Struct Multidisc Optim (2012) 46:93–109
DOI 10.1007/s00158-011-0749-1

RESEARCH PAPER

# Constrained multifidelity optimization using model calibration

**Andrew March · Karen Willcox**

Received: 4 April 2011 / Revised: 22 November 2011 / Accepted: 28 November 2011 / Published online: 8 January 2012

**Abstract** 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 bridges these two ideas; our method iteratively calibrates lower-fidelity information to the high-fidelity function in order to find 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 achieves first-order optimality 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 at least an 80% reduction in the number of high-fidelity analyses compared other single-fidelity derivative-free and sequential quadratic programming methods. The method uses approximately the same number of high-fidelity analyses as a multifidelity trust-region algorithm that estimates the high-fidelity gradient using finite differences.

**Keywords** Multifidelity · Derivative-free · Optimization · Multidisciplinary · Aerodynamic design

## Nomenclature

| | |
|---|---|
| $A$ | Active and violated constraint Jacobian |
| $a$ | Sufficient decrease parameter |
| $B$ | A closed and bounded set in $\mathbb{R}^n$ |
| $\mathscr{B}$ | Trust region |
| $C$ | Differentiability Class |
| $c(\mathbf{x})$ | Inequality constraint |
| $d$ | Artificial lower bound for constraint value |
| $e(\mathbf{x})$ | Error model for objective |
| $\bar{e}(\mathbf{x})$ | Error model for constraint |
| $f(\mathbf{x})$ | Objective function |
| $g(\mathbf{x})$ | Inequality constraint |
| $h(\mathbf{x})$ | Equality constraint |
| $\mathscr{L}$ | Expanded level-set in $\mathbb{R}^n$ |
| $L$ | Level-set in $\mathbb{R}^n$ |
| $\mathfrak{L}$ | Lagrangian of trust-region subproblem |
| $\mathscr{M}$ | Space of all fully linear models |
| $m(\mathbf{x})$ | Surrogate model of the high-fidelity objective function |
| $\bar{m}(\mathbf{x})$ | Surrogate model of the high-fidelity constraint |
| $n$ | Number of design variables |
| $\mathbf{p}$ | Arbitrary step vector |
| $RBF$ | Radial Basis Function |

A version of this paper was presented as paper AIAA-2010-9198 at the 13th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Fort Worth, TX, 13–15 September 2010.

A. March (✉) · K. Willcox
Department of Aeronautics & Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
e-mail: amarch@mit.edu

K. Willcox
e-mail: kwillcox@mit.edu

| | |
|---|---|
| $r$ | Radial distance between two points |
| $\mathbf{s}$ | Trust-region step |
| $t$ | Constant between 0 and 1 |
| $w$ | Constraint violation conservatism factor |
| $\mathbf{x}$ | Design vector |
| $\alpha$ | Convergence tolerance multiplier |
| $\beta$ | Convergence tolerance multiplier |
| $\gamma_0$ | Trust-region contraction ratio |
| $\gamma_1$ | Trust-region expansion ratio |
| $\varepsilon$ | Termination tolerance |
| $\varepsilon_2$ | Termination tolerance for trust region size |
| $\Delta$ | Trust region size |
| $\delta_h$ | Linearized step size for feasibility |
| $\delta_x$ | Finite difference step size |
| $\eta_0$ | Trust-region contraction criterion |
| $\eta_1, \eta_2$ | Trust-region expansion criterion |
| $\kappa$ | Bound related to function smoothness |
| $\lambda$ | Lagrange multiplier |
| $\hat{\lambda}$ | Lagrange multiplier for surrogate model |
| $\xi$ | Radial basis function correlation length |
| $\rho$ | Ratio of actual to predicted improvement |
| $\sigma$ | Penalty parameter |
| $\tau$ | Trust-region solution tolerance |
| $\Phi$ | Penalty function |
| $\hat{\Phi}$ | Surrogate penalty function |
| $\phi$ | Radial basis function |

Superscript

| | |
|---|---|
| $*$ | Optimal |
| $+$ | Active or violated inequality constraint |

Subscript

| | |
|---|---|
| 0 | Initial iterate |
| $bhm$ | Bound for Hessian of the surrogate model |
| $blg$ | Upper bound on the Lipschitz constant |
| $c$ | Relating to constraint |
| $cg$ | Relating to constraint gradient |
| $cgm$ | Maximum Lipschitz constant for a constraint gradient |
| $cm$ | Maximum Lipschitz constant for a constraint |
| $f$ | Relating to objective function |
| $fg$ | Lipschitz constant for objective function gradient |
| $FCD$ | Fraction of Cauchy Decrease |
| $g$ | Relating to objective function gradient |
| high | Relating to the high-fidelity function |
| $k$ | Index of trust-region iteration |
| low | Relating to a lower-fidelity function |
| $\bar{m}$ | Related to constraint surrogate model |
| max | User-set maximum value of that parameter |

## 1 Introduction

High-fidelity numerical simulation tools can provide valuable insight to designers of complex systems and support better decisions through optimal design. However, the design budget frequently prevents the use of formal optimization methods in conjunction with high-fidelity simulation tools, due to the high cost of simulations. An additional challenge for optimization is obtaining accurate gradient information for high-fidelity models, especially when the design process employs black-box and/or legacy tools. Instead, designers often set design aspects using crude lower-fidelity performance estimates and later attempt to optimize system subcomponents using higher-fidelity methods. This approach in many cases leads to suboptimal design performance and unnecessary expenses for system operators. An alternative is to use the approximate performance estimates of lower-fidelity simulations in a formal multifidelity optimization framework, which aims to speed the design process towards a high-fidelity optimal design using significantly fewer costly simulations. This paper presents a multifidelity optimization method for the design of complex systems, and targets in particular, systems for which accurate design sensitivity information is challenging to obtain.

