{C_D - 0.05, 0\}^2$	152	147	285	83	642[a]

[a]Indicates 1 × 10^{-4} was added to the low-fidelity objective function.

$$f_{\text{med}}(\mathbf{x}) \approx f_{\text{low}}(\mathbf{x}) + e_{\text{low}}(\mathbf{x}) \tag{21}$$

In the nested approach, the high-fidelity optimization is performed on the approximate high-fidelity function, which is the medium-fidelity function plus the calibration model e_{med}. However, to determine the steps in that optimization, another optimization is performed on a lower-fidelity model. This low-fidelity optimization is performed on the model

$$m(\mathbf{x}) \approx f_{\text{low}}(\mathbf{x}) + e_{\text{low}}(\mathbf{x}) + e_{\text{med}}(\mathbf{x}) \tag{22}$$

but only the low-fidelity calibration model e_{low} is adjusted. The nested approach can quickly become computationally inefficient. To take one step in the high-fidelity space, an optimization is required on the medium-fidelity function. However, for each step in medium-fidelity space, an optimization is required on the lower-fidelity function. Even if the medium and low-fidelity models are cheap to evaluate compared with the high-fidelity model, the potential exponential scaling in the number of lower-fidelity function evaluations required between fidelity levels can counter computational gains.

An alternative to nesting multiple lower-fidelity functions is to use a maximum likelihood estimator to estimate the high-fidelity function. Since the multifidelity optimization method proposed in this paper uses radial basis function interpolants, a variance estimate of the interpolation error can be created using standard Gaussian process techniques [30,31]. This variance estimate reflects the overall level of variability in the observed data (roughly the average error between models) and the observed correlations (how smoothly the error changes in the design space). It is therefore zero at all sampled points and grows with distance away from sampled points [9]. Figure 2 presents an illustration of this. In the case of multiple fidelity levels, the lower-fidelity estimates of $f_{\text{high}}(\mathbf{x})$, for example $f_{\text{low}}(\mathbf{x}) + e_k(\mathbf{x})$, are assumed to have uncertainty that is normally distributed with zero mean and variance $\sigma_k^2(\mathbf{x})$, denoted $\mathcal{N}(0, \sigma_k^2(\mathbf{x}))$. In the two-fidelity optimization, only the value of the surrogate model is used, but in the following, both the value of the surrogate model and the uncertainty estimate are used.

For two lower-fidelity models, the estimates of the high-fidelity function are

$$f_{\text{high}}(\mathbf{x}) \approx f_{\text{med}}(\mathbf{x}) + e_{\text{med},k}(\mathbf{x}) + \mathcal{N}(0, \sigma^2_{\text{med},k}(\mathbf{x})) \tag{23}$$

$$f_{\text{high}}(\mathbf{x}) \approx f_{\text{low}}(\mathbf{x}) + e_{\text{low},k}(\mathbf{x}) + \mathcal{N}(0, \sigma^2_{\text{low},k}(\mathbf{x})) \tag{24}$$

From these two or more models, a maximum likelihood estimate of the high-fidelity function weights each prediction according to a function of the variance estimates. The high-fidelity maximum likelihood estimate has a mean $f_{est,k}$, given by

$$\begin{aligned} f_{est,k}(\mathbf{x}) = (f_{\text{med}}(\mathbf{x}) + e_{\text{med},k}(\mathbf{x}))\left[\frac{\sigma^2_{\text{low},k}(\mathbf{x})}{\sigma^2_{\text{low},k}(\mathbf{x}) + \sigma^2_{\text{med},k}(\mathbf{x})}\right] \\ + (f_{\text{low}}(\mathbf{x}) + e_{\text{low},k}(\mathbf{x}))\left[\frac{\sigma^2_{\text{med},k}(\mathbf{x})}{\sigma^2_{\text{low},k}(\mathbf{x}) + \sigma^2_{\text{med},k}(\mathbf{x})}\right] \end{aligned} \tag{25}$$

The estimate of the high-fidelity function also has a variance $\sigma^2_{est,k}$, which is less than either of the variances of the lower-fidelity models since

$$\frac{1}{\sigma^2_{est,k}(\mathbf{x})} = \frac{1}{\sigma^2_{\text{low},k}(\mathbf{x})} + \frac{1}{\sigma^2_{\text{med},k}(\mathbf{x})} \tag{26}$$

A thorough discussion of using a maximum likelihood estimator to combine two or more estimates with normally distributed uncertainties is available in [[35], Chapter 1]. We note that naming the two lower-fidelity estimates $f_{\text{low}}(\mathbf{x})$ and $f_{\text{med}}(\mathbf{x})$ may be misleading, since the maximum likelihood estimator makes no hierarchical distinction between models. In fact, our approach applies naturally to the case where different models have varying relative levels of fidelity over different regions of the design space.

