Convergent Multifidelity Optimization using Bayesian Model Calibration

Andrew March*, Karen Willcox†

Massachusetts Institute of Technology, Cambridge, Massachusetts, 02139.

Multifidelity optimization approaches seek to bring higher-fidelity analyses earlier into the design process by using performance estimates from lower-fidelity models to accelerate convergence towards the optimum of a high-fidelity design problem. Current multifidelity optimization methods generally fall into two broad categories: provably convergent methods that use either the high-fidelity gradient or a high-fidelity pattern-search, and heuristic model calibration approaches, such as interpolating high-fidelity data or adding a Kriging error model to a lower-fidelity function. This paper presents a multifidelity optimization method that combines these two ideas; our method iteratively calibrates lower-fidelity information to the high-fidelity function, and is provably convergent to an optimum of the high-fidelity design problem. The algorithm developed minimizes a high-fidelity objective function subject to a high-fidelity constraint and other simple constraints. The algorithm never computes the gradient of a high-fidelity function; however, it demonstrates convergence using sensitivity information from the calibrated low-fidelity models, which are constructed to have negligible error in a neighborhood around the solution. The method is demonstrated for aerodynamic shape optimization and shows an 80% reduction in the number of high-fidelity analyses compared with a single-fidelity sequential quadratic programming formulation and a similar number of high-fidelity analyses compared with a multifidelity trust-region algorithm that estimates the high-fidelity gradient using finite differences.

I. Introduction

Complex systems are expensive to build and test, so system designers will instead analyze these systems with the best computational models available. These high-fidelity methods can be computationally expensive and finding optimal or even good designs using them is a daunting task. Typically designers use lower-fidelity analysis methods to design the system and then verify the performance with the high-fidelity methods. The drawback of this approach is that designing with low-fidelity models can lead to poor design decisions. Multifidelity approaches offer a solution to this problem; they can can guarantee a design that is better with respect to high-fidelity model estimates, but use low-fidelity analyses to reduce the required number of high-fidelity evaluations. This reduction is commonly achieved by calibrating low-fidelity analysis methods to high-fidelity analysis results by either adding an error model, scaling or shifting the output, or mapping the design space. However, a complication with some of these calibration methods is that high-fidelity analyses may underly both design objective functions and design constraints. In order for the high-fidelity design to be improved, feasibility must be assured; any error in the calibration method for the constraints may improve the high-fidelity design objective but produce an infeasible design. A multifidelity optimization method should therefore be able to guarantee that high-fidelity performance will be improved, by using an appropriate model calibration method to safeguard against constraint violation and reduce the high-fidelity evaluations of both constraints and objectives.

A powerful technique commonly used for this multifidelity optimization problem is trust-region model management, where at all design iterates the gradients of low-fidelity objective function and constraints are either scaled or shifted such that they match the value and gradient of their high-fidelity counterpart.[1,2] This technique is provably convergent to a locally optimal high-fidelity design and in practice provides a 2–3

*Graduate Student, Department of Aeronautics and Astronautics, MIT

†Associate Professor, Department of Aeronautics and Astronautics, MIT

fold reduction in the number of high-fidelity function calls.[1] However, it is common in engineering design problems that the gradient of a high-fidelity function is unavailable and estimating the gradient is unreliable or too expensive. In these cases, multifidelity methods such as Efficient Global Optimization (EGO)[3] or Surrogate Management Framework (SMF)[4] can be used. 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, whereas, 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 made globally convergent,[5] can be used in a multifidelity setting using Bayesian model calibration methods,[6–8] 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[5] and to the exact metric of selecting points likely to improve the high-fidelity function value;[9] moreover, methods to handle constraints are still somewhat heuristic.[5,9,10] 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 value is known.[11] In SMF, constraints may be handled directly on the surrogate models either with gradient information or with penalties, and for the gradient-free high-fidelity pattern-search can be handled with an augmented Lagrangian,[12–14] exact penalty method,[15] or constraint filtering.[16]

In many cases, sensitivity information is very useful and can accelerate convergence of a derivative-based optimization method towards a local optimum. Accordingly, for multifidelity optimization of a function where the gradient is known, trust-region model management is likely the best method to find a locally optimal design. However, if high-fidelity gradient information is unknown, trust-region model management requires finite differences to produce a surrogate model that satisfies first-order requirements at each design iterate. Finite differences may not always be a viable choice due to noise in the function evaluations, number of design variables, design locations where the objective function fails to exist, or simply computational requirements. In these situations gradient-free multifidelity techniques such as EGO or SMF could be considered, since they converge quickly if the model calibration methods are robust. This paper poses the question: can a low-fidelity surrogate model be calibrated in a sufficiently robust manner as to use low-fidelity sensitivity information to optimize a high-fidelity system with a limited number of high-fidelity evaluations? The answer to this question requires finding a surrogate modeling technique that can capture enough high-fidelity behavior to prove convergence to a high-fidelity optimum, using as few high-fidelity function evaluations as a finite difference based calibration approach.

Creating a fully linear model is one possible surrogate modeling technique to ensure low-fidelity sensitivity information predicts high-fidelity sensitivities well. A fully linear model, formally defined in the next section, establishes Lipschitz-type error bounds between the high-fidelity function and the surrogate model. This ensures that the error between the high-fidelity gradient and surrogate model gradient is locally bounded without ever calculating the high-fidelity gradient. Conn *et al.* showed polynomial interpolation models can be made fully linear, provided the interpolating set satisfied certain geometric requirements,[17] and further developed an unconstrained gradient-free optimization technique using fully linear models.[18,19] Wild *et al.* demonstrated that a radial basis function interpolation could satisfy the requirements for a fully linear model and be used in Conn's derivative-free optimization framework.[20–22] March and Willcox generalized this method to the cases with arbitrary low-fidelity functions or multiple low-fidelity functions using Bayesian model calibration methods typically associated with EGO.[23] This paper demonstrates how fully linear models can be used for constrained optimization of computationally expensive functions when their derivatives are not available. The constraints are partitioned into constraints with and without available derivatives. The constraints without available derivatives are approximated with multifidelity methods; whereas the other constraints are handled either implicitly with a penalty method or explicitly. Two constrained multifidelity methods are presented, the first optimizes a high-fidelity objective function subject to constraints with available derivatives. The second formulation optimizes a high-fidelity objective function subject to high-fidelity constraints and constraints with available derivatives.

Section II of this paper presents the derivative-free method to optimize a high-fidelity objective function

subject to constraints with available derivatives. Fully linear surrogate models of the objective function are minimized within a trust-region setting until convergence to a high-fidelity optimum is demonstrated. Section III presents a technique for minimizing a high-fidelity objective function subject to both constraints with available derivatives and computationally expensive constraints with unavailable derivatives. This algorithm finds a feasible point using the first formulation to reduce the value of the high-fidelity constraint subject to the constraints with available derivatives. After a feasible point is found, the high-fidelity objective is minimized using fully linear surrogate models for both the high-fidelity objective and high-fidelity constraint. Only points that maintain feasibility are accepted. Section IV presents an aerodynamic shape optimization problem to demonstrate the proposed multifidelity optimization techniques and compares the number of high-fidelity function evaluations with other methods. The conclusion in Section V discusses extensions of the method to the cases when there are multiple lower-fidelity models or when constraints are hard (when the objective function fails to exist if the constraints are violated).

II. Constrained Optimization of a Multifidelity Objective Function

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 a constrained optimization problem without ever directly computing or estimating the gradient of the high-fidelity function, but generating sensitivity information from the low-fidelity function to reduce the required number of high-fidelity evaluations. The minimization problem considered is to minimize the high-fidelity objective function subject to equality constraints, $h(\mathbf{x})$, and inequality constraints $g(\mathbf{x})$,

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & f_{\text{high}}(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{h}(\mathbf{x}) = 0 \\ & \mathbf{g}(\mathbf{x}) \leq 0, \end{aligned} \tag{1}$$

where we assume gradient information from $h(\mathbf{x})$ and $g(\mathbf{x})$ is available or can be estimated accurately. To establish convergence it is necessary to assume that all of the functions in Eq. 1 and the low-fidelity models are twice continuously differentiable, are Lipschitz continuous, and have Lipschitz continuous first-derivatives. In addition we will assume that the solution to Eq. 1, $\mathbf{x}^*$, is feasible and is a regular point (satisfies strict linear independent constraint qualification).