Multifidelity optimization of systems with design constraints can be achieved with the approximation model management framework of Alexandrov et al. (1999, 2001). That technique optimizes a sequence of surrogate models of the objective function and constraints that are first-order consistent with their high-fidelity counterparts. The first-order consistent surrogates are constructed by correcting lower-fidelity models with either additive or multiplicative error models based on function value and gradient information of the high-fidelity models. This technique is provably convergent to a locally optimal high-fidelity design and in practice can provide greater than 50% reductions in the number of high-fidelity function calls, translating into important reductions in design turnaround times (Alexandrov et al. 1999). Estimating the needed gradient information is often a significant challenge, especially in the case of black-box simulation tools. For example, the high-fidelity numerical simulation tool may occasionally fail to provide a solution or it may contain noise in the output (e.g., due to numerical tolerances). In these cases, which arise frequently as both objectives and constraints in complex system design, if design sensitivities are unavailable then finite-difference gradient estimates will likely be inaccurate and non-robust. Therefore, there is a need for constrained multifidelity optimization methods that can guarantee convergence to a high-fidelity optimal design while reducing the number of expensive simulations, but that do not require gradient estimates of the high-fidelity models.

Constraint handling in derivative-free optimization is challenging. Common single-fidelity approaches use linear approximations (Powell 1994, 1998), an augmented Lagrangian (Kolda et al. 2003, 2006; Lewis and Torczon 2010), exact penalty method (Liuzzi and Lucidi 2009), or constraint filtering (Audet and Dennis 2004) in combination with either a pattern-search or a simplex method. In the multifidelity setting, if the generated surrogate models are smooth, then surrogate sensitivity information can be used to speed convergence.[1] This may provide an advantage over single-fidelity pattern-search and simplex methods, especially when the dimension of the design space is large. For example, constraint-handling approaches used in conjunction with Efficient Global Optimization (EGO) (Jones et al. 1998) either estimate the probability that a point is both a minimum and that it is feasible, or add a smooth penalty to the surrogate model prior to using optimization to select new high-fidelity sample locations (Sasena et al. 2002; Jones 2001; Rajnarayan et al. 2008). These heuristic approaches may work well in practice, but unfortunately have no guarantee of convergence to a minimum of the high-fidelity design problem. Another constraint-handling method used together with the Surrogate Management Framework (SMF) (Booker et al. 1999) augments a pattern-search method with predictions of locally optimal designs from a surrogate model. The underlying pattern-search method ensures convergence of the method, so a broad range of surrogate models are allowable. One technique for generating the surrogate model is conformal space mapping, where a low-fidelity function is calibrated to the high-fidelity function at locations where the value is known (Castro et al. 2005). This calibrated surrogate enables the inclusion of accurate local penalty methods as a sensitivity-based technique for quickly estimating the location of constrained high-fidelity optima.

These existing multifidelity methods highlight the benefits of exploiting sensitivity information generated from calibrated surrogate models in a derivative-free optimization setting. Here we propose a formal framework, based on a derivative-free trust region approach, to systematically generate calibration data and to control surrogate model quality. Thus, this paper develops a constrained multifidelity optimization method that in practice can quickly find high-fidelity optimal designs and be robust to unavailable or inaccurate gradient estimates, because it generates smooth surrogate models of high-fidelity objectives and constraints without requiring high-fidelity model gradient information. More specifically, this is accomplished by creating a fully linear model (formally defined in the next section), which establishes Lipschitz-type error bounds between the high-fidelity function and the surrogate model. This ensures that the error between the high-fidelity model gradient and surrogate model gradient is locally bounded without ever calculating the high-fidelity gradient. Conn et al. (2008) showed that polynomial interpolation models can be made fully linear, provided the interpolating set satisfies certain geometric requirements, and further developed an unconstrained gradient-free optimization technique using fully linear models (Conn et al. 2009a, b). Wild et al. (2008) 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 (Wild 2009; Wild and Shoemaker 2009). March and Willcox (2010) generalized this method to the cases with arbitrary low-fidelity functions or multiple low-fidelity functions using Bayesian model calibration methods.

Section 2 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 no further progress is possible or when convergence to a high-fidelity optimum is achieved. Section 3 presents a technique for minimizing a high-fidelity objective function subject to both constraints with available derivatives and computationally expensive constraints with unavailable 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. Section 4 presents an aerodynamic shape optimization problem to demonstrate the proposed multifidelity optimization techniques and compares the results with other single-fidelity methods and approximation model management using finite difference gradient estimates. Finally, Section 5 concludes the paper and discusses extensions of the method to the case when constraints are hard (when the objective function fails to exist if the constraints are violated).

## 2 Constrained optimization of a multifidelity objective function

This section considers the constrained optimization of a high-fidelity function, $f_{\text{high}}(\mathbf{x})$, that accurately estimates system metrics of interest but for which accurate gradient estimates are unavailable. We first present a formalized problem statement and some qualifying assumptions. We then present a trust-region framework, the surrogate-based optimization problems performed within the trust region, and the trust region updating scheme. We follow this with an algorithmic implementation of the method, and with a brief discussion of algorithmic limitations and theoretical considerations needed for robustness.

[1] Note that in the multifidelity setting, we use the term "derivative-free" to indicate an absence of derivatives of the high-fidelity model.

### 2.1 Problem setup and assumptions

We seek the vector $\mathbf{x} \in \mathbb{R}^n$ of $n$ design variables that minimizes the value of the high-fidelity objective function subject to equality constraints, $\mathbf{h}(\mathbf{x})$, and inequality constraints $\mathbf{g}(\mathbf{x})$,

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

where we assume gradients of $\mathbf{h}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ with respect to $\mathbf{x}$ are available or can be estimated accurately. To reduce the number of evaluations of $f_{\text{high}}(\mathbf{x})$ we use a low-fidelity function, $f_{\text{low}}(\mathbf{x})$, that estimates the same metric as $f_{\text{high}}(\mathbf{x})$ but with cheaper evaluation cost and lower accuracy. We seek to find the solution to (1) without estimating gradients of $f_{\text{high}}(\mathbf{x})$, by calibrating $f_{\text{low}}(\mathbf{x})$ to $f_{\text{high}}(\mathbf{x})$ and using sensitivity information from the calibrated surrogate model. The calibration strategy employed may break down should either $f_{\text{high}}(\mathbf{x})$ or $f_{\text{low}}(\mathbf{x})$ not be twice continuously differentiable or not have a Lipschitz continuous first derivative, although in many such cases the algorithm may still perform well.

### 2.2 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 $k$th trust-region iteration. Following the general Bayesian calibration approach in Kennedy and O'Hagan (2000), 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 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 $k$, $\mathscr{B}_k$, to be the region centered at $\mathbf{x}_k$ with size $\Delta_k$,

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

where any norm on $\mathbb{R}^n$ can be used.

