51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference<BR> 18th
12 - 15 April 2010, Orlando, Florida

AIAA 2010-2912

# A Provably Convergent Multifidelity Optimization Algorithm not Requiring High-Fidelity Derivatives

Andrew March*, Karen Willcox†

*Massachusetts Institute of Technology, Cambridge, Massachusetts, 02139.*

**This paper presents a provably convergent multifidelity optimization algorithm for unconstrained problems. 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 an unconstrained 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 |
| $C$ | Continuous function |
| $c$ | A positive finite constant |
| $\mathbf{d}$ | High-fidelity function sample point |
| $\delta x$ | Finite difference step size |
| $e(\mathbf{x})$ | 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 |
| $\mathcal{N}$ | Normal distribution |
| $n$ | number of design variables |
| $p_{max}$ | Maximum number of points to be used in the interpolation |
| $RBF$ | Radial Basis Function |
| $r$ | Radius |
| $\mathbf{s}$ | Trust region step |
| $\mathbf{x}$ | Design vector |

*Graduate Student, Department of Aeronautics and Astronautics, MIT, amarch@mit.edu, Student Member AIAA
†Associate Professor, Department of Aeronautics and Astronautics, MIT, kwillcox@mit.edu, Associate Fellow AIAA

| | |
|---|---|
| $x_i$ | 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 |
| $\alpha$ | Trust region contraction ratio used in convergence check |
| $\epsilon$ | Termination tolerance for gradient norm |
| $\epsilon_2$ | Termination tolerance for trust region size |
| $\Delta$ | Trust region radius |
| $\phi$ | Radial basis function |
| $\gamma_0$ | Trust region contraction ratio |
| $\gamma_1$ | Trust region expansion ratio |
| $\eta$ | Trust region update criterion |
| $\kappa$ | Bound related to function smoothness |
| $\lambda$ | Coefficient of a radial basis function |
| $\mu$ | Mean of a Gaussian process |
| $\nu$ | Coefficients of the polynomial basis |
| $\Pi$ | Polynomial basis for $\mathbb{R}^n$ |
| $\pi$ | A component of the polynomial basis |
| $\theta$ | Positive constant, used to build fully linear models |
| $\rho$ | Ratio of actual improvement to predicted improvement |
| $\sigma^2$ | Variance of a Gaussian process |
| $\xi$ | Radial basis function correlation length |

*Superscript*

| | |
|---|---|
| $^*$ | Optimal |

*Subscript*

| | |
|---|---|
| $_0$ | Initial iterate |
| $_{bhm}$ | Bound for Hessian of the surrogate model |
| $_{blg}$ | Upper bound on the Lipschitz constant |
| $_{\text{est}}$ | Maximum likelihood estimate |
| $_f$ | Relating to function |
| $_{FCD}$ | Fraction of Cauchy Decrease |
| $_g$ | Relating to gradient |
| $_{\text{high}}$ | 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 |
| $_{\text{low}}$ | Relating to a lower-fidelity function |
| $_{\text{max}}$ | User-set maximum value of that parameter |
| $_{\text{min}}$ | User-set minimum value of that parameter |
| $_{\text{med}}$ | Relating to an intermediate-fidelity function |

# I. Introduction

The expense of either building or testing a complex system drives system 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] Another multifidelity approach is to combine a pattern-search with conformal space mapping, where the least squares difference between a low-fidelity function and high-fidelity function is minimized at a collection of points by mapping the low-fidelity design space to the high-fidelity design space. This method is also provably convergent to an optimum of the high-fidelity function.[8] A third general approach is Efficient Global Optimization (EGO) developed by Jones *et al.*[9] In this method, a Bayesian uncertainty approach is used to find regions in the design space with a high likelihood of having an optimal solution. EGO uses a combination of a regression model and an uncertainty estimate based on distance from known points as a way to find better points. An improvement to Jones' approach is to use model calibration techniques to model the difference or quotient between a high- and low-fidelity function as opposed to modeling the high-fidelity function itself. In this way, a low-fidelity model can increase the efficiency of finding an optimum of a high-fidelity function in situations where using only a regression surface requires a considerable number of function evaluations for calibration.[10–12] These model calibration techniques are generally based on heuristic methods and are not provably convergent to an optimum of the high-fidelity function.

In this paper, we present a provably convergent multifidelity optimization algorithm based on model calibration. 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 re-used. The challenge we address here is to produce a surrogate model that captures local function behavior sufficiently well to prove convergence, while capturing global function behavior to speed convergence. Carter proved that a trust-region algorithm is convergent provided 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.[13] Oeuvray showed that a radial basis function interpolation satisfies this criterion from Carter, provided the interpolation points satisfy certain conditions.[14] Conn *et al.* 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.[15] Conn *et al.* 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 trust-region framework.[16] Wild *et al.* then developed an algorithm to produce fully linear radial basis function interpolation models and showed that his method could be used within Conn's provably convergent optimization framework.[17–19]

This paper combines the provably convergent optimization frameworks of Wild *et al.* and Conn *et al.* 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.* [16] Section III discusses the approach of Wild *et al.* [17] 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 *et al.* [19] 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, Section 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.* [16] to solve (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 $k$th trust-region iteration. Following the general Bayesian calibration approach in Ref. 11, we define $e_k(\mathbf{x})$ to be a model of the error between the high- and low-fidelity functions on the $k$th 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 $\Delta_k$,

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

where any norm can be used, provided there exist constants $c_1$ and $c_2$ such that

$$\|\cdot\|_2 \leq c_1\|\cdot\| \qquad \text{and} \qquad \|\cdot\| \leq c_2\|\cdot\|_2. \tag{4}$$