II.A. Trust-region Model Management

From an initial design vector $\mathbf{x}_0$, the trust-region method generates a sequence of design vectors that each reduce a merit function consisting of the high-fidelity function value and penalized constraint violation, where we denote $\mathbf{x}_k$ to be this design vector on the kth trust-region iteration. Following the general Bayesian calibration approach in Ref. 7, 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,

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

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

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

To solve the constrained optimization problem presented in Eq. 1 we define a merit function, $\Phi(\mathbf{x}_k, \sigma_k)$, where σ_k is a parameter that must go to infinity as the iteration number k goes to infinity and serves to increase the penalty placed on the constraint violation. We also must assume that the merit function and the

trust-region iterates satisfy the following four properties. First, the merit function with the initial penalty, σ_0, must be bounded from below within a relaxed level-set, $\mathcal{L}(\mathbf{x}_0, \sigma_0)$, defined as

$$\begin{aligned} L(\mathbf{x}_0, \sigma_0) &= \{\mathbf{x} \in \mathbb{R}^n : \Phi(\mathbf{x}, \sigma_0) \leq \Phi(\mathbf{x}_0, \sigma_0)\} && (5) \\ B(\mathbf{x}_k) &= \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x} - \mathbf{x}_k\| \leq \Delta_{\max}\} && (6) \\ \mathcal{L}(\mathbf{x}_0, \sigma_0) &= L(\mathbf{x}_0, \sigma_0) \bigcup_{\mathbf{x_k} \in L(\mathbf{x}_0, \sigma_0)} B(\mathbf{x}_k), && (7) \end{aligned}$$

where $\Delta_{\max}$ is the maximum allowable trust-region size and 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 $\mathbf{x}_0$. Second, the level sets of $\Phi(\mathbf{x}_k, \sigma_k > \sigma_0)$ must be contained within $L(\mathbf{x}_0, \sigma_0)$. Third, $L(\mathbf{x}_0, \sigma_0)$ must be a compact set, and fourth, all design iterates $\mathbf{x}_k$ must remain within $L(\mathbf{x}_0, \sigma_0)$.

Although other merit functions are possible, we restrict our attention to merit functions based on quadratic penalty functions because it is trivial to show that they are bounded from below if the objective function obtains a finite global minimum. The merit function used in this method is the objective function plus the scaled sum-squares of the constraint violation, where $\mathbf{g}^+(\mathbf{x})$ are the values of the violated inequality constraints,

$$\Phi(\mathbf{x}, \sigma_k) = f_{\text{high}}(\mathbf{x}) + \frac{\sigma_k}{2}\mathbf{h}(\mathbf{x})^T\mathbf{h}(\mathbf{x}) + \frac{\sigma_k}{2}\mathbf{g}^+(\mathbf{x})^T\mathbf{g}^+(\mathbf{x}). \quad (8)$$

The parameter σ_k is a penalty weight, which must go to $+\infty$ as the iteration k goes to $+\infty$. Note that when using a quadratic penalty function for constrained optimization, the sequence of iterates generated, $\{\mathbf{x}_k\}$, can either terminate at a feasible regular point at which the Karush-Kuhn-Tucker (KKT) conditions are satisfied, or at a point that minimizes the squared norm of the constraint violation, $\mathbf{h}(\mathbf{x})^T\mathbf{h}(\mathbf{x}) + \mathbf{g}^+(\mathbf{x})^T\mathbf{g}^+(\mathbf{x})$.[24,25] Accordingly, we assume that the iterates generated by minimizing the merit function are such that the sequence $\{\mathbf{x}_k\}$ always remains within $L(\mathbf{x}_0, \sigma_0)$ and terminates at a limit point, $\mathbf{x}^*$ that is both regular and feasible.

We now define a surrogate merit function, $\hat{\Phi}(\mathbf{x}, \sigma_k)$, which replaces the objective function with a surrogate model of the expensive objective function.

$$\hat{\Phi}(\mathbf{x}, \sigma_k) = m_k(\mathbf{x}) + \frac{\sigma_k}{2}\mathbf{h}(\mathbf{x})^T\mathbf{h}(\mathbf{x}) + \frac{\sigma_k}{2}\mathbf{g}^+(\mathbf{x})^T\mathbf{g}^+(\mathbf{x}). \quad (9)$$

Optimization is performed on this function, and updates to the trust-region are based on how changes in this surrogate merit function compare with changes in the original merit function, $\Phi(\mathbf{x}, \sigma_k)$.

We further require 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.*:

Definition 1. *Let a function $f_{high}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ that is continuously differentiable and has a Lipschitz continuous derivative, 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 κ_f, κ_g and κ_{blg} such that for any $\mathbf{x} \in L(\mathbf{x}_0, \sigma_0)$ 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

$$\|\nabla f_{high}(\mathbf{x}) - \nabla m_k(\mathbf{x})\| \leq \kappa_g \Delta_k \quad \forall \mathbf{x} \in \mathcal{B}_k \quad (10)$$

and the error between the model and the function satisfies

$$|f_{high}(\mathbf{x}) - m_k(\mathbf{x})| \leq \kappa_f \Delta_k^2 \quad \forall \mathbf{x} \in \mathcal{B}_k. \quad (11)$$

Such a model $m_k(\mathbf{x})$ is called fully linear on $\mathcal{B}_k$.[19]

II.B. Trust-region Subproblem

At each trust-region iteration a point likely to decrease the merit function is found by solving one of two minimization problems on the fully linear model for a step $\mathbf{s}_k$, on a trust region of size Δ_k:

$$\begin{aligned} \min_{\mathbf{s}_k \in \mathbb{R}^n} \quad & m_k(\mathbf{x}_k + \mathbf{s}_k) \\ \text{s.t.} \quad & \mathbf{h}(\mathbf{x}_k + \mathbf{s}_k) = 0 \\ & \mathbf{g}(\mathbf{x}_k + \mathbf{s}_k) \leq 0 \\ & \|\mathbf{s}_k\| \leq \Delta_k, \end{aligned} \tag{12}$$

or

$$\begin{aligned} \min_{\mathbf{s}_k \in \mathbb{R}^n} \quad & \hat{\Phi}_k(\mathbf{x}_k + \mathbf{s}_k, \sigma_k) \\ \text{s.t.} \quad & \|\mathbf{s}_k\| \leq \Delta_k. \end{aligned} \tag{13}$$

Ideally the first trust-region subproblem, Eq. 12, will be used for optimization; however, typical nonlinear programming techniques do not allow a user to specify which constraints must be satisfied, so if some of the constraints are violated it may not be possible to reduce the value of the merit function and guarantee that the trust-region constraint is satisfied. Accordingly, when the current iterate is infeasible, the second trust-region subproblem, Eq. 13, must be used. A problem with this subproblem is that the norm of the objective function Hessian grows without bound due to the penalty increasing to infinity. Therefore, a fundamental assumption in trust-region algorithms, that the subproblem have a bounded norm for the Hessian, can become violated. So, the algorithm must be structured in a way that once the design iterate is "close" to feasible, the subproblem in Eq. 12 must be used, since the Hessian of this subproblem remains bounded.

We require that for whichever trust-region subproblem is used, the subproblem must be solved such that the 2-norm of the first-order optimality conditions is less than a constant τ_k. This requirement is stated as $\|\nabla_x \mathfrak{L}_k\| \leq \tau_k$, where $\mathfrak{L}_k$ is the Lagrangian for the trust-region subproblem used. There are two requirements for τ_k. First $\tau_k < \epsilon$, where ϵ is the desired termination tolerance for the optimization problem in Eq. 1. Second, τ_k must decrease to zero as the number of iterations goes to infinity. Accordingly, we define $\tau_k = \min[\beta\epsilon, c\Delta_k]$, with a constant $\beta \in (0,1)$ to satisfy the overall tolerance criteria, and a constant $c \in (a,1)$ multiplying Δ_k to assure that τ_k goes to zero. The constant a will be defined as part of a sufficient decrease condition that forces the size of the trust-region to decrease to zero in the next section.

II.C. Trust-region Updating

Without using the high-fidelity function gradient, the trust-region update scheme must ensure the size of the trust-region decreases to zero to establish convergence. To do this, we use a requirement similar to the fraction of Cauchy decrease requirement in an the unconstrained trust-region formulation, see for example Ref. 19. We require that the improvement in our merit function is at least a small constant a, $0 < a \leq \epsilon$, multiplying Δ_k,

$$\hat{\Phi}(\mathbf{x}_k, \sigma_k) - \hat{\Phi}(\mathbf{x}_k + \mathbf{s}_k, \sigma_k) \geq a\Delta_k. \tag{14}$$

The sufficient decrease condition is enforced through the trust region update parameter, ρ_k. The update parameter is the ratio of the actual reduction in the merit function to the predicted reduction in the merit function unless the sufficient decrease condition is not met,

$$\rho_k = \begin{cases} 0 & \hat{\Phi}(\mathbf{x}_k, \sigma_k) - \hat{\Phi}(\mathbf{x}_k + \mathbf{s}_k, \sigma_k) < a\Delta_k \\ \frac{\Phi(\mathbf{x}_k, \sigma_k) - \Phi(\mathbf{x}_k + \mathbf{s}_k, \sigma_k)}{\hat{\Phi}(\mathbf{x}_k, \sigma_k) - \hat{\Phi}(\mathbf{x}_k + \mathbf{s}_k, \sigma_k)} & \text{otherwise.} \end{cases} \tag{15}$$

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, kept constant if the prediction is fair, and the trust region is contracted if the model predicts the change poorly. Specifically, we update the trust region size using

$$\Delta_{k+1} = \begin{cases} \min\{\gamma_1 \Delta_k, \Delta_{\max}\} & \text{if } \eta_1 \leq \rho_k \leq \eta_2, \\ \gamma_0 \Delta_k & \text{if } \rho_k \leq \eta_0, \\ \Delta_k & \text{otherwise,} \end{cases} \tag{16}$$

where $0 < \eta_0 < \eta_1 < 1 < \eta_2$, $0 < \gamma_0 < 1$, and $\gamma_1 > 1$. Regardless of whether or not a sufficient decrease has been found, the trust-region location will be updated if the trial point has decreased the value of the merit function,

$$\mathbf{x}_{k+1} = \begin{cases} \mathbf{x}_k + \mathbf{s}_k & \text{if } \Phi(\mathbf{x}_k, \sigma_k) > \Phi(\mathbf{x}_k + \mathbf{s}_k, \sigma_k) \\ \mathbf{x}_k & \text{otherwise.} \end{cases} \tag{17}$$