To solve the constrained optimization problem presented in (1) we define a merit function, $\Phi(\mathbf{x}_k, \sigma_k)$, where $\sigma_k$ 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. To prevent divergence of this algorithm, we need the penalty function to satisfy some basic properties. First, the merit function with the initial penalty, $\sigma_0$, must be bounded from below within a relaxed level-set, $\mathscr{L}(\mathbf{x}_0, \sigma_0)$, defined as

$$L(\mathbf{x}_0, \sigma_0) = \left\{\mathbf{x} \in \mathbb{R}^n : \Phi(\mathbf{x}, \sigma_0) \le \Phi(\mathbf{x}_0, \sigma_0)\right\} \tag{4}$$

$$B(\mathbf{x}_k) = \left\{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x} - \mathbf{x}_k\| \le \Delta_{\max}\right\} \tag{5}$$

$$\mathscr{L}(\mathbf{x}_0, \sigma_0) = L(\mathbf{x}_0, \sigma_0) \bigcup_{\mathbf{x_k} \in L(\mathbf{x}_0, \sigma_0)} B(\mathbf{x}_k), \tag{6}$$

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 $L(\mathbf{x}_0, \sigma_0)$, and third, $L(\mathbf{x}_0, \sigma_0)$ must be a compact set. These properties ensure that all design iterates, $\mathbf{x}_k$, remain within $L(\mathbf{x}_0, \sigma_0)$.

Although other merit functions, such as augmented Lagrangians, 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 and there is no need to consider arbitrarily bad Lagrange multiplier estimates. The merit function used in this method is the objective function plus the scaled sum-squares of the constraint violation, where $\mathbf{g}^+(\mathbf{x})$ denotes the values of the nonnegative inequality constraints,

$$\Phi(\mathbf{x}, \sigma_k) = f_{\text{high}}(\mathbf{x}) + \frac{\sigma_k}{2}\mathbf{h}(\mathbf{x})^{\mathsf{T}}\mathbf{h}(\mathbf{x}) + \frac{\sigma_k}{2}\mathbf{g}^+(\mathbf{x})^{\mathsf{T}}\mathbf{g}^+(\mathbf{x}). \tag{7}$$

The parameter $\sigma_k$ 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, under suitable hypotheses on the optimization algorithm and penalty function, 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, $\mathbf{h}(\mathbf{x})^{\mathsf{T}}\mathbf{h}(\mathbf{x}) + \mathbf{g}^+(\mathbf{x})^{\mathsf{T}}\mathbf{g}^+(\mathbf{x})$ (Nocedal and Wright 2006; Bertsekas 1999).

We now define a surrogate merit function, $\hat{\Phi}(\mathbf{x}, \sigma_k)$, which replaces $f_{\text{high}}(\mathbf{x})$ with its surrogate model $m_k(\mathbf{x})$,

$$\hat{\Phi}(\mathbf{x}, \sigma_k) = m_k(\mathbf{x}) + \frac{\sigma_k}{2}\mathbf{h}(\mathbf{x})^{\mathsf{T}}\mathbf{h}(\mathbf{x}) + \frac{\sigma_k}{2}\mathbf{g}^+(\mathbf{x})^{\mathsf{T}}\mathbf{g}^+(\mathbf{x}). \tag{8}$$

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, $\Phi(\mathbf{x}, \sigma_k)$.

Our calibration strategy is to make the surrogate models $m_k(\mathbf{x})$ *fully linear*, where the following definition of a fully linear model is from Conn et al.:

**Definition 1** Let a function $f_{\text{high}}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ that is continuously differentiable and has a Lipschitz continuous derivative, be given. A set of model functions $\mathscr{M} = \{m : \mathbb{R}^n \to \mathbb{R}, m \in C^1\}$ 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 $\mathbf{x} \in L(\mathbf{x}_0, \sigma_0)$ and $\Delta_k \in (0, \Delta_{\max}]$ there exists a model function $m_k(\mathbf{x})$ in $\mathscr{M}$ with Lipschitz continuous gradient and corresponding Lipschitz constant bounded by $k_{blg}$, and such that the error between the gradient of the model and the gradient of the function satisfies

$$\|\nabla f_{\text{high}}(\mathbf{x}) - \nabla m_k(\mathbf{x})\| \leq \kappa_g \Delta_k \quad \forall \mathbf{x} \in \mathscr{B}_k \tag{9}$$

and the error between the model and the function satisfies

$$\left| f_{\text{high}}(\mathbf{x}) - m_k(\mathbf{x}) \right| \leq \kappa_f \Delta_k^2 \quad \forall \mathbf{x} \in \mathscr{B}_k. \tag{10}$$

Such a model $m_k(\mathbf{x})$ is called fully linear on $\mathscr{B}_k$ (Conn et al. 2009a).

## 2.3 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 $\Delta_k$:

$$\begin{aligned} \min_{\mathbf{s}_k \in \mathbb{R}^n} \quad & m_k(\mathbf{x}_k + \mathbf{s}_k) \\ \text{s.t.} \quad & \mathbf{h}(\mathbf{x}_k + \mathbf{s}_k) = 0 \\ & \mathbf{g}(\mathbf{x}_k + \mathbf{s}_k) \leq 0 \\ & \|\mathbf{s}_k\| \leq \Delta_k, \end{aligned} \tag{11}$$

or

$$\begin{aligned} \min_{\mathbf{s}_k \in \mathbb{R}^n} \quad & \hat{\Phi}_k(\mathbf{x}_k + \mathbf{s}_k, \sigma_k) \\ \text{s.t.} \quad & \|\mathbf{s}_k\| \leq \Delta_k. \end{aligned} \tag{12}$$

The subproblem in (12) is used initially to reduce constraint infeasibility. However, there is a limitation with this subproblem that the norm of the objective function Hessian grows without bound due to the penalty parameter increasing to infinity. Therefore to both speed convergence and prevent Hessian conditioning issues, the subproblem in (11) with explicit constraint handling is used as soon as a point that satisfies $\mathbf{h}(\mathbf{x}) = 0$ and $\mathbf{g}(\mathbf{x}) \leq 0$ exists within the current trust region. This is estimated by a linear approximation to the constraints, however, if the linearized estimate falsely suggests (11) has a feasible solution then we take recourse to (12).[2]

For both trust-region subproblems, (11) and (12), the subproblem must be solved such that the 2-norm of the first-order optimality conditions is less than a constant $\tau_k$. 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 $\tau_k$. First $\tau_k < \varepsilon$, where $\varepsilon$ is the desired termination tolerance for the optimization problem in (1). Second, $\tau_k$ must decrease to zero as the number of iterations goes to infinity. Accordingly, we define

$$\tau_k = \min\left[\beta\varepsilon, \alpha\Delta_k\right], \tag{13}$$

with a constant $\beta \in (0, 1)$ to satisfy the overall tolerance criteria, and a constant $\alpha \in (a, 1)$ multiplying $\Delta_k$ to ensure that $\tau_k$ 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 subsection.