We present a schematic of the behavior of this maximum likelihood estimator in Fig. 6. In the first case with two similar models, the combined estimate has a similar mean with a reduced variance. In the second case with two dissimilar estimates, the combined estimate has the average mean of the two models again with lower variance. In the third case when one model has a considerably smaller variance than the other model, the combined estimate has a similar mean and slightly reduced variance than the model with the lower variance. Accordingly, the maximum likelihood estimate is the best probabilistic guess of the high-fidelity function at a noncalibrated point.

This method provides flexibility while still being provably convergent to a high-fidelity optimum using our multifidelity optimization approach. The requirements for convergence are that the

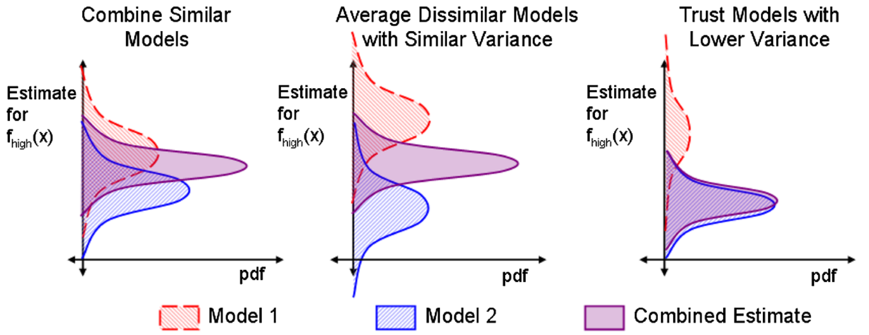

Combine Similar Models
Average Dissimilar Models with Similar Variance
Trust Models with Lower Variance
Estimate for f$_{high}$(x)
pdf
Estimate for f$_{high}$(x)
pdf
Estimate for f$_{high}$(x)
pdf
Model 1
Model 2
Combined Estimate

Fig. 6 Behavior of the combined maximum likelihood estimate given the behavior of the individual estimates.

surrogate model upon which the optimization is performed be smooth and exactly interpolate the function at the necessary calibration points. Using this maximum likelihood estimator, only one of the lower-fidelity functions needs to be sampled at the calibration points because at a calibration point an individual Gaussian process model has zero variance. Accordingly, at that calibration point the model is known to be correct and the other lower-fidelity information is not used. Therefore, the user has flexibility in selecting which of the lower-fidelity models are calibrated at a calibration point. For example, the calibration procedure could choose a ratio, such as one intermediate-fidelity update for every three low-fidelity updates, or simply update both the intermediate-fidelity and low-fidelity models each time a new calibration point is needed.

The changes to Algorithm 1 necessary to use multiple low-fidelity models are minimal. In steps 1, 3, and 7, $f_{est,k}(\mathbf{x})$ should be used in lieu of $m_k(\mathbf{x})$, and in step 6 at least one of the error models, $e_{\mathrm{med},k}(\mathbf{x})$ or $e_{\mathrm{low},k}(\mathbf{x})$, needs to be updated using Algorithm 2 to make $f_{est,k+1}(\mathbf{x})$ fully linear on the updated trust region.

Optimization results show that the nesting approach suffers from poor scaling between fidelity levels and that the maximum likelihood approach speeds convergence of our multifidelity optimization method even if the lowest-fidelity function is a poor representation of the high-fidelity function. In all examples presented, the calibration strategy employed for the maximum likelihood method is to update all lower-fidelity models whenever the optimization method requires a new calibration point.

The first example is an optimization of the Rosenbrock function with two parabolic lower-fidelity functions. The number of required function evaluations for each fidelity level is presented in Table 5. Using the maximum likelihood approach, the number of high-fidelity function evaluations has been reduced by 34%, and the number of combined lower-fidelity evaluations has been reduced by 27%. However, combining the multiple lower-fidelity functions through nesting leads to a large increase in the number of function evaluations at each level.