A new surrogate model, $m_{k+1}(\mathbf{x})$, is then built such that it is fully linear on a region $\mathcal{B}_{k+1}$ having center $\mathbf{x}_{k+1}$ and size Δ_{k+1}. The new fully linear model is constructed using the procedure of Wild *et al.*[20] with the calibration technique of March and Willcox.[23]

II.D. Termination

For termination, we must establish that the first-order KKT conditions,

$$\|\nabla f_{\text{high}}(\mathbf{x}_k) + A(\mathbf{x}_k)^T \lambda(\mathbf{x}_k)\| \leq \epsilon, \tag{18}$$

$$\| \left[\mathbf{h}(\mathbf{x}_k), \mathbf{g}^+(\mathbf{x}_k)\right] \| \leq \epsilon \tag{19}$$

are satisfied at $\mathbf{x}_k$, where $A(\mathbf{x}_k)$ is defined to be the Jacobian of all active constraints at that point,

$$A(\mathbf{x}_k) = \left[\nabla \mathbf{h}(\mathbf{x}_k), \nabla \mathbf{g}^+(\mathbf{x}_k)\right]^T. \tag{20}$$

The constraint violation criteria, Eq. 19, can be evaluated directly. However, the first-order condition, Eq. 18, cannot be verified directly in the derivative-free case because the gradient $\nabla f_{\text{high}}(\mathbf{x}_k)$ is unknown. Therefore, we must use the property of a fully linear model given in Eq. 10, which shows that as $\Delta_k \to 0$, then $\|\nabla f_{\text{high}}(\mathbf{x}) - \nabla m_k(\mathbf{x})\| \to 0$. We know that when the constraint violation termination condition is satisfied the trust-region problem in Eq. 12 can be solved. Therefore if the trust-region constraint is inactive, we know that

$$\|\nabla m(\mathbf{x}_k) + A(\mathbf{x}_k)^T \hat{\lambda}(\mathbf{x}_k)\| \leq \min\left[\beta\epsilon, c\Delta_k\right], \tag{21}$$

where $\hat{\lambda}$ are the appropriate Lagrange multipliers computed using the surrogate model as opposed to the high-fidelity function. Therefore a Δ_k sufficiently small, for example $\Delta_k \leq \epsilon_2$ for a small constant ϵ_2 is sufficient to show $\|\nabla f_{\text{high}}(\mathbf{x}) - \nabla m_k(\mathbf{x})\| \approx 0$, $\|\hat{\lambda} - \lambda\| \approx 0$, and $\|\nabla f_{\text{high}}(\mathbf{x}_k) + A(\mathbf{x}_k)^T \hat{\lambda}(\mathbf{x}_k)\| \leq \epsilon$.

II.E. Convergence Discussion

To demonstrate convergence of this algorithm we must first show that the trust-region subproblem that handles the constraints explicitly, Eq. 12, can be solved in a finite number of iterations, secondly that the size of the trust-region must go to zero, and finally that the algorithm cannot stop before the KKT conditions are satisfied for the original optimization problem, Eq. 1. To demonstrate that the trust-region subproblem handling the constraints directly can be solved we must show that a feasible point (or point feasible to within numerical tolerance) must exist within a trust region that will be encountered by solving the surrogate penalty trust-region subproblem, Eq. 13.

We begin by establishing a bound on the constraint violation in terms of the penalty parameter σ_k. We use the definition of $A^+(\mathbf{x})$ from Conn *et al.*,[26]

$$A^+(\mathbf{x}) = A(\mathbf{x})^T (A(\mathbf{x})A(\mathbf{x})^T)^{-1}, \tag{22}$$

so that least square Lagrange multipliers λ which approximate,

$$\nabla f_{\text{high}}(\mathbf{x}) + A(\mathbf{x})^T \lambda = 0, \tag{23}$$

can be found by the matrix-vector product,

$$\lambda(\mathbf{x}) = -A^+(\mathbf{x})^T \nabla f_{\text{high}}(\mathbf{x}). \tag{24}$$

Given the assumption that minimizing a quadratic penalty function terminates at a point that is both feasible and regular, Conn *et al.* shows that if the objective function is used exactly, then there exists an iteration k after which all $A^+(\mathbf{x}_k)$ will exist, be bounded such that $\|A^+(\mathbf{x}_k)\| \leq \kappa_1$ for a positive constant κ_1, and converge to $A^+(\mathbf{x}^*)$.[26] Conn *et al.* then establishes the bound for the constraint violation,

$$\| \left[\mathbf{h}(\mathbf{x}_k)^T \ \mathbf{g}^+(\mathbf{x}_k)^T\right] \| \leq \frac{\kappa_1\omega_k + \|\lambda(\mathbf{x}^*)\| + \kappa_2\|\mathbf{x}_k - \mathbf{x}^*\|}{\sigma_k}, \tag{25}$$

where κ_2 is a positive constant and $\|\nabla\Phi(\mathbf{x}_k,\sigma_k)\| = \|\nabla f_{\text{high}}(\mathbf{x}_k) + \sigma_k A(\mathbf{x}_k)^T \left[\mathbf{h}(\mathbf{x}_k)^T \ \mathbf{g}^+(\mathbf{x}_k)^T\right]^T \| \leq \omega_k$, for a convergence tolerance ω_k.[26] Using the surrogate model, for any solution to Eq. 13 where the trust-region is an inactive constraint, we have that, $\|\nabla\hat{\Phi}(\mathbf{x}_k,\sigma_k)\| = \|\nabla m_k(\mathbf{x}_k) + \sigma_k A(\mathbf{x}_k)^T \left[\mathbf{h}(\mathbf{x}_k)^T \ \mathbf{g}^+(\mathbf{x}_k)^T\right]^T \| \leq \tau_k$. Accordingly, when the trust-region constraint is inactive, using the definitions of τ_k and a fully linear model we have that,

$$\| \left[\mathbf{h}(\mathbf{x}_k)^T \ \mathbf{g}^+(\mathbf{x}_k)^T\right] \| \leq \frac{\kappa_1(c+\kappa_g)\Delta_k + \|\lambda(\mathbf{x}^*)\| + \kappa_2\|\mathbf{x}_k - \mathbf{x}^*\|}{\sigma_k}. \tag{26}$$

Now, using the fact that all $\mathbf{x}_k$ lie within the compact set $L(\mathbf{x}_0,\sigma_0)$, and that $\mathbf{x}^*$ is a regular point we can say that there exists a finite constant that bounds the numerator. Therefore by increasing the penalty parameter σ_k to infinity we may force the constraint violation to zero. Moreover, if we examine the constraint violation bound of Eq. 26 further, we see that the bound has the form $\kappa_3\Delta_k/\sigma_k + \kappa_4/\sigma_k$, with κ_3, κ_4 being arbitrary positive constants. It will be shown shortly that $\{\Delta_k\}$ converges to zero, therefore we must have that $\{\sigma_k\}$ increases to infinity strictly faster than $\{\Delta_k\}$ converges to zero, or that the series $\sigma_k\Delta_k$ diverges, in order to mitigate the term κ_4/σ_k. This criteria is enforced by construction and ensures that an iteration k exists such that a feasible point will reside in the interior of a trust region of size Δ_k and the subproblem using the constraints explicitly, Eq. 12, can be solved.

We now use our sufficient decrease condition, that $\rho_k = 0$ unless $\hat{\Phi}(\mathbf{x}_k,\sigma_k) - \hat{\Phi}(\mathbf{x}_k + \mathbf{s}_k,\sigma_k) \geq a\Delta_k$, to establish the proposition,

$$\lim_{k\to+\infty} \Delta_k = 0. \tag{27}$$

From the sufficient decrease condition, we know that the trust region size decreases unless the change in the merit function $\Phi(\mathbf{x}_k,\sigma_k) - \Phi(\mathbf{x}_k + \mathbf{s}_k,\sigma_k) \geq \eta_0 a\Delta_k$. Therefore we must show that the total number of times in which the size of the trust region is kept constant or increased must be bounded. To demonstrate this, we have assumed a priori that the merit function is bounded from below and also that the trust-region algorithm remains within a level-set of the function from where it initiated. Let us now consider the merit function written in an alternate form,

$$\Phi(\mathbf{x}_k,\sigma_k) = f_{\text{high}}(\mathbf{x}_k) + \frac{\sigma_k}{2}\| \left[\mathbf{h}(\mathbf{x}_k), \mathbf{g}^+(\mathbf{x}_k)\right] \|^2. \tag{28}$$

From Eq. 26 we know that if σ_k is large enough such that the bound on the constraint violation is less than unity, then we have established that at each subsequent iteration,

$$\| \left[\mathbf{h}(\mathbf{x}_k)^T \ \mathbf{g}^+(\mathbf{x}_k)^T\right] \|^2 \leq \left[\frac{\kappa_1(c+\kappa_g)\Delta_k + \|\lambda^*\| + \kappa_2\|\mathbf{x}_k - \mathbf{x}^*\|}{\sigma_k}\right]^2. \tag{29}$$

Now combining Eqs. 28 and 29, we can show that at an iteration k an upper bound in the total remaining change in the merit function is

$$f_{\text{high}}(\mathbf{x}_k) - \min_{\mathbf{x}\in L(\mathbf{x}_0,\sigma_0)} f_{\text{high}}(\mathbf{x}) + \frac{\left[\kappa_1(c+\kappa_g)\Delta_k + \|\lambda(\mathbf{x}^*)\| + \kappa_2\|\mathbf{x}_k - \mathbf{x}^*\|\right]^2}{\sigma_k}. \tag{30}$$

Given that that Δ_k is always bounded from above by $\Delta_{\max}$, $\|\lambda(\mathbf{x}^*)\|$ is bounded because $\mathbf{x}^*$ is a regular point, and $\|\mathbf{x}_k - \mathbf{x}^*\|$ is bounded because we've assumed $L(\mathbf{x}_0,\sigma_0)$ is a compact set. Therefore, if the series $\{1/\sigma_k\}$ has a finite sum, then the total improvement in the merit function is finite. Accordingly, the sum of the series $\{\Delta_k\}$ must be finite, and $\Delta_k \to 0$ as $k \to \infty$.