## 2.4 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, a requirement similar to the fraction of Cauchy decrease requirement in an the unconstrained trust-region formulation is used (see, for example, Conn et al. 2009a). We require that the improvement in our merit function is at least a small constant $a \in (0, \varepsilon]$, multiplying $\Delta_k$,

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

The sufficient decrease condition is enforced through the trust region update parameter, $\rho_k$. 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, $\Delta_k$, 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, kept constant if the

[2] Note that an initial feasible point, $\mathbf{x}_0$ could be found directly using the gradients of $\mathbf{h}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$; however, since the penalty method includes the descent direction of the objective function it may better guide the optimization process in the case of multiple feasible regions. Should the initial iterate be feasible, the deficiencies of a quadratic penalty function are not an issue.

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 center 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 $\mathscr{B}_{k+1}$ having center $\mathbf{x}_{k+1}$ and size $\Delta_{k+1}$. The new fully linear model is constructed using the procedure of Wild et al. (2008) with the calibration technique of March and Willcox (2010).

### 2.5 Termination

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

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

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

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

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

and $\lambda(\mathbf{x}_k)$ are Lagrange multipliers. The additional complementarity conditions are assumed to be satisfied from the solution of (11). The constraint violation criteria, (19), can be evaluated directly. However, the first-order condition, (18), cannot be verified directly in the derivative-free case because the gradient, $\nabla f_{\text{high}}(\mathbf{x}_k)$, is unknown. Therefore, for first-order optimality we require two conditions: first-order optimality with the surrogate model, and a sufficiently small trust-region. The first-order optimality condition using the surrogate model is

$$\left\| \nabla m_k(\mathbf{x}_k) + A(\mathbf{x}_k)^{\mathsf{T}} \hat{\lambda}(\mathbf{x}_k) \right\| \leq \max\left[ \beta\varepsilon, \alpha\Delta_k, a \right] \leq \varepsilon, \tag{21}$$

where $\hat{\lambda}$ are the Lagrange multipliers computed using the surrogate model and active constraint set estimated from the surrogate model instead of the high-fidelity function. This approximate stationarity condition is similar to what would be obtained using a finite-difference gradient estimate with a fixed step size where the truncation error in the approximate derivative eventually dominates the stationarity measure, but in our case the sufficient decrease modification of the update parameter eventually dominates the stationarity measure (Boggs and Dennis 1976). For $\Delta_k \to 0$, we have from (9) that $\|\nabla f_{\text{high}}(\mathbf{x}) - \nabla m_k(\mathbf{x})\| \to 0$, and also $\|\hat{\lambda}(\mathbf{x}_k) - \lambda(\mathbf{x}_k)\| \to 0$. Therefore, we have first-order optimality as given in (18). In practice, the algorithm is terminated when the constraint violation is small, (19), first-order optimality is satisfied on the model, (21), and the trust region is small, say $\Delta_k < \varepsilon_2$ for a small $\varepsilon_2$.

### 2.6 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 is listed in Table 2 in Section 4. A key element of this algorithm is the logic to switch from the penalty function trust-region subproblem, (12), to the subproblem that uses the constraints explicitly, (11). Handling the constraints explicitly 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, $\| \left[ \mathbf{h}(\mathbf{x}_k)^{\mathsf{T}}, \mathbf{g}^{+}(\mathbf{x}_k)^{\mathsf{T}} \right] \| \leq \varepsilon$, or the linearized steps, $\delta_h$, satisfying $\mathbf{h}(\mathbf{x}) + \nabla \mathbf{h}(\mathbf{x})^{\mathsf{T}} \delta_h = 0$ 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 (2010) 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 in many situations may improve performance.

### 2.7 Theoretical considerations

This subsection discusses some performance limitations of the proposed algorithm as well as theoretical considerations needed for robustness.

This paper has not presented a formal convergence theory; at best, such a theory will apply only under many

**Algorithm 1:** Multifidelity Objective Trust-Region Algorithm

```
0:   Set initial parameters, $a$, $\alpha$, $\beta$, $\varepsilon$, $\varepsilon_2$, $\eta_0$, $\eta_1$, $\eta_2$, $\gamma_0$, $\gamma_1$, $\Delta_0$, $\Delta_{\max}$, and $\sigma_0$. Choose initial starting point $\mathbf{x}_0$, and
     build initial surrogate model $m_0(\mathbf{x})$ fully linear on $\{\mathbf{x} : \|\mathbf{x}-\mathbf{x}_0\| \leq \Delta_0\}$. Set $k = 0$.
1:   Update tolerance, $\tau_k = \min\left[\beta\varepsilon, \alpha\Delta_k\right]$.
2:   Choose and solve a trust-region subproblem:
     2a:  If the constraint violation is small, $\|\left[\mathbf{h}(\mathbf{x}_k)^\top\ \mathbf{g}^+(\mathbf{x}_k)^\top\right]\| \leq \varepsilon$, or the maximum linearized step to
          constraint feasibility for all active and violated constraints is smaller than the current trust region
          size, $\Delta_k$, then solve:
              $\min_{\mathbf{s}_k \in \mathbb{R}^n} \quad m_k(\mathbf{x}_k + \mathbf{s}_k)$
              $\text{s.t.} \quad \mathbf{h}(\mathbf{x}_k + \mathbf{s}_k) = 0$
              $\qquad \mathbf{g}(\mathbf{x}_k + \mathbf{s}_k) \leq 0$
              $\qquad \|\mathbf{s}_k\| \leq \Delta_k,$
          to convergence tolerance $\tau_k$.
     2b:  If 2a is not used or fails to converge to the required tolerance, solve the trust-region subproblem:
              $\min_{\mathbf{s}_k \in \mathbb{R}^n} \quad \hat{\Phi}_k(\mathbf{x}_k + \mathbf{s}_k, \sigma_k)$
              $\text{s.t.} \quad \|\mathbf{s}_k\| \leq \Delta_k.$
          to convergence tolerance $\tau_k$.
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 $\rho_k$,
         $\Delta_{k+1} = \begin{cases} \min\{\gamma_1\Delta_k, \Delta_{\max}\} & \text{if } \eta_1 \leq \rho_k \leq \eta_2 \\ \gamma_0\Delta_k & \text{if } \rho_k \leq \eta_0, \\ \Delta_k & \text{otherwise,} \end{cases}$
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, $\sigma_{k+1} = \max\left[e^{k+1/10}, 1/\Delta_{k+1}^{1.1}\right]$. Increment $k$.
10:  Check for convergence: if $\|\nabla m_k(\mathbf{x}_k) + A(\mathbf{x}_k)^\top\hat{\lambda}\| \leq \varepsilon$, $\|\left[\mathbf{h}(\mathbf{x}_k)^\top, \mathbf{g}^+(\mathbf{x}_k)^\top\right]\| \leq \varepsilon$, and $\Delta_k \leq \varepsilon_2$ the algorithm
     is converged, otherwise go to step 1.
```