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 local minimum 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 (i) continuously differentiable, (ii) have a Lipschitz continuous derivative, and (iii) be bounded from below within a region of a relaxed level-set, $\mathcal{L}(\mathbf{x}_0)$, defined as

$$L(\mathbf{x}_0) = \{\mathbf{x} \in \mathbb{R}^n : f_{\text{high}}(\mathbf{x}) \leq f_{\text{high}}(\mathbf{x}_0)\} \tag{5}$$

$$B(\mathbf{x}_k) = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x} - \mathbf{x}_k\| \leq \Delta_{\max}\} \tag{6}$$

$$\mathcal{L}(\mathbf{x}_0) = L(\mathbf{x}_0) \bigcup_{\mathbf{x_k} \in L(\mathbf{x}_0)} B(\mathbf{x}_k), \tag{7}$$

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.*:[16]

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

*There exist positive constants $\kappa_f, \kappa_g$ and $\kappa_{blg}$ 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*

$$\|\nabla f_{high}(\mathbf{x}) - \nabla m_k(\mathbf{x})\| \leq \kappa_g \Delta_k \quad \forall \mathbf{x} \in \mathcal{B}_k \tag{8}$$

*and the error between the model and the function satisfies*

$$|f_{high}(\mathbf{x}) - m_k(\mathbf{x})| \leq \kappa_f \Delta_k^2 \quad \forall \mathbf{x} \in \mathcal{B}_k. \tag{9}$$

*Such a model $m_k(\mathbf{x})$ is called fully linear on $\mathcal{B}_k$.*[16]

At iteration $k$, the trust-region algorithm solves the subproblem

$$\begin{aligned} \min_{\mathbf{s}_k} \quad & m_k(\mathbf{x}_k + \mathbf{s}_k) \\ \text{s.t.} \quad & \|\mathbf{s}_k\| \leq \Delta_k \end{aligned} \tag{10}$$

to determine the trust-region step $\mathbf{s}_k$. 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 2-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 $m_k(\mathbf{x}_k)$.[16] The sufficient decrease condition requires the step to satisfy the fraction of Cauchy decrease. As given in Ref. 16 and Ref. 18, this requires that for some constant, $\kappa_{FCD} \in (0, 1)$, the step $\mathbf{s}_k$ satisfies

$$m_k(\mathbf{x}_k) - m_k(\mathbf{x}_k + \mathbf{s}_k) \geq \frac{\kappa_{FCD}}{2}\|\nabla m_k(\mathbf{x}_k)\| \min\left[\frac{\|\nabla m_k(\mathbf{x}_k)\|}{\kappa_{bhm}}, \frac{\|\nabla m_k(\mathbf{x}_k)\|_2}{\|\nabla m_k(\mathbf{x}_k)\|}\Delta_k\right]. \tag{11}$$

The high-fidelity function $f_{\text{high}}$ 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

$$\rho_k = \frac{f_{\text{high}}(\mathbf{x}_k) - f_{\text{high}}(\mathbf{x}_k + \mathbf{s}_k)}{m_k(\mathbf{x}_k) - m_k(\mathbf{x}_k + \mathbf{s}_k)}. \tag{12}$$

The trial point is accepted or rejected according to

$$\mathbf{x}_{k+1} = \begin{cases} \mathbf{x}_k + \mathbf{s}_k & \text{if } \rho_k > 0 \\ \mathbf{x}_k & \text{otherwise.} \end{cases} \tag{13}$$

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, $\Delta_k$, 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

$$\Delta_{k+1} = \begin{cases} \min\{\gamma_1 \Delta_k, \Delta_{\max}\} & \text{if } \rho_k \geq \eta \\ \gamma_0 \Delta_k & \text{if } \rho_k < \eta, \end{cases} \tag{14}$$

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 $\Delta_{k+1}$.

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 $\epsilon$, 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. 8, 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 $\Delta_{k+1}$. 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 $\Delta_{k+1}$, or $\Delta_{k+1} \leq \epsilon_2$, for a small $\epsilon_2$, which terminates the algorithm. This process of checking for convergence is referred to as the criticality check in Conn *et al.*[16]

## 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 Section II. This section first presents an overview of the radial basis function (RBF) interpolation approach of Wild *et al.*,[17] 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 $j$th 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 and all of the vectors $\mathbf{y}_i$. This notation is shown graphically in Figure 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 $\mathcal{Y}$,

$$e_k(\mathbf{x}_k + \mathbf{y}_i) = f_{\text{high}}(\mathbf{x}_k + \mathbf{y}_i) - f_{\text{low}}(\mathbf{x}_k + \mathbf{y}_i) \quad \forall \mathbf{y}_i \in \mathcal{Y}. \tag{15}$$

The RBF interpolation has the form

$$e_k(\mathbf{x}) = \sum_{i=1}^{|\mathcal{Y}|} \lambda_i \phi\left(\|\mathbf{x} - \mathbf{x}_k - \mathbf{y}_i\|\right) + \sum_{i=1}^{n+1} \nu_i \pi_i(\mathbf{x} - \mathbf{x}_k), \tag{16}$$