We have currently placed two restrictions on the series $\{\sigma_k\}$, that it grow faster than $1/\Delta_k$ and that $\{1/\sigma_k\}$ has a finite sum. A third restriction comes from the fact that we may not grow σ_k arbitrarily quickly

for two reasons, first the solution $\mathbf{x}_{k+1}$ can move significantly far from $\mathbf{x}_k$ such as to cause problems in the trust region framework and secondly, the error between the surrogate model and high-fidelity function may cause arbitrarily poor estimates for $\mathbf{x}_{k+1}$. To place a conservative upper bound on the growth rate for the series $\{\sigma_k\}$ we compare the value of the merit function at a point $\Phi(\mathbf{x}_k + \mathbf{p}, \sigma_k)$ with its linearized prediction based on $\Phi(\mathbf{x}_k, \sigma_k)$. From the mean value theorem, we know there exists a $t \in (0, 1)$ such that,

$$\Phi(\mathbf{x}_k + \mathbf{p}, \sigma_k) = \Phi(\mathbf{x}_k, \sigma_k) + \nabla\Phi(\mathbf{x}_k + t\mathbf{p}, \sigma_k)^T \mathbf{p}. \tag{31}$$

We may also create the linearized prediction of $\Phi(\mathbf{x}_k + \mathbf{p}, \sigma_k)$, $\tilde{\Phi}(\mathbf{x}_k + \mathbf{p}, \sigma_k)$ as,

$$\tilde{\Phi}(\mathbf{x}_k + \mathbf{p}, \sigma_k) = \Phi(\mathbf{x}_k, \sigma_k) + \nabla\Phi(\mathbf{x}_k, \sigma_k)^T \mathbf{p}, \tag{32}$$

which provides us the bound,

$$\|\Phi(\mathbf{x}_k + \mathbf{p}, \sigma_k) - \tilde{\Phi}(\mathbf{x}_k + \mathbf{p}, \sigma_k)\| \leq \|\nabla\Phi(\mathbf{x}_k + t\mathbf{p}, \sigma_k) - \nabla\Phi(\mathbf{x}_k, \sigma_k)\|\|\mathbf{p}\|. \tag{33}$$

If κ_{fg} is the Lipschitz constant for $\nabla f_{\text{high}}(\mathbf{x})$, κ_c is the maximum Lipschitz constant for the constraints, and κ_{cd} is the maximum Lipschitz constant for a constraint gradient we can show that

$$\begin{aligned}\|\Phi(\mathbf{x}_k + \mathbf{p}, \sigma_k) - \tilde{\Phi}(\mathbf{x}_k + \mathbf{p}, \sigma_k)\| \leq \\ \left[\kappa_{fg} + \sigma_k \left(\kappa_c \| \left[\mathbf{h}(\mathbf{x}_k)^T \ \mathbf{g}^+(\mathbf{x}_k)^T\right] \| + \kappa_{cd}\|A(\mathbf{x}_k)\|\|\mathbf{p}\| + \kappa_c\kappa_{cd}\|\mathbf{p}\|^2\right)\right] \|\mathbf{p}\|^2.\end{aligned} \tag{34}$$

We have a similar result for $\hat{\Phi}(\mathbf{x}, \sigma_k)$ by replacing κ_{fg} with the sum $\kappa_{fg} + \kappa_g$ using the definition of a fully linear model, and by bounding $\|\mathbf{p}\|$ by Δ_k. Therefore, we may show the error in a linearized prediction of the surrogate model will go to zero provided that the series $\{\sigma_k\Delta_k^2\}$ converges to zero. $\{\Delta_k\}$ will converge R-linearly to zero with constant γ_0, therefore $\{\sigma_k\}$ must grow strictly slower than $\{1/\gamma_0^{2k}\}$. Accordingly, if the penalty parameter $\{\sigma_k\}$ grows at a rate exceeding both $\{k\}$ and $\{1/\Delta_k\}$ then $\{\Delta_k\}$ will converge to zero and an iterate $\mathbf{x}_k$ will exist for a finite k such that the trust-region problem handling the constraints explicitly will be feasible. In addition, if $\{\sigma_k\}$ grows slower than $\{1/\Delta_k^2\}$ then the error between the predicted values of the merit function and the actual values of the merit function will converge to zero within the trust region. It also suggests that a good choice for $\{\sigma_k\}$ has the form $\sigma_k = \max\left[e^{k/10}, 1/\Delta_k^{1.1}\right]$.

To demonstrate that the sequence of iterates generated by this algorithm terminates at a point $\mathbf{x}^*$ where the KKT conditions must be satisfied to within a tolerance of ϵ, we will assume for contradiction that the algorithm terminates at a regular and feasible point $\mathbf{x}^*$ at which the KKT conditions are not satisfied. We have established for these assumptions that the algorithm will always terminate by solving the trust-region subproblem in Eq. 12. If the KKT conditions are not satisfied, then as the trust-region size decreases to zero the KKT conditions will not be satisfied for the surrogate model. Therefore, the possible solutions to Eq. 12 are the current iterate or a point on the trust-region boundary. If the solution to the trust-region subproblem is the current iterate, then $\|\nabla m_k(\mathbf{x}_k)\| \leq \beta\epsilon$ because the error between the surrogate model and high-fidelity function will decrease to zero. This shows that $\|\nabla f_{\text{high}}(\mathbf{x})\| \leq \epsilon$ is a necessary contradiction. If the solution to Eq. 12 is always on the trust-region boundary and the trust-region continues to shrink, then the decrease in the surrogate model value must be less than $a\Delta_k$. For any sufficiently small trust-region size, the fully linear model ensures that the error between the surrogate model and high-fidelity function decreases to zero. Accordingly, for sufficiently small trust-region size, this shows that $\|\nabla f_{\text{high}}(\mathbf{x})\| \leq a$, and since $a < \epsilon$ then we have established a contradiction and the KKT conditions are satisfied to a tolerance of ϵ.

II.F. Implementation

The numerical implementation of the multifidelity optimization algorithm, which does not compute the gradient of the high-fidelity objective function, is presented as Algorithm 1. A set of possible parameters that may be used in this algorithm are listed in Table 2 in Section IV. A key element of this algorithm is the logic to switch from the penalty function trust-region subproblem, Eq. 13, to the subproblem that uses the constraints explicitly, Eq. 12. Handling the constraints exactly will generally lead to faster convergence and fewer function evaluations; however, a feasible solution to this subproblem likely does not exist at early iterations. If either the constraint violation is sufficiently small, $\| \left[\mathbf{h}(\mathbf{x}_k)^T \ \mathbf{g}^+(\mathbf{x}_k)^T\right] \| \leq \epsilon$, or the linearized steps, δ_h, satisfying $h(\mathbf{x}) + \nabla h(\mathbf{x})^T\delta_h = 0$ for all equality and inequality constraints are all smaller than the

size of the trust region, then the subproblem with the explicit constraints is attempted. If the optimization fails, then the penalty function subproblem is solved.

This method may be accelerated with the use of multiple lower-fidelity models. March and Willcox[23] suggest a multifidelity filtering technique to combine estimates from multiple low-fidelity functions into a single maximum likelihood estimate of the high-fidelity function value. That technique will work unmodified within this multifidelity optimization framework and will likely improve performance.