The second example is to optimize a supersonic airfoil for minimum drag with respect to an Euler code, Cart3D. Two lower-fidelity methods are used: shock-expansion theory and a panel method. These results, presented in Table 6, also show that the maximum likelihood approach converges faster and with fewer calibration points than the original multifidelity method using only the panel method. The nesting approach failed to converge as the step size required in the intermediate-fidelity optimization became too small. The likely cause of this is that the adjoint-based mesh refinement used in Cart3D allows numerical oscillations in the output functional at a level that is still significant in the optimization, and this makes the necessary calibration surface nonsmooth. The lack of smoothness violates the convergence criteria of this method.

Table 5 Number of function calls required to optimize the Rosenbrock function using multiple lower-fidelity functions. The maximum likelihood approach requires the least high-fidelity function evaluations to converge and the nested approach the most

Method	$(x_2 - x_1^2)^2 + (1 - x_1)^2$	$(x_1 - 1)^2 + x_2^2$	$x_1^2 + x_2^2$
Two-Fidelities	87	0	6975
Max. likelihood	57	2533	2533
Nested	137	4880	50,455

Table 6 Number of function calls required to optimize an airfoil for minimum drag using the Euler equations (Cart3D) with multiple lower-fidelity models

Method	Cart3D	Shock-Expansion	Panel method
Two-Fidelities	88	0	47,679
Max. likelihood	66	23,297	23,297
Nested	66[a]	7920[a]	167,644[a]

[a]Indicates that solution was not converged due to numerical limitations.

Table 7 Number of function calls required to optimize an airfoil for minimum drag using shock-expansion theory with multiple lower-fidelity models

Method	Shock-Expansion	Panel method	Camberline
Two-Fidelities	126	43665	0
Max. Likelihood	84	30057	30057
Nested	212[a]	59217[a]	342916[a]

[a]Indicates a minimum number of function evaluations as opposed to an average value from random starting points.

The final example demonstrates that the maximum likelihood approach can still benefit from a poor low-fidelity model. The results in Table 7 are for minimizing the drag of a supersonic airfoil using shock-expansion theory, with the panel method as an intermediate-fidelity function; however, unlike the preceding example, the lowest-fidelity model is quite poor and uses the panel method only on the camberline of the airfoil. Using this method, any symmetric airfoil at zero angle of attack has no drag and many of the predicted trends are incorrect compared with the panel method or shock-expansion theory. The optimization results show an important benefit of this maximum likelihood approach: even adding this additional bad information, the number of high-fidelity function calls has been reduced by 33%, and the number of intermediate-fidelity function calls has decreased by 31%. An additional point of note is the magnitude to which the nested approach suffers by adding poor low-fidelity information. In most test problems, the nested optimization was terminated due to an exceptionally large number of function evaluations. The results presented are the minimum number of function evaluations the nested approach required to converge.

From this observed behavior, we recommend the use of the two-fidelity optimization approach demonstrated in Sec. V for the optimization of any computationally expensive function for which accurate gradient information is not available. When considering the addition of a third or fourth fidelity level, we note there are diminishing returns for each additional fidelity level. In the three-fidelity examples presented, the third fidelity level reduced the number of high-fidelity evaluations in the two-fidelity case by about 30%, which is about an 8% additional reduction over a single-fidelity method. We therefore expect the greatest multifidelity benefit when going from a single-fidelity level to two-fidelity levels, and diminishing benefit for each additional level. However, if high-fidelity function evaluations are extremely costly (e.g., requiring hours or days on a supercomputer), then the additional reduction may certainly be worthwhile. Since the reduction in high-fidelity evaluations using gradient-free methods is similar to what has been observed using gradient-based multifidelity optimization methods [5], we expect similar returns for additional fidelity levels in the gradient-based case.

VII. Conclusions

This paper has presented a provably convergent multifidelity optimization method that does not require computation of derivatives of the high-fidelity function. The optimization results show that this method reduces the number of high-fidelity function calls required to find a local minimum compared with other state-of-the-art methods. The method creates surrogate models that retain accurate local behavior while also capturing some global behavior of the high-fidelity function. However, some downfalls of the method are that its performance is sensitive to the quality of the low-fidelity model and that its overhead increases dramatically with the number of design variables and the number of calibration points used to build the radial basis function model. Accordingly, this approach is only recommended for high-fidelity functions that require a considerable wall-clock time. Furthermore, in this unconstrained optimization algorithm engineering design constraints are handled with a penalty method, but using surrogate models for the constraints and objective may be desirable. While trust region methods exist for constrained optimization problems, incorporating multifidelity models in both objective function and constraints presents several challenges for future work.