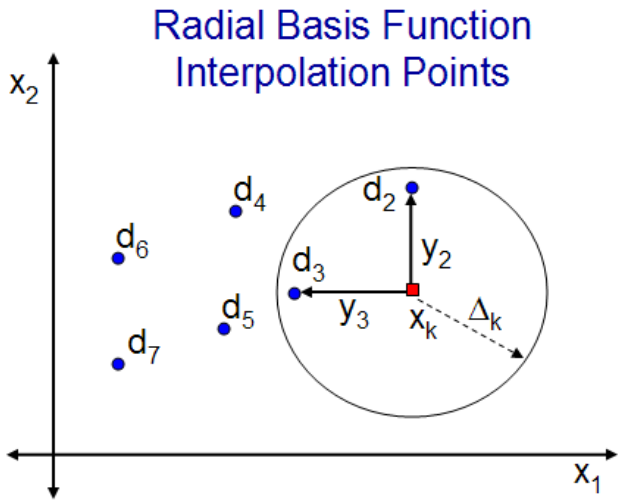



**Figure 1. Graphical representation of the notation used to define points and vectors in and around the trust region.**

where $\phi$ is any positive definite, twice continuously differentiable RBF with $\phi'(0) = 0$, and the second term in (16) represents a linear tail, where $\pi_i$ denotes the $i^{th}$ component of the vector $\Pi(\mathbf{x} - \mathbf{x}_k) = [1\ (\mathbf{x} - \mathbf{x}_k)]^T$. The coefficients $\lambda_i$ and $\nu_i$ represent the RBF interpolation, and are found by the QR-factorization technique of Wild *et al.*[17] In order for the model to be fully linear, the RBF coefficients $\lambda_i$ and $\nu_i$ must be bounded in magnitude. This is achieved by using the interpolation point selection method in Wild *et al.*[17] 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 for affine independence. If fewer than $n+1$ affinely independent points are found, additional high-fidelity function evaluations are required to generate them. 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 $\lambda_i$ and $\nu_i$. 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 (i) and (ii) above.[18]

In order 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 (i) and (ii) above. Condition (i), 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 (ii), 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. (7). For the high-fidelity function we require

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

and for the low-fidelity function,

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

with Lipschitz constants $\kappa_{\text{high}}$ and $\kappa_{\text{low}}$, respectively. Since the difference between any two functions with Lipschitz continuous first derivatives is also Lipschitz continuous, we obtain

$$\frac{\|\nabla[f_{\text{high}}(\mathbf{x}_1) - f_{\text{low}}(\mathbf{x}_1)] - \nabla[f_{\text{high}}(\mathbf{x}_2) - f_{\text{low}}(\mathbf{x}_2)]\|}{\|\mathbf{x}_1 - \mathbf{x}_2\|} \leq \kappa_{\text{high}} + \kappa_{\text{low}} \quad \forall \mathbf{x}_1, \mathbf{x}_2 \in \mathcal{L}(\mathbf{x}_0), \tag{19}$$

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.* [16] holds, and this multifidelity algorithm is provably convergent to an optimum of the high-fidelity function.

# IV. Numerical Implementation of Algorithms

This section presents an overview of the numerical implementation of the multifidelity optimization algorithm and suggests a manner in which the method of Wild *et al.* [19] to generate fully linear models can be used in a flexible Bayesian calibration setting. The first subsection, Section IV.A, implements the the trust region based optimization algorithm presented in Section II. Whenever creation of a new fully linear model is needed, the method discussed in Section III is implemented using the algorithm presented in Section IV.B.

## IV.A. Trust Region Implementation

Algorithm 1 provides an overview of the numerical implementation of the trust-region optimization method presented in Section II. For each trust-region iteration, the algorithm guarantees that a step is found that satisfies the fraction of Cauchy decrease, Eq. 11. 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 re-evaluated. 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 Section II, while recommended values and sensitivity of results to those values will be presented in Section V.

**Algorithm 1:** Trust-Region Algorithm for Iteration $k$

1: Solve the trust-region subproblem using nonlinear programming techniques to find the $\mathbf{s}_k$ that solves,

$$\begin{aligned} \min_{\mathbf{s}_k} \quad & m_k(\mathbf{x}_k + \mathbf{s}_k) \\ \text{s.t.} \quad & \|\mathbf{s}_k\| \leq \Delta_k. \end{aligned}$$

1a: If the subproblem solution fails to satisfy the fraction of Cauchy decrease, Eq. 11, the simple line search from Conn *et al.* is used.[16]

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,

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

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 $\rho_k$,

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

5: Create a new model $m_{k+1}(\mathbf{x})$ that is fully linear on $\{\mathbf{x} : \|\mathbf{x} - \mathbf{x}_{k+1}\| \leq \Delta_{k+1}\}$ using Algorithm 2.

6: Check for convergence: if $\|\nabla m_{k+1}(\mathbf{x}_{k+1})\| > \epsilon$, algorithm is not converged—go to step 1. Otherwise,

6a: While $\|\nabla m_{k+1}(\mathbf{x}_{k+1})\| \leq \epsilon$ and $\Delta_{k+1} > \epsilon_2$,

6b: Reduce the trust region size, $\alpha \Delta_{k+1} \rightarrow \Delta_{k+1}$.

6c: Update model $m_{k+1}(\mathbf{x})$ to be fully linear on $\{\mathbf{x} : \|\mathbf{x} - \mathbf{x}_{k+1}\| \leq \Delta_{k+1}\}$ using Algorithm 2.

### IV.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 $\xi$ 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 $\xi$ that best represents the data.[20,21] Therefore, our process to generate a fully linear surrogate model uses the method of Wild *et al.*[17] 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 $\xi^*$, where $\xi^*$ is chosen as the value of $\xi$ 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 Section VI).

**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.* [19]).
4: Compute the RBF coefficients using the QR factorization technique of Wild *et al.* [17]
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.[20,21]
6: Select the $\xi_i$ with the maximum likelihood.
  6a: If the maximum likelihood is $-\infty$ choose the largest $\xi_i$. This model corresponds to a linear regression of the high-fidelity function at the calibration points included in $\mathcal{Y}$, but still satisfies conditions for convergence.
7: Return the set of calibration vectors $\mathcal{Y}$, RBF coefficients, and updated database of high-fidelity function evaluations.