III. Multifidelity Objective and Constraint Optimization

We now consider a more general constrained optimization problem with a computationally expensive objective function and computationally expensive constraints. We assume for both the expensive objective and expensive constraints that the gradients are either unavailable, unreliable or expensive to estimate. Accordingly, we augment the multifidelity optimization problem in Eq. 1 with the high-fidelity constraint, $c_{\text{high}}(\mathbf{x}) \leq 0$. In addition, we have a low-fidelity estimate of this constraint, $c_{\text{low}}(\mathbf{x})$, which estimates the same metric as the high-fidelity constraint, but with unknown error. Therefore, our goal is to find the vector $\mathbf{x} \in \mathbb{R}^n$ of n design variables that solves the nonlinear constrained optimization problem,

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & f_{\text{high}}(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{h}(\mathbf{x}) = 0 \\ & \mathbf{g}(\mathbf{x}) \leq 0 \\ & c_{\text{high}}(\mathbf{x}) \leq 0, \end{aligned} \tag{35}$$

where $\mathbf{h}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ represent vectors of inexpensive equality and inequality constraints with derivatives that are either known or may be estimated cheaply. The same assumptions for the expensive objective function formulation are made for the functions presented in this formulation, including that a quadratic penalty function with the new high-fidelity constraint is bounded from below within an initial expanded level-set. A point of note is that multiple high-fidelity constraints can be used if an initial point $\mathbf{x}_0$ is given that is feasible with respect to all constraints; however, due to the effort required to construct approximations of the multiple high-fidelity constraints, it is recommended that all of the high-fidelity constraints be combined into a single high-fidelity constraint through, as an example, a discriminant function.[27,28] This optimization problem will be solved using the multifidelity optimization method presented in Section II to find a feasible point, and then an interior point formulation is presented in Section III.B to solve the optimization problem in Eq. 35. Convergence of the algorithm is discussed in Section III.C, and the numerical implementation is presented in Section III.D.

III.A. Finding a Feasible Point

This algorithm begins by finding a point that is feasible with respect to all of the constraints and then minimizes the high-fidelity objective function subject to all of the constraints in a manner similar to an interior-point method. In order to find an initial feasible point, the algorithm presented in Section II iterates on the optimization problem,

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & c_{\text{high}}(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{h}(\mathbf{x}) = 0 \\ & \mathbf{g}(\mathbf{x}) \leq 0, \end{aligned} \tag{36}$$

until a point that is feasible with respect to Eq. 35 is found. If this optimization problem is unconstrained then the trust-region algorithm of Conn *et al.*[19] is used with the multifidelity calibration method of March and Willcox.[23] The optimization problem in Eq. 36 may violate one of the assumptions of the multifidelity objective function method in that $c_{\text{high}}(\mathbf{x})$ may not be bounded from below. This issue will be addressed in the numerical implementation of the method in Section III.D.

III.B. Interior Point Trust-region Method

To minimize the high-fidelity objective function subject to the constraints, this formulation again uses the general Bayesian calibration approach in Ref. 7 to create fully linear surrogate models for both the high-

Algorithm 1: Multifidelity Objective Trust-Region Algorithm

1: Update tolerance, $\tau_k = \min[\beta\epsilon, c\Delta_k]$.

2: Choose and solve a trust-region subproblem:

2a: If the maximum linearized step to constraint feasibility for all active constraints is smaller than the current trust region size, Δ_k then solve:

$$\begin{aligned} \min_{\mathbf{s}_k \in \mathbb{R}^n} \quad & m_k(\mathbf{x}_k + \mathbf{s}_k) \\ \text{s.t.} \quad & \mathbf{h}(\mathbf{x}_k + \mathbf{s}_k) = 0 \\ & \mathbf{g}(\mathbf{x}_k + \mathbf{s}_k) \leq 0 \\ & \|\mathbf{s}_k\| \leq \Delta_k, \end{aligned}$$

to convergence tolerance τ_k.

2b: If 2a is not used or fails to converge to the required tolerance, solve the trust-region subproblem:

$$\begin{aligned} \min_{\mathbf{s}_k \in \mathbb{R}^n} \quad & \hat{\Phi}_k(\mathbf{x}_k + \mathbf{s}_k, \sigma_k) \\ \text{s.t.} \quad & \|\mathbf{s}_k\| \leq \Delta_k. \end{aligned}$$

to convergence tolerance τ_k.

3: If $f_{\text{high}}(\mathbf{x}_k + \mathbf{s}_k)$ has not been evaluated previously, evaluate the high-fidelity function at that point.

3a: Store $f_{\text{high}}(\mathbf{x}_k + \mathbf{s}_k)$ in database.

4: Compute the merit function $\Phi(\mathbf{x}_k, \sigma_k)$, $\Phi(\mathbf{x}_k + \mathbf{s}_k, \sigma_k)$ and the surrogate merit function, $\hat{\Phi}(\mathbf{x}_k + \mathbf{s}_k, \sigma_k)$.

5: Compute the ratio of actual improvement to predicted improvement,

$$\rho_k = \begin{cases} 0 & \hat{\Phi}(\mathbf{x}_k, \sigma_k) - \hat{\Phi}(\mathbf{x}_k + \mathbf{s}_k, \sigma_k) < a\Delta_k \\ \frac{\Phi(\mathbf{x}_k, \sigma_k) - \Phi(\mathbf{x}_k + \mathbf{s}_k, \sigma_k)}{\hat{\Phi}(\mathbf{x}_k, \sigma_k) - \hat{\Phi}(\mathbf{x}_k + \mathbf{s}_k, \sigma_k)} & \text{otherwise.} \end{cases}$$

6: Update the trust region size according to ρ_k,

$$\Delta_{k+1} \quad = \quad \begin{cases} \min\{\gamma_1 \Delta_k, \Delta_{\max}\} & \text{if } \eta_1 \leq \rho_k \leq \eta_2 \\ \gamma_0 \Delta_k & \text{if } \rho_k \leq \eta_0, \\ \Delta_k & \text{otherwise,} \end{cases}$$

7: Accept or reject the trial point according to improvement in the merit function,

$$\mathbf{x}_{k+1} = \begin{cases} \mathbf{x}_k + \mathbf{s}_k & \Phi(\mathbf{x}_k, \sigma_k) - \Phi(\mathbf{x}_k + \mathbf{s}_k, \sigma_k) > 0 \\ \mathbf{x}_k & \text{otherwise.} \end{cases}$$

8: Create new model $m_{k+1}(\mathbf{x})$ fully linear on $\{\mathbf{x} : \|\mathbf{x} - \mathbf{x}_{k+1}\| \leq \Delta_{k+1}\}$.

9: Increment the penalty, $\sigma_{k+1} = \max\left[e^{k+1/10}, 1/\Delta_{k+1}^{1.1}\right]$.

10: Check for convergence, if the trust region constraint is inactive, $\|\nabla m_{k+1}(\mathbf{x}_{k+1}) + A(\mathbf{x}_{k+1})^T \hat{\lambda}\| \leq \beta\epsilon$, $\|[\mathbf{h}(\mathbf{x}_k), \mathbf{g}^+(\mathbf{x}_k)]\| \leq \epsilon$, and $\Delta_k \leq \epsilon_2$ the algorithm is converged, otherwise go to step 1.

fidelity objective function and the high-fidelity constraint. The surrogate model for the objective function is $m_k(\mathbf{x})$ as defined in Eq. 2. For the constraint, the surrogate model, $\bar{m}_k(\mathbf{x})$, is defined as

$$\bar{m}_k(\mathbf{x}) = c_{\text{low}}(\mathbf{x}) + \bar{e}_k(\mathbf{x}). \tag{37}$$

From the fact that $\bar{m}(\mathbf{x})$ is a fully linear model for $c_{\text{high}}(\mathbf{x})$, we know

$$\|\nabla c_{\text{high}}(\mathbf{x}) - \nabla \bar{m}_k(\mathbf{x})\| \leq \kappa_{cg}\Delta_k \quad \forall \mathbf{x} \in \mathcal{B}_k \tag{38}$$

$$|c_{\text{high}}(\mathbf{x}) - \bar{m}_k(\mathbf{x})| \leq \kappa_{cf}\Delta_k^2 \quad \forall \mathbf{x} \in \mathcal{B}_k. \tag{39}$$

In addition, we require that our procedure to construct fully linear models ensures that at the current design iterate, the fully linear models have the exact value of the function they are modeling. Accordingly, we have that

$$f_{\text{high}}(\mathbf{x}_k) = m_k(\mathbf{x}_k) \tag{40}$$

$$c_{\text{high}}(\mathbf{x}_k) = \bar{m}_k(\mathbf{x}_k). \tag{41}$$

This is required so that every trust-region subproblem is feasible at the initial point $\mathbf{x}_k$.

The trust-region subproblem used is to minimize the surrogate high-fidelity objective function subject to the easy constraints, the surrogate high-fidelity constraint, and the trust-region constraint,

$$\begin{aligned} \min_{\mathbf{s}_k \in \mathbb{R}^n} \quad & m_k(\mathbf{x}_k + \mathbf{s}_k) \\ \text{s.t.} \quad & \mathbf{h}(\mathbf{x}_k + \mathbf{s}_k) = 0 \\ & \mathbf{g}(\mathbf{x}_k + \mathbf{s}_k) \leq 0 \\ & \bar{m}_k(\mathbf{x}_k + \mathbf{s}_k) \leq \max\{c_{\text{high}}(\mathbf{x}_k), -e\Delta_k\} \\ & \|\mathbf{s}_k\| \leq \Delta_k. \end{aligned} \tag{42}$$

The trust-region subproblem is solved to the same termination tolerance as the multifidelity objective function formulation, $\|\nabla_x \mathfrak{L}_k\| \leq \tau_k$, where $\mathfrak{L}_k$ is the Lagrangian. The surrogate model constraint does not have zero as a right hand side to account for the fact the algorithm is looking for interior points. The right hand side, $\max\{c_{\text{high}}(\mathbf{x}_k), -e\Delta_k\}$, where e is a positive constant greater than c used in the definition of τ_k, assures that the constraint is initially feasible and that protection of constraint violation decreases to zero as the number of iterations increase to infinity.