restrictive assumptions. In addition, we can at best guarantee convergence to a near-optimal solution (i.e., optimality to an a priori tolerance level and not to stationarity). Forcing the trust region size to decrease every time the sufficient decrease condition is not satisfied means that if the projection of the gradient onto the feasible domain is less than the parameter $a$, then the algorithm can fail to make progress. This limitation is presented in (21); however, it is not seen as severe because $a$ can be set at any value arbitrarily close to (but strictly greater than) zero.

A second limitation is that the model calibration strategy, generating a fully linear model, is theoretically only guaranteed to be possible for functions that are twice continuously differentiable and have Lipschitz-continuous first derivatives. Though this assumption may seem to limit some desired opportunities for derivative-free optimization of functions with noise, noise with certain characteristics like that discussed in Conn et al. (2009b, Section 9.3), or the case of models with dynamic accuracy as in Conn et al. (2000, Section 10.6) can be accommodated in this framework. Approaches such as those in Moré and Wild (2010) may be used to characterize the noise in a specific problem. However, our algorithm applies even in the case of general noise, where no guarantees can be made on the quality of the surrogate models. As will be shown in the test problem in Section 4, in such a case our approach exhibits robustness and significant advantages over gradient-based methods that are susceptible to poor gradient estimates.

Algorithm 1 is more complicated than conventional gradient-based trust-region algorithms using quadratic penalty functions, such as the algorithm in Conn et al. (2000, Chapter 14). There are three sources of added complexity. First, our algorithm increases the penalty parameter after each trust-region subproblem as opposed to after a completed minimization of the penalty function. This aspect can significantly reduce the number of required high-fidelity evaluations, but adds complexity to the algorithm. Second, our algorithm switches between two trust region subproblems to avoid numerical issues associated with the conditioning of the quadratic penalty function Hessian. The third reason for added complexity is that in derivative-free optimization the size of the trust-region must decrease to zero in order to demonstrate optimality. This aspect serves to add unwanted coupling between the penalty parameter and the size of the trust region, which must be handled appropriately. We now discuss how these two aspects of the algorithm place important constraints on the penalty parameter $\sigma_k$.

The requirements for the penalty parameter, $\sigma_k$, are that (i) $\lim_{k\to\infty}\sigma_k\Delta_k = \infty$, (ii) $\lim_{k\to\infty}\sigma_k\Delta_k^2 = 0$, and (iii) $\sum_{k=0}^{\infty} 1/\sigma_k$ is finite. The lower bound for the growth of $\sigma_k$, that (i) $\lim_{k\to\infty}\sigma_k\Delta_k = \infty$, comes from the properties of the minima of quadratic penalty functions presented in Conn et al. (2000, Chapter 14), properties of a fully linear model, and (13). If $\mathbf{x}_k$ is at an approximate minimizer of the surrogate quadratic penalty function, (8), then a bound on the constraint violation is

$$\left\| \left[\mathbf{h}(\mathbf{x}_k)^{\top}\ \mathbf{g}^{+}(\mathbf{x}_k)^{\top}\right] \right\| \leq \frac{\kappa_1(\max\{\alpha\Delta_k, a\} + \kappa_g\Delta_k) + \|\lambda(\mathbf{x}^*)\| + \kappa_2\|\mathbf{x}_k - \mathbf{x}^*\|}{\sigma_k}, \tag{22}$$

where $\mathbf{x}^*$ is a KKT point of (1), and $\kappa_1$, $\kappa_2$ are finite positive constants. If $\{\sigma_k\Delta_k\}$ diverges then an iteration exists where both the bound, (22), holds (the trust region size must be large enough that a feasible point exists in the interior) and the constraint violation is less than the given tolerance $\varepsilon$. This enables the switching between the two trust region subproblems, (12) and (11). The upper bound for the growth of $\sigma_k$, that (ii) $\lim_{k\to\infty}\sigma_k\Delta_k^2 = 0$, comes from the smoothness of the quadratic penalty function. To establish an upper bound for the Hessian 2-norm for the subproblem in (12) 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 $\Phi(\mathbf{x}_k, \sigma_k)$, $\tilde{\Phi}(\mathbf{x}_k + \mathbf{p}, \sigma_k)$. If $\kappa_{fg}$ is the Lipschitz constant for $\nabla f_{\text{high}}(\mathbf{x})$, $\kappa_{cm}$ is the maximum Lipschitz constant for the constraints, and $\kappa_{cgm}$ is the maximum Lipschitz constant for a constraint gradient, we can show that

$$\begin{aligned}\|\Phi(\mathbf{x}_k+\mathbf{p}, \sigma_k) - \tilde{\Phi}(\mathbf{x}_k+\mathbf{p}, \sigma_k)\| &\leq \Big[\kappa_{fg} + \sigma_k\Big(\kappa_{cm}\left\|\left[\mathbf{h}(\mathbf{x}_k)^{\top}\ \mathbf{g}^{+}(\mathbf{x}_k)^{\top}\right]\right\| \\ &\qquad + \kappa_{cgm}\|A(\mathbf{x}_k)\|\,\|\mathbf{p}\| + \kappa_{cm}\kappa_{cgm}\|\mathbf{p}\|^2\Big)\Big] \\ &\quad \times \|\mathbf{p}\|^2.\end{aligned} \tag{23}$$

We have a similar result for $\hat{\Phi}(\mathbf{x}, \sigma_k)$ by replacing $\kappa_{fg}$ with the sum $\kappa_{fg} + \kappa_g$, using the definition of a fully linear model, and by bounding $\|\mathbf{p}\|$ by $\Delta_k$. Therefore, we may show the error in a linearized prediction of the surrogate model will go to zero and that Lipschitz-type smoothness is ensured provided that the sequence $\{\sigma_k\Delta_k^2\}$ converges to zero regardless of the constraint violation.