## 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.

### V.A. Rosenbrock Function

The first example multifidelity optimization example is the Rosenbrock function,

$$\min_{\mathbf{x}\in\mathbb{R}^2} f_{\text{high}}(\mathbf{x}) = \left(x_2 - x_1^2\right)^2 + \left(1 - x_1\right)^2. \tag{20}$$

The minimum of the Rosenbrock function is at $x^* = (1, 1)$ and $f(x^*) = 0$. Table 2 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. Convergence results are presented for the case when the low-fidelity function is parabolic, $f_{\text{low}}(\mathbf{x}) = x_1^2 + x_2^2$, in Figure 2(a), and a surface plot of the Rosenbrock function and this low-fidelity function is shown in Figure 2(b). For all of the examples in this section the optimization parameters used are given in Table 1 and are discussed in the remainder of this subsection.

| Parameter | Description | value |
|---|---|---|
| $\phi(r)$ | RBF Correlation | $e^{-r^2/\xi^2}$ |
| $\xi$ | RBF spatial correlation length | See Table 2 |
| $\Delta_0$ | Initial trust region size | $\max[10, \|\mathbf{x}_0\|_\infty]$ |
| $\Delta_{\max}$ | Maximum trust region size | $10^3\Delta_0$ |
| $\epsilon, \epsilon_2$ | Termination tolerances | $5 \times 10^{-4}$ |
| $\gamma_0$ | Trust region contraction ratio | 0.5 |
| $\gamma_1$ | Trust region expansion ratio | 2 |
| $\eta$ | Trust region expansion criterion | 0.2 |
| $\alpha$ | Trust region contraction ratio used in convergence check | 0.9 |
| $\kappa_{FCD}$ | Fraction of Cauchy decrease requirement | $10^{-4}$ |
| $p_{\max}$ | Maximum number of calibration points | 50 |
| $\theta_1$ | Minimum projection into null-space of calibration vectors | $10^{-3}$ |
| $\theta_2$ | RBF coefficient conditioning parameter | $10^{-4}$ |
| $\theta_3$ | Expanded trust-region size to find basis, $\theta_3\Delta_k$ | 10 |
| $\theta_4$ | Maximum calibration region size, $\theta_4\Delta_k$ | 10 |
| $\delta x$ | Finite difference step size | $10^{-6}$ |

**Table 1. Optimization parameters used in the Rosenbrock function demonstration.**

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, $\xi$. The first method is to fix a value of $\xi$, 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.[20,21] 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 2 show that the correlation length has a moderate impact on the convergence rate of the method. For this problem, using either $\xi = 2$ or $\xi^*$, the correlation length that maximizes the likelihood at each trust-region iteration, leads to the best result.

The results in Table 2 show that using a multifidelity framework can reduce the number of high-fidelity function calls. As a baseline, the average number of function calls for a quasi-Newton method directly optimizing the Rosenbrock function is 69. The Bayesian calibration approach uses between 5 and 180 high-fidelity function evaluations depending on the quality of the low-fidelity model. The worst case, 180 high-fidelity function evaluations, corresponds to not having a lower-fidelity model and simply approximating the function with a RBF interpolation. The best case, 5 high-fidelity evaluations, corresponds to the case when the low-fidelity function exactly models the high-fidelity function. With a moderately good low-fidelity function (e.g., a $4^{th}$ degree polynomial), the multifidelity method performs similarly to the quasi-Newton method. Clearly the performance of this method compared to conventional optimization methods depends considerably on the quality of the low-fidelity function used. However, when this method is compared with other multifidelity methods such as the first-order consistent trust-region approaches of Alexandrov *et al.*,[5] it

uses fewer high-fidelity function evaluations. Results for two first-order consistent trust-region methods are presented in Table 2 along with the results of the Bayesian calibration method. The first uses a multiplicative correction defined in Ref. 5, while the second uses an additive correction. The first-order consistent methods require $n+1$ high-fidelity function evaluations to estimate the high-fidelity gradient at each $\mathbf{x}_k$.

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. 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 to the cost of repeatedly solving for RBF coefficients.

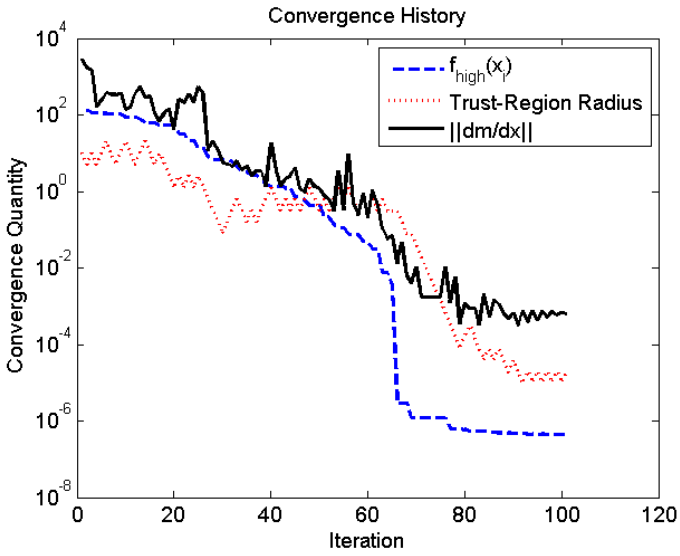
| Low-Fidelity Function | $\xi=1$ | $\xi=2$ | $\xi=3$ | $\xi=5$ | $\xi^*$ | Mult.-Corr. | add-Corr. |
|---|---|---|---|---|---|---|---|
| $f_{\text{low}}(\mathbf{x})=0$ | 148 | 107 | 177 | 223 | 178 | 289 | 503 |
| $f_{\text{low}}(\mathbf{x})=x_1^2+x_2^2$ | 129 | 77 | 106 | 203 | 76 | 312 | 401 |
| $f_{\text{low}}(\mathbf{x})=x_1^4+x_2^2$ | 74 | 74 | 73 | 87 | 65 | 171 | 289 |
| $f_{\text{low}}(\mathbf{x})=f_{\text{high}}(\mathbf{x})$ | 5 | 5 | 5 | 5 | 7 | 7 | 6 |
| $f_{\text{low}}(\mathbf{x})=-x_1^2-x_2^2$ | 195 | 130 | 132 | 250 | 100 | 352 | fail |