The center of the trust region is updated if a decrease in the objective function is found at a feasible point,

$$\mathbf{x}_{k+1} = \begin{cases} \mathbf{x}_k + \mathbf{s}_k & \text{if } f_{\text{high}}(\mathbf{x}_k) > f_{\text{high}}(\mathbf{x}_k + \mathbf{s}_k) \text{ and } c_{\text{high}}(\mathbf{x}_k + \mathbf{s}_k) \leq 0 \\ \mathbf{x}_k & \text{otherwise.} \end{cases} \tag{43}$$

The trust-region size update must ensure that the predictions of the surrogate models are accurate and that the size of the trust region goes to zero in the limit as the number of iterations goes to infinity. Therefore, we again impose a sufficient decrease condition, that the change in the objective function is at least a constant, a, multiplying Δ_k,

$$\Delta_{k+1} = \begin{cases} \min\{\gamma_1\Delta_k, \Delta_{\max}\} & \text{if } f_{\text{high}}(\mathbf{x}_k + \mathbf{s}_k) - f_{\text{high}}(\mathbf{x}_k) \geq a\Delta_k \text{ and } c_{\text{high}}(\mathbf{x}_k + \mathbf{s}_k) \leq 0 \\ \gamma_0\Delta_k & \text{otherwise.} \end{cases} \tag{44}$$

New surrogate models, $m_{k+1}(\mathbf{x})$ and $\bar{m}_{k+1}(\mathbf{x})$, are then built such that they are fully linear on a region $\mathcal{B}_{k+1}$ having center $\mathbf{x}_{k+1}$ and size Δ_{k+1}. The new fully linear models are constructed using the procedure of Wild *et al.*[20] with the calibration technique of March and Willcox.[23]

Convergence of the algorithm is verified by assuring the KKT conditions are satisfied on the surrogate models and that the trust-region radius is sufficiently small to establish that the error between the surrogate model gradients and the high-fidelity function gradients is sufficiently small.

III.C. Multifidelity Objective and Constraints Convergence Discussion

To demonstrate convergence of this algorithm we must first show that the trust region size goes to zero and second that the algorithm cannot stop before the KKT conditions are satisfied for the high-fidelity problem. To establish that the trust region size goes to zero, we use the assumption that a quadratic penalty function including all of constraints is bounded from below. This means that the high fidelity objective function must be bounded from below within the feasible region of Eq. 36. We therefore know that from the first feasible iterate found, the difference between the high-fidelity function at that point and the value of the high-fidelity function at an optimum of this problem is finite. Since we have required that the decrease in the objective function is at least a constant a multiplying Δ_k this means the sum of the series $\{\Delta_k\}$ is bounded, and accordingly

$$\lim_{k \to +\infty} \Delta_k = 0. \tag{45}$$

We now establish that this algorithm must converge to a limit point, $\mathbf{x}^*$, where the KKT conditions,

$$\|\nabla f_{\text{high}}(\mathbf{x}_k) + A(\mathbf{x}_k)^T \lambda(\mathbf{x}_k) + \nabla c_{\text{high}}(\mathbf{x}_k)\lambda_c\| \leq \epsilon, \tag{46}$$

$$\| \left[\mathbf{h}(\mathbf{x}_k), \mathbf{g}^+(\mathbf{x}_k)\right] \| \leq \epsilon, \tag{47}$$

$$c_{\text{high}}(\mathbf{x}_k) \leq 0, \tag{48}$$

with λ_c the Lagrange multiplier for the high-fidelity constraint, are satisfied. The constraint violation criteria, Eq. 47 and Eq. 48, are established by construction and the use of Algorithm 1. To establish the first-order condition, Eq. 46, we must use the property of a fully linear model given in Eqs. 10 and 38, which shows that as $\Delta_k \to 0$, then $\|\nabla f_{\text{high}}(\mathbf{x}) - \nabla m_k(\mathbf{x})\| \to 0$ and a similar conclusion for the high-fidelity constraint. When the trust-region subproblem given in Eq. 42 has been solved and the trust-region constraint is inactive, we know that

$$\|\nabla m(\mathbf{x}_k) + A(\mathbf{x}_k)^T \hat{\lambda}(\mathbf{x}_k) + \nabla \bar{m}(\mathbf{x}_k)\lambda_{\bar{m}}\| \leq \min\left[\beta\epsilon, c\Delta_k\right], \tag{49}$$

where $\hat{\lambda}$ are the appropriate Lagrange multipliers computed using the surrogate model as opposed to the high-fidelity function (see Eq. 24) and $\lambda_{\bar{m}}$ is a similar Lagrange multiplier for the surrogate model of the high-fidelity constraint. Thus we have established that as $\Delta_k \to 0$, then $\|\nabla f_{\text{high}}(\mathbf{x}) - \nabla m_k(\mathbf{x})\| \to 0$, $\|\nabla c_{\text{high}}(\mathbf{x}) - \nabla \bar{m}_k(\mathbf{x})\| \to 0$, $\|\hat{\lambda} - \lambda\| \to 0$, $\|\lambda_c - \lambda_{\bar{m}}\| \to 0$, and $\|\nabla f_{\text{high}}(\mathbf{x}_k) + A(\mathbf{x}_k)^T \hat{\lambda}(\mathbf{x}_k) + \nabla c_{\text{high}}(\mathbf{x}_k)\lambda_{\bar{m}}\| \leq \epsilon$. In addition, if no constraints are active we have only established convergence to a point such that, $\|\nabla f_{\text{high}}(\mathbf{x})\| \leq a$, but this also satisfies the first-order KKT conditions. The other way in which this algorithm might fail, is that the sequence of iterates could always have $c_{\text{high}}(\mathbf{x}_k + \mathbf{s}_k) > 0$, in which case the trust region will shrink to zero without making any progress. However, a necessary contradiction is that the error in the value of a fully linear model decreases with the square of the trust region size. The fact that a margin is kept with respect to the high-fidelity constraint violation and that the error in the value of the constraint value prediction decays at a faster rate than the error in the first-order KKT condition shows the prediction of the constraint surrogate model value will not force the size of the trust-region to zero before the KKT conditions are satisfied.

III.D. Multifidelity Objective and Constraint Implementation

The numerical implementation of this multifidelity optimization algorithm is presented as Algorithm 2. A set of possible parameters that may be used in this algorithm are listed in Table 2 in Section IV. An important implementation issue with this algorithm is finding the initial feasible point. Algorithm 1 is used to minimize the high-fidelity constraint value subject to the constraints with available derivatives in order to find a point that is feasible. However, Algorithm 1 uses a quadratic penalty function to handle the constraints with available derivatives if the constraints are violated. This requires that the objective function is bounded from below to assure the penalty method is effective. Accordingly, a more general problem than Eq. 36 to find an initial feasible point is to use,

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} \quad & \max\{c_{\text{high}}(\mathbf{x}) + d, 0\}^2 \\ \text{s.t.} \quad & \mathbf{h}(\mathbf{x}) = 0 \\ & \mathbf{g}(\mathbf{x}) \leq 0. \end{aligned} \tag{50}$$

The maximization in the objective prevents the need for the high-fidelity constraint to be bounded from below. The constant d is used to account for the fact that the surrogate model will have some error in its prediction of $c_{\text{high}}(\mathbf{x})$, so looking for a slightly negative value of the constraint may save iterations as compared to seeking a value that is exactly zero. For example, if d is larger than the κ_{cf} in Eq. 39 and the value of the constraint surrogate model is more negative than the size of the trust region squared, then we are assured the actual value of the high-fidelity constraint is negative.

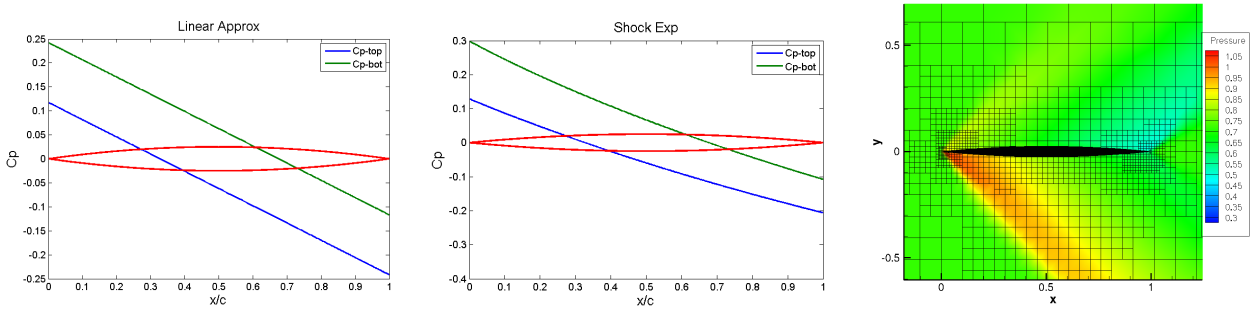
A similar issue is in the solution of Eq. 42, where a slightly negative value of the surrogate constraint is desired. If this subproblem is solved with an interior point algorithm this should be satisfied automatically; however, if a sequential quadratic programming method is used the constraint violation will have a numerical tolerance that is either slightly negative or slightly positive. It may be necessary to bias the subproblem to look for a value of the constraint that is more negative than the optimizer constraint violation tolerance to assure the solution is an interior point.