The final requirement, that (iii) $\sum_{k=0}^{\infty} 1/\sigma_k$ is finite, comes from the need for the size of the trust region to go to zero in gradient-free optimization. The sufficient decrease condition, that $\rho_k = 0$ unless $\hat{\Phi}(\mathbf{x}_k, \sigma_k) - \hat{\Phi}(\mathbf{x}_k + \mathbf{s}_k, \sigma_k) \geq a\Delta_k$ ensures that the trust region size decreases unless the change in the merit function $\Phi(\mathbf{x}_k, \sigma_k) - \Phi(\mathbf{x}_k + \mathbf{s}_k, \sigma_k) \geq \eta_0 a\Delta_k$. This provides an upper bound on the total number of times in which the size of the trust region is kept constant or increased. We have assumed that the merit function is bounded from below and also that the trust-region iterates remain within a level-set $L(\mathbf{x}_0, \sigma_0)$ as defined by (4). We now consider the merit function written in an alternate form,

$$\Phi(\mathbf{x}_k, \sigma_k) = f_{\text{high}}(\mathbf{x}_k) + \frac{\sigma_k}{2}\left\|\left[\mathbf{h}(\mathbf{x}_k)^{\top}, \mathbf{g}^{+}(\mathbf{x}_k)^{\top}\right]\right\|^2. \tag{24}$$

From (22), if $\sigma_k$ is large enough such that the bound on the constraint violation is less than unity, we may use the bound on the constraint violation in (22) to show that an upper bound on the total remaining change in the merit function is

$$f_{\text{high}}(\mathbf{x}_k) - \min_{\mathbf{x}\in L(\mathbf{x}_0,\sigma_0)} f_{\text{high}}(\mathbf{x}) + \frac{\left[\kappa_1(\max\{\alpha\Delta_k, a\}+\kappa_g\Delta_k)+\|\lambda(\mathbf{x}^*)\|+\kappa_2\|\mathbf{x}_k-\mathbf{x}^*\|\right]^2}{\sigma_k}. \tag{25}$$

Each term in the numerator is bounded from above because $\Delta_k$ is always bounded from above by $\Delta_{\max}$, $\|\lambda(\mathbf{x}^*)\|$ is bounded from above because $\mathbf{x}^*$ is a regular point, and $\|\mathbf{x}_k - \mathbf{x}^*\|$ is bounded from above because $L(\mathbf{x}_0, \sigma_0)$ is a compact set. Therefore, if the series $\{1/\sigma_k\}$ has a finite sum, then the total remaining improvement in the merit function is finite. Accordingly, the sum of the series $\{\Delta_k\}$ must be finite, and $\Delta_k \to 0$ as $k \to \infty$. The prescribed sequence for $\{\sigma_k\}$ in our algorithm satisfies these requirements for a broad range of problems.

## 3 Multifidelity objective and constraint optimization

This section considers a more general constrained optimization problem with a computationally expensive objective function and computationally expensive constraints. The specific problem considered is where the gradients for both the expensive objective and expensive constraints are either unavailable, unreliable or expensive to estimate. Accordingly, the multifidelity optimization problem in (1) is augmented with the high-fidelity constraint, $c_{\text{high}}(\mathbf{x}) \le 0$. In addition, we have a low-fidelity estimate of this constraint function, $c_{\text{low}}(\mathbf{x})$, which estimates the same metric as $c_{\text{high}}(\mathbf{x})$, 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,

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & f_{\text{high}}(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{h}(\mathbf{x}) = 0 \\ & \mathbf{g}(\mathbf{x}) \le 0 \\ & c_{\text{high}}(\mathbf{x}) \le 0, \end{aligned} \tag{26}$$

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 given in Section 2.1 are made for the functions presented in this formulation. It is also necessary to make an assumption similar to that in Section 2.2, 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 (Rvachev 1963; Papalambros and Wilde 2000).

The optimization problem in (26) is solved in two phases. First, the multifidelity optimization method presented in Section 2 is used to find a feasible point, and then an interior point formulation is used to find a minimum of the optimization problem in (26). The interior point formulation is presented in Section 3.2 and the numerical implementation is presented in Section 3.3.

### 3.1 Finding a feasible point

This algorithm begins by finding a point that is feasible with respect to all of the constraints by applying Algorithm 1 to the optimization problem

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad & c_{\text{high}}(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{h}(\mathbf{x}) = 0 \\ & \mathbf{g}(\mathbf{x}) \le 0, \end{aligned} \tag{27}$$

until a point that is feasible with respect to the constraints in (26) is found. If this optimization problem is unconstrained (i.e., there are no constraints $\mathbf{h}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$) then the trust-region algorithm of Conn et al. (2009a) is used with the multifidelity calibration method of March and Willcox (2010). The optimization problem in (27) may violate one of the assumptions for Algorithm 1 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 3.3.

### 3.2 Interior point trust-region method

Once a feasible point is found we minimize the high-fidelity objective function ensuring that we never again violate the constraints, that is we solve (26). This is accomplished in two steps, first by solving trust region subproblems that use fully linear surrogate models for both the high-fidelity objective function and the high-fidelity constraint. Second, the trust region step is evaluated for feasibility and any infeasible step is rejected. The surrogate model for the objective function is $m_k(\mathbf{x})$ as defined in (2). For the constraint, the surrogate model, $\bar{m}_k(\mathbf{x})$, is defined as

$$\bar{m}_k(\mathbf{x}) = c_{\text{low}}(\mathbf{x}) + \bar{e}_k(\mathbf{x}). \tag{28}$$

From the definition of a fully linear model, (9) and (10), $\bar{m}_k(\mathbf{x})$ satisfies

$$\left\|\nabla c_{\text{high}}(\mathbf{x}) - \nabla \bar{m}_k(\mathbf{x})\right\| \le \kappa_{cg}\Delta_k \quad \forall \mathbf{x} \in \mathscr{B}_k, \tag{29}$$

$$\left|c_{\text{high}}(\mathbf{x}) - \bar{m}_k(\mathbf{x})\right| \le \kappa_c \Delta_k^2 \quad \forall \mathbf{x} \in \mathscr{B}_k. \tag{30}$$