**Table 2. Table of average number of function evaluations required to minimize the Rosenbrock function, Eq. 20, from a random initial point on $x_1, x_2 \in [-5,5]$. Results for a selection of Gaussian radial bases function spatial parameters, $\xi$, are shown. $\xi^*$ corresponds to optimizing the spatial parameter according to a maximum likelihood criteria.[20] Also included are the number of function evaluations required using first-order consistent trust region methods with a multiplicative correction and an additive correction. For a standard quasi-Newton method the average number of function evaluations is 69.**

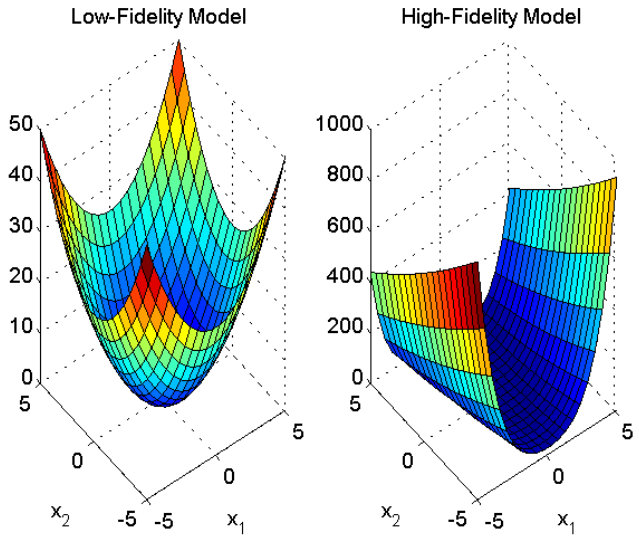
As with any optimization algorithms, tuning parameters can affect performance significantly; however, the best choices for these tuning parameters can be highly problem dependent. A sensitivity study measured the impact of algorithm parameters on the number of high-fidelity function evaluations for the Rosenbrock example using $f_{\text{low}}(\mathbf{x})=x_1^2+x_2^2$ as the low-fidelity function (Figure 2). For all of these tests, one parameter is varied and the remainder are all set to the values in Table 1. 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 $\eta$ 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 $\eta$ have the fewest high-fidelity function calls, and any value $0 \leq \eta \leq 0.2$ performs well. For the trust region expansion ratio, $\gamma_1$, the best results are at $\gamma_1 \approx 2$, and high-fidelity function evaluations increase substantially for other values. Similarly, for the contraction ratio, $\gamma_0$, 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, $\kappa_{FCD}$, the results show the number of high-fidelity evaluations is fairly insensitive to any value $0 < \kappa_{FCD} < 10^{-2}$. Similarly, for the trust-region contraction ratio used in the algorithm convergence check, $\alpha$, the number of high-fidelity function evaluations is insensitive to any value $0.5 < \alpha < 0.95$.

The method of Wild *et al.* to generate fully linear models requires four tuning parameters, $\theta_1$, $\theta_2$, $\theta_3$, and $\theta_4$.[17,19] The parameter $\theta_1$ $(0 < \theta_1 < 1)$ determines the acceptable points when finding the affinely independent basis in the vicinity of the trust region in Algorithm 2. As $\theta_1$ increases, the calibration points added to the basis must have a larger projection into the null-space 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 $\theta_1$ within two orders of magnitude of this value, the number of function evaluations increases by less than 50%. The second parameter, $\theta_2$ $(0 < \theta_2 < 1)$, is used in the AddPoints algorithm of Wild *et al.* [19] to ensure that the RBF coefficients remain bounded when adding additional calibration points. The number of allowable calibration points increases as $\theta_2$ 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, $\theta_3$ and $\theta_4$, used in the calibration point selection algorithm, are significant to the algorithm's performance. The parameter $\theta_3$ $(\theta_3 > 1)$ 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 $\{\mathbf{x} : \|\mathbf{x}-\mathbf{x}_k\| \leq \theta_3\Delta_k\}$ in order to find $n+1$ affinely independent points prior to evaluating the high-fidelity function in additional




(a) Convergence plot.