A final implementation note is that if a high-fidelity constraint has numerical noise or steep gradients it may be wise to shrink the trust region at a slower rate, increasing γ_0. This will help to ensure that the trust-region does not decrease to zero at a suboptimal point.

IV. Supersonic Airfoil Design Test Problem

This section presents results of the two multifidelity optimization algorithms on a supersonic airfoil design problem. The airfoil design problem has 11 parameters, the angle of attack, 5 spline points on the upper surface and 5 spline points on the lower surface. However, there is an actual total of 7 spline points on both the upper and lower surfaces because the leading and trailing edges must be sharp for the low-fidelity methods used. The airfoils are constrained such that the minimum thickness-to-chord ratio is 0.05 and that the thickness everywhere on the airfoil must be positive. In addition, there are lower and upper bounds for all spline points.

Three supersonic airfoil analysis models are available: a linearized panel method, a nonlinear shock-expansion theory method, and Cart3D, an Euler CFD solver.[29] Note, that Cart3D has a finite convergence tolerance so there is some numerical noise in the lift and drag predictions. In addition, because random airfoils are used as initial conditions, Cart3D may fail to converge, in which case the results of the panel method are used. Figure 1 shows computed pressure distributions for each of the models for a 5% thick biconvex airfoil at Mach 1.5 and 2° angle of attack. Table 1 provides the estimated lift and drag coefficients for the same airfoil and indicates the approximate level of accuracy of the codes with respect to each other.



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

Figure 1. Supersonic airfoil model estimated pressure distribution comparisons for a 5% thick biconvex airfoil at Mach 1.5 and 2° angle of attack.

The following sections present results for three optimization examples each using this airfoil problem to demonstrate the capabilities of the optimization algorithms presented. In the first example, Section IV.A, the airfoil drag will be minimized using the constrained multifidelity objective function formulation presented in Section II with only the simple geometric constraints. In the second example, Section IV.B, the airfoil lift to drag ratio will be maximized subject to a constraint that the drag coefficient is less than 0.01, where the constraint will be handled with the multifidelity framework presented in Section III. In the final example, Section IV.C, the airfoil lift to drag ratio will be maximized subject to the constrained drag coefficient and both the objective function and the constraints are handled with the multifidelity framework presented in

Algorithm 2: Multifidelity Objective and Constraint Trust-Region Algorithm

0: Find a feasible design vector using Algorithm 1 to iterate on:

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & \max\{c_{\text{high}}(\mathbf{x}) + d, 0\}^2 \\ \text{s.t.} \quad & \mathbf{h}(\mathbf{x}) = 0 \\ & \mathbf{g}(\mathbf{x}) \leq 0. \end{aligned}$$

1: Update tolerance, $\tau_k = \min\left[\beta\epsilon, c\Delta_k\right]$.

2: Solve the trust-region subproblem:

$$\begin{aligned} \min_{\mathbf{s}_k\in\mathbb{R}^n} \quad & m_k(\mathbf{x}_k + \mathbf{s}_k) \\ \text{s.t.} \quad & \mathbf{h}(\mathbf{x}_k + \mathbf{s}_k) = 0 \\ & \mathbf{g}(\mathbf{x}_k + \mathbf{s}_k) \leq 0 \\ & \bar{m}_k(\mathbf{x}_k + \mathbf{s}_k) \leq \min\{c_{\text{high}}(\mathbf{x}_k), -e\Delta_k\} \\ & \|\mathbf{s}_k\| \leq \Delta_k. \end{aligned}$$

3: If $f_{\text{high}}(\mathbf{x}_k + \mathbf{s}_k)$ or $c_{\text{high}}(\mathbf{x}_k + \mathbf{s}_k)$ have not been evaluated previously, evaluate the high-fidelity functions at that point.

3a: Store $f_{\text{high}}(\mathbf{x}_k + \mathbf{s}_k)$ and $c_{\text{high}}(\mathbf{x}_k + \mathbf{s}_k)$ in a database.

4: Accept or reject the trial point according to:

$$\mathbf{x}_{k+1} = \begin{cases} \mathbf{x}_k + \mathbf{s}_k & \text{if } f_{\text{high}}(\mathbf{x}_k) > f_{\text{high}}(\mathbf{x}_k + \mathbf{s}_k) \text{ and } c_{\text{high}}(\mathbf{x}_k + \mathbf{s}_k) \leq 0 \\ \mathbf{x}_k & \text{otherwise.} \end{cases}$$

5: Update the trust region size according to:

$$\Delta_{k+1} = \begin{cases} \min\{\gamma_1\Delta_k, \Delta_{\max}\} & \text{if } f_{\text{high}}(\mathbf{x}_k + \mathbf{s}_k) - f_{\text{high}}(\mathbf{x}_k) \geq a\Delta_k \text{ and } c_{\text{high}}(\mathbf{x}_k + \mathbf{s}_k) \leq 0 \\ \gamma_0\Delta_k & \text{otherwise.} \end{cases}$$

6: Create new models $m_{k+1}(\mathbf{x})$ and $\bar{m}_{k+1}(\mathbf{x})$ fully linear on $\{\mathbf{x} : \|\mathbf{x} - \mathbf{x}_{k+1}\| \leq \Delta_{k+1}\}$.

7: Check for convergence, if the trust region constraint is inactive, $\|\nabla m(\mathbf{x}_k) + A(\mathbf{x}_k)^T\hat{\lambda}(\mathbf{x}_k) + \nabla\bar{m}(\mathbf{x}_k)\lambda_{\bar{m}}\| \leq \beta\epsilon$, and $\Delta_k \leq \epsilon_2$, the algorithm is converged, otherwise go to step 1.

Section III. The initial airfoils for all problems will be randomly generated and likely will not satisfy the constraints.

The three airfoil problems will be solved with four optimization algorithms: Sequential Quadratic Programming[30] (SQP), a first-order consistent multifidelity trust-region algorithm that uses a sequential quadratic programming formulation and an additive correction,[2] the high-fidelity-gradient-free approach presented in this paper using a Gaussian radial basis function and a fixed spatial correlation parameter, $\exp -r^2/\xi^2$ with $\xi = 2$, and the approach presented in this paper using a maximum likelihood estimate to find an improved correlation length, $\xi = \xi^*$, for the Gaussian radial basis function.[23] The Gaussian correlation functions used in these results are all isotropic, meaning they are the same in all directions. An anisotropic correlation function will likely speed convergence of this algorithm and reduce sensitivity to the problem scaling. The parameters used for the optimization algorithm are presented in Table 2.

	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 1. 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.

Constant	Description	Value
a	Sufficient decrease constant	1×10^{-4}
β	Overall Convergence tolerance multiplier	1×10^{-2}
c	Convergence tolerance based on trust region size	1×10^{-2}
d	Artificial lower bound for constraint value	1
e	Additional conservatism for constraint violation	0.1
ϵ, ϵ_2	Termination Tolerance	5×10^{-4}
γ_0	Trust region contraction ratio	0.5
γ_1	Trust region expansion ratio	2
η_0	Trust region contraction criterion	0.25
η_1, η_2	Trust region expansion criterion	$0.75, 2.0$
Δ_0	Initial trust region radius	1
Δ_{max}	Maximum trust region size	20
σ_k	Penalty parameter	$\max\left[e^{k/10}, 1/\Delta_k^{1.1}\right]$
δ_x	Finite difference step	1×10^{-5}

Table 2. List of constants used in the algorithm. All parameters used in constructing the radial basis function error model are the same is in Ref. 23.

IV.A. Multifidelity Objective Function Results

This section presents optimization results in terms of the number of function evaluations required to find the minimum drag for a supersonic airfoil at Mach 1.5 with only geometric constraints on the design. Two cases are tested, the first uses the shock-expansion method as the high-fidelity function and the panel method as the low-fidelity function, the second uses Cart3D as the high-fidelity function and the panel method as the low-fidelity function. These problems are solved using the multifidelity optimization algorithm for a computationally expensive objective function and constraints with available derivatives presented in Section II. The average number of high-fidelity function evaluations required to find a locally optimal design are presented in Table 3, and show that this approach uses approximately 78% fewer high-fidelity function evaluations than SQP and approximately 30% fewer function evaluations than the first order consistent trust-region method using finite difference gradient estimates.

IV.B. Multifidelity Constraint Results

This section presents optimization results in terms of the number of function evaluations required to find the maximum lift to drag ratio for a supersonic airfoil at Mach 1.5 subject to both geometric constraints and

High-Fidelity	Low-Fidelity	SQP	First-Order TR	RBF, $\xi = 2$	RBF, $\xi = \xi^*$
Shock-Expansion	Panel Method	314 (-)	110 (-65%)	73 (-77%)	68 (-78%)
Cart3D	Panel Method	359*(-)	109 (-70%)	80 (-78%)	79 (-78%)

Table 3. The average number of high-fidelity function evaluations to minimize the drag of a supersonic airfoil with only geometric constraints. The asterisk for the Cart3D results means a significant fraction of the optimizations failed and the average is taken over fewer samples. The numbers in parentheses indicate the percentage reduction in high-fidelity function evaluations relative to SQP.

the requirement that the drag coefficient is less than 0.01. The lift to drag ratio is computed with the panel method; however, the drag coefficient constraint is handled using the multifidelity technique presented in Section III. Two cases are examined, in the first the shock-expansion method is the high-fidelity constraint and the panel method is the low-fidelity constraint, in the second case Cart3D is the high-fidelity constraint and the panel method is the low-fidelity constraint. Table 4 presents the average number of high-fidelity constraint evaluations required to find the optimal design using SQP, a first-order consistent trust-region algorithm and the techniques developed in this paper. A significant decrease in the number of high-fidelity function evaluations is observed when compared with SQP; however, performance is almost the same as the first-order consistent trust-region algorithm.