In addition, we require that our procedure to construct fully linear models ensures that at the current design iterate, the fully linear models exactly interpolate the function they are modeling,

$$m_k(\mathbf{x}_k) = f_{\text{high}}(\mathbf{x}_k), \tag{31}$$

$$\bar{m}_k(\mathbf{x}_k) = c_{\text{high}}(\mathbf{x}_k). \tag{32}$$

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

The trust-region subproblem is

$$\begin{aligned} \min_{\mathbf{s}_k \in \mathbb{R}^n} \quad & m_k(\mathbf{x}_k + \mathbf{s}_k) \\ \text{s.t.} \quad & \mathbf{h}(\mathbf{x}_k + \mathbf{s}_k) = 0 \\ & \mathbf{g}(\mathbf{x}_k + \mathbf{s}_k) \le 0 \\ & \bar{m}_k(\mathbf{x}_k + \mathbf{s}_k) \le \max\{c_{\text{high}}(\mathbf{x}_k), -w\Delta_k\} \\ & \|\mathbf{s}_k\| \le \Delta_k. \end{aligned} \tag{33}$$

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, $\max\{c_{\text{high}}(\mathbf{x}_k), -w\Delta_k\}$, ensures that the constraint is initially feasible and that protection of constraint violation decreases to zero as the number of iterations increase to infinity. The constant $w$ must be greater than $\alpha$, which is defined as part of the termination tolerance $\tau_k$ in (13). The trust-region subproblem is solved to the same termination tolerance as the multifidelity objective function formulation, $\|\nabla_x \mathfrak{L}_k\| \le \tau_k$, where $\mathfrak{L}_k$ is the Lagrangian of (33).

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

$$\mathbf{x}_{k+1} = \begin{cases} \mathbf{x}_k + \mathbf{s}_k & \text{if } f_{\text{high}}(\mathbf{x}_k) > f_{\text{high}}(\mathbf{x}_k + \mathbf{s}_k) \\ & \quad \text{and } c_{\text{high}}(\mathbf{x}_k + \mathbf{s}_k) \le 0 \\ \mathbf{x}_k & \text{otherwise,} \end{cases} \tag{34}$$

with $\mathbf{h}(\mathbf{x}_k + \mathbf{s}_k) = 0$ and $\mathbf{g}(\mathbf{x}_k + \mathbf{s}_k) \le 0$ already satisfied in (33). 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 $\Delta_k$,

$$\Delta_{k+1} = \begin{cases} \min\{\gamma_1\Delta_k, \Delta_{\max}\} & \text{if } f_{\text{high}}(\mathbf{x}_k + \mathbf{s}_k) \\ & \quad - f_{\text{high}}(\mathbf{x}_k) \ge a\Delta_k \\ & \quad \text{and } c_{\text{high}}(\mathbf{x}_k + \mathbf{s}_k) \le 0 \\ \gamma_0\Delta_k & \text{otherwise.} \end{cases} \tag{35}$$

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 $\mathscr{B}_{k+1}$ having center $\mathbf{x}_{k+1}$ and size $\Delta_{k+1}$. The new fully linear models are constructed using the procedure of Wild et al. (2008) with the calibration technique of March and Willcox (2010).

### 3.3 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 is listed in Table 2 in Section 4. 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. The convergence of a penalty function requires that the objective function is bounded from below, therefore a more general problem than (27) to find an initial feasible point is to use,

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} \quad & \left[\max\{c_{\text{high}}(\mathbf{x}) + d, 0\}\right]^2 \\ \text{s.t.} \quad & \mathbf{h}(\mathbf{x}) = 0 \\ & \mathbf{g}(\mathbf{x}) \le 0. \end{aligned} \tag{36}$$

The maximization in the objective prevents the need for the high-fidelity constraint to be bounded from below, and the constraint violation is squared to ensure the gradient of the objective is continuous. 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 $d \ge \kappa_c\Delta_k^2$ and $\bar{m}_k(\mathbf{x}) + d = 0$ then (30) guarantees that $c_{\text{high}}(\mathbf{x}) \le 0$.

There is a similar consideration in the solution of (33), 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