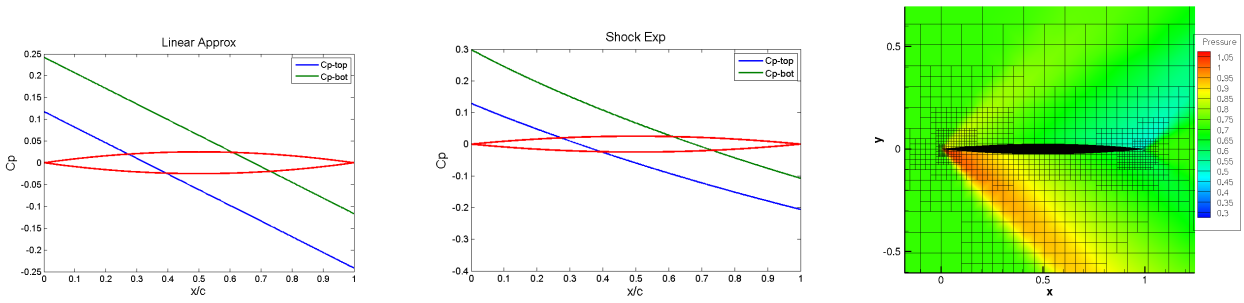

(b) Parabolic low-fidelity function and the Rosenbrock function.

**Figure 2. Rosenbrock function and a similar low-fidelity model.**

locations. The results show that the number of function calls is insensitive to $\theta_3$ for $1 < \theta_3 \leq 10$, with $\theta_3 \approx 3$ yielding the best results. The parameter $\theta_4$ ($\theta_4 > 1$) 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 $\{\mathbf{x} : \|\mathbf{x} - \mathbf{x}_k\| \leq \theta_4 \Delta_k\}$ 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 $\theta_4 \approx 10$ is the best value, with the number of high-fidelity function calls increasing substantially if $\theta_4 < 5$ or $\theta_4 > 15$.

### V.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 an Euler solver Cart3D.[22] 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° 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 eleven 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.




(a) Panel method. (b) Shock-expansion. (c) Cart3D.

**Figure 3. Supersonic airfoil model comparisons at Mach 1.5 and 2° angle of attack.**

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.

| | Panel | Shock-Expansion | Cart3D |
|---|---|---|---|
| $C_L$ | 0.1244 | 0.1278 | 0.1250 |
| % Diff | 0.46% | 2.26% | 0.00% |
| $C_D$ | 0.0164 | 0.0167 | 0.01666 |
| % Diff | 1.56% | 0.24% | 0.00% |

**Table 3. 5% thick biconvex airfoil results comparison at Mach 1.5 and 2° angle of attack. Percent difference is taken with respect to the Cart3D results.**

The optimization parameters used by this method are the same as in Table 1, with the exception that the RBF correlation length is either $\xi = 2$ or optimized at each iteration. A consecutive step size of less than $5 \times 10^{-6}$ 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 Figure 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 than the multiplicative-correction approach. In addition, the additive correction outperforms the multiplicative correction for this problem, but the RBF calibration outperformed it. The method of maximizing the likelihood of the RBF calibration performs slightly better than just using a fixed correlation length.

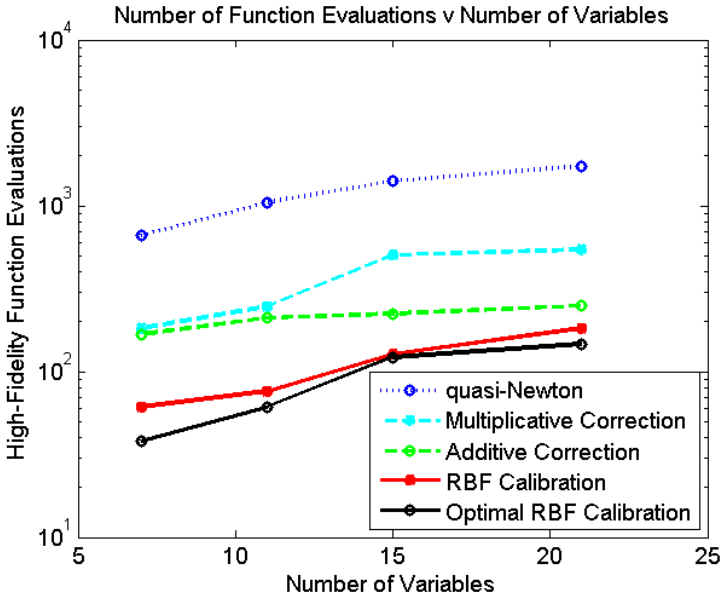



**Figure 4. Number of shock-expansion theory evaluations required to minimize the drag of a supersonic airfoil verse the number of parameters. The low-fidelity model is the supersonic panel method.**

As a final 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 not Lipschitz continuous because there is finite precision and a finite difference estimate of the derivative only measures numerical noise. However, in the calibration algorithm the trust region radius converges to zero, which forces a small step size and this is a supplemental termination criteria. So the method is not provably convergent in this case, but it still does converge to the correct solution. On average, the airfoil parameterized with 11 variables requires 88 high-fidelity (Cart3D) function evaluations. A comparison of the optimum airfoils from the panel method, shock-expansion theory, and Cart3D is presented in Figure 5.