This paper has also shown that a multifidelity optimization method based on a maximum likelihood estimator is an effective way of combining many fidelity levels to optimize a high-fidelity function. The maximum likelihood estimator permits flexible sampling strategies among the low-fidelity models and is robust with respect to poor low-fidelity estimates. In addition, the estimator offers a natural and automated way of selecting among different models that are known to be accurate in different parts of the design space, which is frequently the case in engineering design.

Acknowledgments

The authors gratefully acknowledge support from NASA Langley Research Center contract NNL07AA33C, technical monitor Natalia Alexandrov, and a National Science Foundation graduate research fellowship. In addition, we wish to thank Michael Aftosmis and Marian Nemec for support with Cart3D, and Andrew Conn for useful comments in improving this paper.

References

[1] Simpson, T., Peplinski, J., Koch, P., and Allen, J., "Metamodels for Computerbased Engineering Design: Survey and Recommendations," *Engineering with Computers*, Vol. 17, No. 2, 2001, pp. 129–150. doi:10.1007/PL00007198

[2] Venter, G., Haftka, R., and Starnes, J., "Construction of Response Surface Approximations for Design Optimization," *AIAA Journal*, Vol. 36, No. 12, 1998, pp. 2242–2249. doi:10.2514/2.333

[3] Eldred, M., Giunta, A., and Collis, S., "Second-Order Corrections for Surrogate-Based Optimization with Model Hierarchies," *Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA Paper 2004-4457, 2004.

[4] Antoulas, A., *Approximation of Large-Scale Dynamical Systems*, Cambridge Univ. Press, New York, 2005.

[5] Alexandrov, N., Lewis, R., Gumbert, C., Green, L., and Newman, P., "Approximation and Model Management in Aerodynamic Optimization with Variable-Fidelity Models," *Journal of Aircraft*, Vol. 38, No. 6, Nov.–Dec. 2001, pp. 1093–1101.

[6] Alexandrov, N., Dennis, J., Lewis, R., and Torczon, V., "A Trust Region Framework for Managing the Use of Approximation Models in Optimization," NASA, Tech. Rept. CR-201745, Oct. 1997.

[7] Alexandrov, N., Lewis, R., Gumbert, C., Green, L., and Newman, P., "Optimization with Variable-fidelity Models Applied to Wing Design," NASA, Tech. Rept. CR-209826, Dec. 1999.

[8] Haftka, R. T., and Gürdal, Z., *Elements of Structural Optimization, Vol 11 of Solid Mechanics and its Applications*, Kluwer Academic Publishers, Norwell, MA, 1992.

[9] Jones, D., Schonlau, M., and Welch, W., "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, Vol. 13, No. 4, 1998, pp. 455–492. doi:10.1023/A:1008306431147

[10] Booker, A. J., Dennis, J. E., Frank, P. D., Serafini, D. B., Torczon, V., and Trosset, M. W., "A Rigorous Framework for Optimization of Expensive Functions by Surrogates," *Structural Optimization*, Vol. 17, No. 1, February 1999, pp. 1–13. doi:10.1007/BF01197708

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

[12] Giunta, A. A., Dudley, J. M., Narducci, R., Grossman, B., Haftka, R. T., Mason, W. H., and Watson, L. T., "Noisy Aerodynamic Response and Smooth Approximations in HSCT Design," AIAA Paper 1994-4376, 1994.

[13] Giunta, A. A., and Eldred, M. S., "Implementation of a Trust Region Model Management Strategy in the Dakota Optimization Toolkit," *8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA Paper 2000-4935, Sept. 2000.

[14] Gutmann, H.-M., "A Radial Basis Function Method for Global Optimization," *Journal of Global Optimization*, Vol. 19, No. 3, 2001, pp. 201–227. doi:10.1023/A:1011255519438

[15] Jones, D., "A Taxonomy of Global Optimization Methods Based on Response Surfaces," *Journal of Global Optimization*, Vol. 21, No. 4, 2001, pp. 345–383. doi:10.1023/A:1012771025575

[16] Kennedy, M., and O'Hagan, A., "Bayesian Calibration of Computer Models," *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, Vol. 63, No. 3, 2001, pp. 425–464.