High-Fidelity	Low-Fidelity	SQP	First-Order TR	RBF, $\xi = 2$	RBF, $\xi = \xi^*$
Shock-Expansion	Panel Method	827 (-)	104 (-87%)	104 (-87%)	115 (-86%)
Cart3D	Panel Method	909*(-)	100 (-89%)	103 (-89%)	105 (-88%)

Table 4. The average number of high-fidelity constraint evaluations required to maximize the lift to drag ratio of a supersonic airfoil estimated with a panel method subject to a multifidelity constraint. The asterisk for the Cart3D results means a significant fraction of the optimizations failed and the average is taken over fewer samples. The numbers in parentheses indicate the percentage reduction in high-fidelity function evaluations relative to SQP.

IV.C. Multifidelity Objective Function and Constraint Results

This section presents optimization results in terms of the number of function evaluations required to find the maximum lift to drag ratio for a supersonic airfoil at Mach 1.5 subject to geometric constraints and the requirement that the drag coefficient is less than 0.01. In this case, both the lift to drag ratio and the drag coefficient constraint are handled using the multifidelity technique presented in Section III. In the first case, the shock-expansion method is the high-fidelity analysis used to estimate both metrics of interest and the panel method is the low-fidelity analysis, in the second case Cart3D is the high-fidelity analysis and the panel method is the low-fidelity analysis. Table 5 presents the number of function evaluations required to find the optimal design using SQP, a first-order consistent trust-region algorithm and the techniques developed in this paper. Again a significant reduction in the number of high-fidelity function evaluations, both in terms of the constraint and the objective, are observed compared with SQP, and a similar number of high-fidelity function evaluations are observed when compared with the first-order consistent trust region approach using finite differences.

	High-Fidelity	Low-Fidelity	SQP	First-Order TR	RBF, $\xi = 2$	RBF, $\xi = \xi^*$
Objective:	Shock-Expansion	Panel Method	773 (-)	132 (-83%)	93 (-88%)	90 (-88%)
Constraint:	Shock-Expansion	Panel Method	773 (-)	132 (-83%)	97 (-87%)	96 (-88%)
	High-Fidelity	Low-Fidelity	SQP	First-Order TR	RBF, $\xi = 2$	RBF, $\xi = \xi^*$
Objective:	Cart3D	Panel Method	1168*(-)	97 (-92%)	104 (-91%)	112 (-90%)
Constraint:	Cart3D	Panel Method	2335*(-)	97 (-96%)	115 (-95%)	128 (-94%)

Table 5. The average number of high-fidelity objective function and high-fidelity constraint evaluations to optimize a supersonic airfoil for a maximum lift to drag ratio subject to a maximum drag constraint. The asterisk for the Cart3D results means a significant fraction of the optimizations failed and the average is taken over fewer samples. The numbers in parentheses indicate the percentage reduction in high-fidelity function evaluations relative to SQP.

V. Conclusion

This paper has presented two algorithms for multifidelity constrained optimization of computationally expensive functions when their derivatives are not available. The first method minimizes a high-fidelity objective function without using its derivative while satisfying constraints with available derivatives. The second method minimizes a high-fidelity objective without using its derivative while satisfying both constraints with available derivatives and an additional high-fidelity constraint without an available derivative. Both of these methods support multiple lower-fidelity models through the use of a multifidelity filtering technique without any modifications to the methods. The performance of the methods show a significant reduction in the number of high-fidelity function evaluations required to optimize a computationally expensive function when an appropriate low-fidelity function is used compared to a single-fidelity sequential quadratic programming formulation. In addition, this paper demonstrated that these methods performed similarly to a first-order consistent trust-region algorithm with gradients estimated using finite-differences. This shows that these methods provide significant opportunity for optimization of computationally expensive functions without available gradients.

The behavior of the algorithms presented are slightly atypical of nonlinear programming methods. Although the algorithms guarantee convergence to a local minimum of the high-fidelity problem, the local minimum they converge to may not be the one in the immediate vicinity of the initial iterate. For example, the initial model may completely ignore the fact that the initial iterate is on a local minimum and move the iterate to a different point with a lower function value in another region of the design space. In addition, provided the trust-region subproblems can be solved, this algorithm should still be able to converge to high-fidelity optima at which the KKT conditions are not satisfied. This situation occurs when constraint qualification conditions are not satisfied and can cause robustness issues for nonlinear programming methods. In the algorithms presented, the design vector will approach a local minimum and the sufficient decrease test for the change in the objective function value will fail. This causes the size of the trust region to decay to zero around the local minimum even though the KKT conditions are not satisfied.

In the case of hard constraints, or when the objective function fails to exist if the constraints are violated, it is still be possible to use the algorithm that approximates both a high-fidelity objective and constraint. After the initial feasible point is found, no design iterate will be accepted if the high-fidelity constraint is violated. Therefore the overall flow of the algorithm is unchanged. What must be changed is the technique to build fully linear models. In order to build a fully linear model, the objective function must be evaluated at a set of $n+1$ points that span $\mathbb{R}^n$. This requirement prohibits equality constraints and means that strict linear independent constraint qualification must be satisfied everywhere in the design space (preventing two inequality constraints from mimicking an equality constraint). If these two additional conditions hold then it will be possible to construct fully linear models everywhere in the feasible design space and use this algorithm to optimize computationally expensive functions with hard constraints. Therefore this paper has presented a provably convergent multifidelity optimization algorithm that does not require estimating high-fidelity gradients, enables the use of multiple low-fidelity models, enables optimization of functions with hard constraints, exhibits robustness to optima at which the KKT conditions are not satisfied, and performs similarly in terms of the number of function evaluations to finite difference based multifidelity optimization methods.

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

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

[3]Jones, D., Schonlau, M., and Welch, W., "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of*

Global Optimization, Vol. 13, 1998, pp. 455–492.

[4]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.

[5]Jones, D., "A taxonomy of global optimization methods based on response surfaces," *Journal of Global Optimization*, Vol. 21, 2001, pp. 345–383.

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

[7]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.

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

[9]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.

[10]Rajnarayan, D., Haas, A., and Kroo, I., "A Multifidelity Gradient-Free Optimization Method and Application to Aerodynamic Design," *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Vol. 2008-6020, Victoria, British Columbia, September 10-12 2008.

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

[12]Kolda, T. G., Lewis, R. M., and Torczon, V., "Optimization by Direct Search: New Perspectives on Classical and Modern Methods," *SIAM Review*, Vol. 45, No. 3, 2003, pp. 385–482.

[13]Kolda, T. G., Lewis, R. M., and Torczon, V., "A Generating Set Direct Search Augmented Lagrangian Algorithm for Optimization with a Combination of General and Linear Constraints," Tech. Rep. SAND2006-5315, Sandia, August 2006.

[14]Lewis, R. M. and Torczon, V., "A Direct Search Approach to Nonlinear Programming Problems Using an Augmented Lagrangian Method with Explicit Treatment of Linear Constraints," Tech. Rep. WM-CS-2010-01, College of William and Mary Department of Computer Science, January 2010.

[15]Liuzzi, G. and Lucidi, S., "A Derivative-Free Algorithm for Inequality Constrained Nonlinear Programming via Smoothing of an l_∞ Penalty Function," *SIAM Journal of Optimization*, Vol. 20, No. 1, 2009, pp. 1–29.

[16]Audet, C. and Dennis, J. E., "A Pattern Search Filter Method for Nonlinear Programming without Derivatives," *SIAM Journal of Optimization*, Vol. 14, No. 4, 2004, pp. 980–1010.

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

[18]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, PA, 2009.

[19]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 of Optimization*, Vol. 20, No. 1, 2009, pp. 387–415.

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

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

[22]Wild, S. 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.

[23]March, A. and Willcox, K., "A Provably Convergent Multifidelity Optimization Algorithm not Requiring High-Fidelity Gradients," 6th AIAA Multidisciplinary Design Optimization Specialist Conference, Orlando, FL, 2010.

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

[25]Bertsekas, D. P., *Nonlinear Programming, 2nd ed.*, Athena Scientific, 1999.

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

[27]Rvachev, V., "On the analytical description of some geometric objects," Tech. Rep. 4, Reports of Ukrainian Academy of Sciences, 1963, (in Russian).

[28]Papalambros, P. Y. and Wilde, D. J., *Principles of Optimal Design 2nd ed.*, Cambridge University Press, 2000.

[29]Aftosmis, M. J., "Solution Adaptive Cartesian Grid Methods for Aerodynamic Flows with Complex Geometries," *28th Computational Fluid Dynamics Lecture Series, von Karman Institute for Fluid Dynamics*, Rhode-Saint-Genése, Belgium, March 3-7 1997, Lecture Series 1997-02.

[30]MathWorks, Inc., "Constrained Nonlinear Optimization," Optimization Toolbox User's Guide, V. 5, 2010.