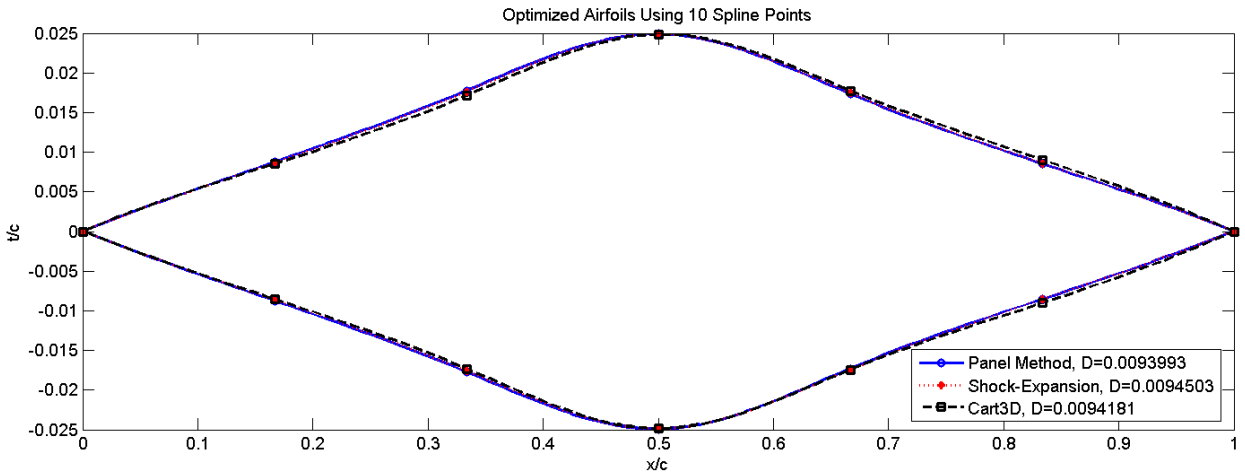



**Figure 5. Minimum drag airfoils from each of the three analysis models. The panel method airfoil is generated by a quasi-Newton method, but the shock-expansion and Cart3D airfoils are generated with this RBF calibration method using the panel method as a low-fidelity model.**

## 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,

$$f_{\text{high}}(\mathbf{x}) \approx f_{\text{med}}(\mathbf{x}) + e_{\text{med}}(\mathbf{x}) \tag{21}$$
$$f_{\text{med}}(\mathbf{x}) \approx f_{\text{low}}(\mathbf{x}) + e_{\text{low}}(\mathbf{x}). \tag{22}$$

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 $e_{\text{med}}$. 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

$$m(\mathbf{x}) \approx f_{\text{low}}(\mathbf{x}) + e_{\text{low}}(\mathbf{x}) + e_{\text{med}}(\mathbf{x}), \tag{23}$$

but only the low-fidelity calibration model $e_{\text{low}}$ is adjusted.

A problem with the nested approach is that on the low-fidelity optimization, a constrained optimization that uses model calibration techniques must be performed due to the trust region at the higher level. Moreover, this method likely requires a considerable reduction in the resources required to run each of the lower-fidelity models. The reason for this is that in order 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. So, if an optimization routine requires 50 function evaluations to converge, then for one high-fidelity step, 50 intermediate-fidelity evaluations will be required, and 2,500 lower-fidelity evaluations will be required. If the number of optimization iterations is of the same order at each level, then there will be an exponential scaling in the number of function evaluations required between fidelity levels.

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.[20,21] In the case of multiple fidelity levels, the calibration of $f_{\text{high}}(\mathbf{x}) \approx f_{\text{low}}(\mathbf{x}) + e(\mathbf{x})$ is modified so that the error model, $e(\mathbf{x})$ is treated as the mean of a Gaussian process, and the error model also includes a variance model. In this case, the error model, normally distributed with mean $\mu(\mathbf{x})$ and variance $\sigma^2(\mathbf{x})$, is written as $\mathcal{N}\left(\mu(\mathbf{x}), \sigma^2(\mathbf{x})\right)$. In the case of multiple lower-fidelity functions there are multiple estimates of the high-fidelity function, for example,

$$f_{\text{high}}(\mathbf{x}) \approx f_{\text{med}}(\mathbf{x}) + \mathcal{N}\left(\mu_{\text{med}}(\mathbf{x}), \sigma^2_{\text{med}}(\mathbf{x})\right) \tag{24}$$
$$f_{\text{high}}(\mathbf{x}) \approx f_{\text{low}}(\mathbf{x}) + \mathcal{N}\left(\mu_{\text{low}}(\mathbf{x}), \sigma^2_{\text{low}}(\mathbf{x})\right). \tag{25}$$

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 $f_{\text{est}}$, given by

$$f_{\text{est}}(\mathbf{x}) = \left(f_{\text{med}}(\mathbf{x}) + \mu_{\text{med}}(\mathbf{x})\right)\left[\frac{\sigma^2_{\text{low}}(\mathbf{x})}{\sigma^2_{\text{low}}(\mathbf{x}) + \sigma^2_{\text{med}}(\mathbf{x})}\right] + \left(f_{\text{low}}(\mathbf{x}) + \mu_{\text{low}}(\mathbf{x})\right)\left[\frac{\sigma^2_{\text{med}}(\mathbf{x})}{\sigma^2_{\text{low}}(\mathbf{x}) + \sigma^2_{\text{med}}(\mathbf{x})}\right] \tag{26}$$

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

$$\frac{1}{\sigma^2_{\text{est}}(\mathbf{x})} = \frac{1}{\sigma^2_{\text{low}}(\mathbf{x})} + \frac{1}{\sigma^2_{\text{med}}(\mathbf{x})}. \tag{27}$$

A schematic of the behavior of this maximum likelihood estimate is shown in Figure 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 non-calibrated point.