[17] Kennedy, M., and O'Hagan, A., "Predicting the Output from a Complex Computer Code When Fast Approximations are Available," *Biometrika*, Vol. 87, No. 1, 2000, pp. 1–13. doi:10.1093/biomet/87.1.1

[18] Leary, S., Bhaskar, A., and Keane, A., "A Knowledge-Based Approach to Response Surface Modelling in Multifidelity Optimization," *Journal of Global Optimization*, Vol. 26, No. 3, 2003, pp. 297–319. doi:10.1023/A:1023283917997

[19] Sasena, M. J., Papalambros, P., and Goovaerts, P., "Exploration of Metamodeling Sampling Criteria for Constrained Global Optimization," *Engineering Optimization*, Vol. 34, No. 3, 2002, pp. 263–278. doi:10.1080/03052150211751

[20] Castro, J., Gray, G., Giunta, A., and Hough, P., "Developing a Computationally Efficient Dynamic Multilevel Hybrid Optimization Scheme using Multifidelity Model Interactions," SAND, Tech. Rept. 2005-7498, Nov. 2005.

[21] Carter, R., "On the Global Convergence of Trust Region Algorithms Using Inexact Gradient Information," *SIAM Journal on Numerical Analysis*, Vol. 28, No. 1, 1991, pp. 251–265. doi:10.1137/0728014

[22] Conn, A., Gould, N., and Toint, P., *Trust-Region Methods*, MPS/SIAM Series on Optimization, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2000.

[23] Oeuvray, R., "Trust-Region Methods Based on Radial Basis Functions with Application to Biomedical Imaging," Ph.D. Thesis, Ecole Polytechnique Federale de Lausanne, 2005.

[24] Conn, A., Scheinberg, K., and Vicente, L., "Geometry of Interpolation Sets in Derivative Free Optimization," *Mathematical Programming*, Vol. 111, Nos. 1–2, 2006, pp. 141–172. doi:10.1007/s10107-006-0073-5

[25] Conn, A., Scheinberg, K., and Vicente, L., "Global Convergence of General Derivative-Free Trust-Region Algorithms to First- and Second-Order Critical Points," *SIAM Journal on Optimization*, Vol. 20, No. 1, 2009, pp. 387–415. doi:10.1137/060673424

[26] Wild, S., Regis, R., and Shoemaker, C., "ORBIT: Optimization by Radial Basis Function Interpolation in Trust-Regions," *SIAM Journal on Scientific Computing*, Vol. 30, No. 6, 2008, pp. 3197–3219. doi:10.1137/070691814

[27] Wild, S., "Derivative-Free Optimization Algorithms for Computationally Expensive Functions," Ph.D. Thesis, Cornell University, Jan. 2009.

[28] Wild, S., and Shoemaker, C. A., "Global Convergence of Radial Basis Function Trust-Region Algorithms," Mathematics and Computer Science Division, Tech. Rept. Preprint ANL/MCS-P1580-0209, Feb. 2009.

[29] Nocedal, J., and Wright, S., *Numerical Optimization*, 2nd ed., Springer, New York, 2006.

[30] Lophaven, S., Nielsen, H., and Sondergaard, J., "Aspects of the Matlab Toolbox DACE," Technical University of Denmark, Tech. Rept. IMM-REP-2002-13, Aug. 2002.

[31] Rasmussen, C., and Williams, C., *Gaussian Processes for Machine Learning*, The MIT Press, Cambridge, MA, 2006.

[32] MathWorks, Inc., "Unconstrained Nonlinear Optimization Algorithms," Optimization Toolbox User's Guide, V. 5, 2010.

[33] Jones, D. R., Perttunen, C. D., and Stuckmann, B. E., "Lipschitzian Optimization Without the Lipschitz Constant," *Journal of Optimization Theory and Applications*, Vol. 79, No. 1, Oct. 1993, pp. 157–181. doi:10.1007/BF00941892

[34] Nemec, M., Aftosmis, M., and Wintzer, M., "Adjoint-Based Adaptive Mesh Refinement for Complex Geometries," *46th AIAA Aerospace Sciences Meeting*, AIAA Paper 2008-0725, Jan. 2008.

[35] Maybeck, P. S., *Stochastic Models, Estimation, and Control*, Vol. 1, Academic Press, New York, 1979.

R. Kapania
Associate Editor