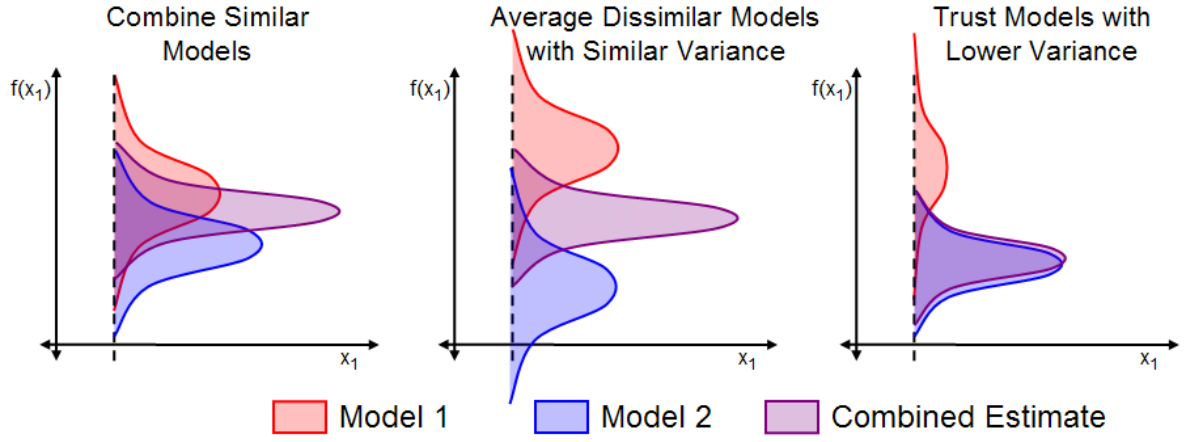



**Figure 6. Behavior of the combined maximum likelihood estimate given the behavior of the individual estimates.**

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 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, so that prediction is trusted implicitly and the other lower-fidelity information is not used. Also, with a smooth Gaussian process covariance function, the variance estimate will be a smooth function. This makes the model of the high-fidelity function, $f_{\text{est}}(\mathbf{x})$, a smooth model that satisfies the optimization algorithm requirements for convergence to a high-fidelity optimum. Therefore, the user may choose any method of selecting which lower-fidelity models are calibrated at a required calibration point, as only one needs to be. 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.

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 4. 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.

| 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 | 50455 |

**Table 4. 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.**

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 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 non-smooth. The lack of smoothness violates the convergence criteria of this method.

| Method | Cart3D | Shock-expansion | Panel Method |
|---|---|---|---|
| Two-Fidelities | 88 | 0 | 47679 |
| Max. Likelihood | 66 | 23297 | 23297 |
| Nested | 66* | 7920* | 167644* |

**Table 5. Number of function calls required to optimize an airfoil for minimum drag using the Euler equations (Cart3D) with multiple lower-fidelity models. An asterisk indicates that solution was not converged due to numerical limitations.**

The final example demonstrates that the maximum likelihood approach can still benefit from a poor low-fidelity model. The results in Table 6 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 to 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.

| Method | Shock-expansion | Panel Method | Camberline |
|---|---|---|---|
| Two-Fidelities | 126 | 43665 | 0 |
| Max. Likelihood | 84 | 30057 | 30057 |
| Nested | 212* | 59217* | 342916* |

**Table 6. Number of function calls required to optimize an airfoil for minimum drag using shock-expansion theory with multiple lower-fidelity models. An asterisk indicates a minimum number of function evaluations as opposed to an average value from random starting points.**

## VII. Conclusion

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, a downfall of the method is that the 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. 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.

## Acknowledgements

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.

## References

[1]T. W. Simpson, J. D. Peplinski, P. N. Koch, and J. K. Allen, "Metamodels for Computer-based Engineering Design: Survey and recommendations," *Engineering with Computers*, Vol. 17, 2001, pp. 129150.

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

[3]M. Eldred, S. Giunta, and S. Collis, "Second-order corrections for surrogate-based optimization with model hierarchies," In Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2004.

[4]A. C. Antoulas and D.C. Sorensen, *Approximation of Large-Scale Dynamical Systems*, SIAM, 2005.

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

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

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

[8]J. Castro, G. Gray, A. Giunta, and P. Hough, "Developing a Computationally Efficient Dynamic Multilevel Hybrid Optimization Scheme using Multifidelity Model Interactions," Tech. Rep. SAND2005-7498, Sandia, November 2005.

[9]D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, Vol. 13, 1998, pp. 455–492.

[10]M. Kennedy and A. O'Hagan, "Bayesian Calibration of Computer Models," *Journal of the Royal Statistical Society*, Vol. 63, No. 2, 2001, pp. 425–464.

[11]M. Kennedy and A. O'Hagan, "Predicting the Output From a Complex Computer Code When Fast Approximations Are Available," *Biometrika*, Vol. 87, No. 1, 2000, pp. 1–13.

[12]S. Leary, A. Bhaskar, and A. Keane, "A Knowledge-Based Approach to Response Surface Modelling in Multifidelity Optimization," *Journal of Global Optimization*, Vol. 26, 2003, pp. 297–319.

[13]Carter, R. G., "On the Global Convergence of Trust Region Algorithms Using Inexact Gradient Information," *SIAM Journal of Numerical Analysis*, Vol. 28, No. 1, 1991, pp. 251–265.

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

[15]A. R. Conn, K. Scheinberg, and L. Vicente, "Geometry of Interpolation Sets in Derivative Free Optimization," *Mathematical Programming*, Vol. 111, No. 1-2, 2008, pp. 141–172.

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

[17]S. M. Wild, R. G. Regis, and C. A. Shoemaker, "ORBIT: Optimization by Radial Basis Function Interpolation in Trust-Regions," *SIAM Journal of Scientific Computing*, Vol. 30, No. 6, 2008, pp. 3197–3219.

[18]Wild, S. M., *Derivative-Free Optimization Algorithms for Computationally Expensive Functions*, Ph.D. thesis, Cornell University, January 2009.

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

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

[21]C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, The MIT Press, 2006.

[22]M. Nemec, M. Aftosmis, and M. Wintzer, "Adjoint-Based Adaptive Mesh Refinement for Complex Geometries," *46th AIAA Aerospace Sciences Meeting*, Vol. 2008-0725, Reno, NV, January 7-10 2008.