**Algorithm 2:** Multifidelity Objective and Constraint Trust-Region Algorithm

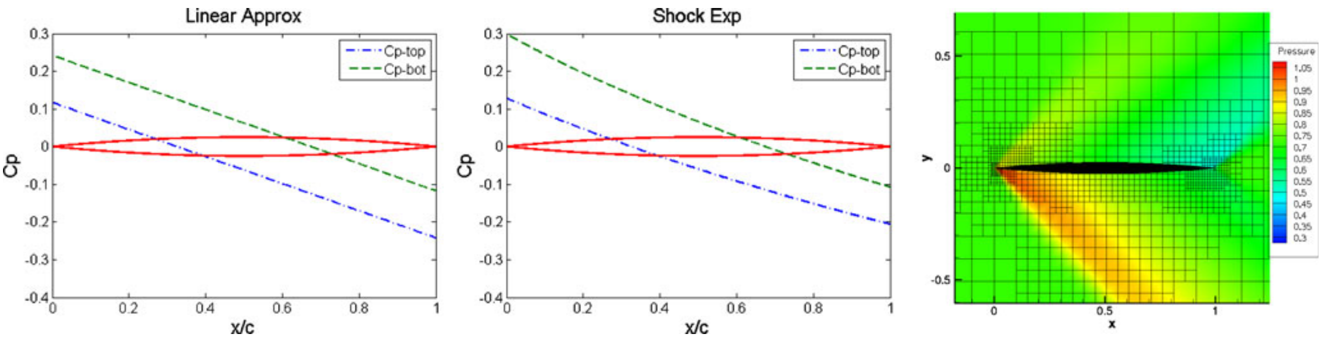
```
-1: Find a feasible design vector using Algorithm 1 to iterate on:
        $\min_{\mathbf{x}\in\mathbb{R}^n} \; [\max\{c_{\text{high}}(\mathbf{x})+d,0\}]^2$
        $\text{s.t.} \; \mathbf{h}(\mathbf{x})=0$
        $\mathbf{g}(\mathbf{x})\le 0,$
    -1a: Return all evaluations of $c_{\text{high}}(\mathbf{x})$.
0:  Set initial parameters, $a$, $\alpha$, $\beta$, $d$, $w$, $\varepsilon$, $\varepsilon_2$, $\gamma_0$, $\gamma_1$, $\Delta_0$, $\Delta_{\max}$, and $\sigma_0$. Build initial surrogate models $m_0(\mathbf{x})$, $\bar{m}_0(\mathbf{x})$ fully linear on $\{\mathbf{x} : \|\mathbf{x}-\mathbf{x}_0\|\le\Delta_0\}$, where $\mathbf{x}_0$ is the terminal point of step -1. Set $k=0$.
1:  Update tolerance, $\tau_k=\min[\beta\varepsilon,\alpha\Delta_k]$.
2:  Solve the trust-region subproblem:
        $\min_{\mathbf{s}_k\in\mathbb{R}^n} \; m_k(\mathbf{x}_k+\mathbf{s}_k)$
        $\text{s.t.} \; \mathbf{h}(\mathbf{x}_k+\mathbf{s}_k)=0$
        $\mathbf{g}(\mathbf{x}_k+\mathbf{s}_k)\le 0$
        $\bar{m}_k(\mathbf{x}_k+\mathbf{s}_k)\le\min\{c_{\text{high}}(\mathbf{x}_k),-w\Delta_k\}$
        $\|\mathbf{s}_k\|\le\Delta_k.$
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:
        $\mathbf{x}_{k+1}=\begin{cases}\mathbf{x}_k+\mathbf{s}_k & \text{if } f_{\text{high}}(\mathbf{x}_k)>f_{\text{high}}(\mathbf{x}_k+\mathbf{s}_k) \text{ and } c_{\text{high}}(\mathbf{x}_k+\mathbf{s}_k)\le 0\\ \mathbf{x}_k & \text{otherwise.}\end{cases}$
5:  Update the trust region size according to:
        $\Delta_{k+1}=\begin{cases}\min\{\gamma_1\Delta_k,\Delta_{\max}\} & \text{if } f_{\text{high}}(\mathbf{x}_k+\mathbf{s}_k)-f_{\text{high}}(\mathbf{x}_k)\ge a\Delta_k \text{ and } c_{\text{high}}(\mathbf{x}_k+\mathbf{s}_k)\le 0\\ \gamma_0\Delta_k & \text{otherwise.}\end{cases}$
6:  Create new models $m_{k+1}(\mathbf{x})$ and $\bar{m}_{k+1}(\mathbf{x})$ fully linear on $\{\mathbf{x} : \|\mathbf{x}-\mathbf{x}_{k+1}\|\le\Delta_{k+1}\}$. Increment $k$.
7:  Check for convergence, if the trust region constraint is inactive,
    $\|\nabla m(\mathbf{x}_k)+A(\mathbf{x}_k)^{\mathsf{T}}\hat{\lambda}(\mathbf{x}_k)+\nabla\bar{m}(\mathbf{x}_k)\lambda_{\bar{m}}\|\le\varepsilon$, and $\Delta_k\le\varepsilon_2$, the algorithm is converged, otherwise go to step 1.
```

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 ensure the solution is an interior point. This feature avoids difficulties with the convergence of the trust-region iterates.

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 $\gamma_0$. This will help to ensure that the trust-region does not decrease to zero at a suboptimal point.

## 4 Supersonic airfoil design test problem

This section presents results of the two multifidelity optimization algorithms on a supersonic airfoil design problem.



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

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

### 4.1 Problem setup

The supersonic 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 (Aftosmis 1997). 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.

**Table 1** 5% thick biconvex airfoil results comparison at Mach 1.5 and 2° angle of attack

| | 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% |

Percent difference is taken with respect to the Cart3D results

We first present single-fidelity results using state-of-the-art derivative-free methods. Then 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 4.3, the airfoil drag is minimized using the constrained multifidelity objective function formulation presented in Section 2 with only the simple geometric constraints. In the second example, Section 4.4, the airfoil lift-to-drag ratio is maximized subject to a constraint that the drag coefficient is less than 0.01, where the constraint is handled with the multifidelity framework presented in Section 3. In the final example, Section 4.5, the airfoil lift-to-drag ratio is maximized subject to the constrained drag coefficient and both the objective function and the constraints are handled with the multifidelity framework presented in Section 3. The initial airfoils for all problems are randomly generated and likely will not satisfy the constraints.

The three multifidelity airfoil problems are solved with four alternative optimization algorithms: Sequential Quadratic Programming (SQP) (MathWorks, Inc. 2010); a first-order consistent multifidelity trust-region algorithm that uses a SQP formulation and an additive correction (Alexandrov et al. 2001); the high-fidelity-gradient-free approach presented in this paper using a Gaussian radial basis function and a fixed spatial correlation parameter, $\xi = 2$; and the approach presented in this paper using a maximum likelihood estimate to find an improved correlation length, $\xi = \xi^*$. The Gaussian correlation functions used in these results are all isotropic. An anisotropic correlation function (i.e., choosing a correlation length for each direction in the design space) may speed convergence of this algorithm and reduce sensitivity to Hessian conditioning. The parameters used for the optimization algorithm are presented in Table 2. The fully linear models are constructed using the procedure of Wild et al. (2008) with the calibration

**Table 2** List of constants used in the algorithm

| Constant | Description | Value |
|---|---|---|
| $a$ | Sufficient decrease constant | $1 \times 10^{-4}$ |
| $\alpha$ | Convergence tolerance multiplier | $1 \times 10^{-2}$ |
| $\beta$ | Convergence tolerance multiplier | $1 \times 10^{-2}$ |
| $d$ | Artificial lower bound for constraint value | 1 |
| $w$ | Constraint violation conservatism factor | 0.1 |
| $\varepsilon, \varepsilon_2$ | Termination tolerance | $5 \times 10^{-4}$ |
| $\gamma_0$ | Trust region contraction ratio | 0.5 |
| $\gamma_1$ | Trust region expansion ratio | 2 |
| $\eta_0$ | Trust region contraction criterion | 0.25 |
| $\eta_1, \eta_2$ | Trust region expansion criterion | 0.75, 2.0 |
| $\Delta_0$ | Initial trust region radius | 1 |
| $\Delta_{\max}$ | Maximum trust region size | 20 |
| $\sigma_k$ | Penalty parameter | $\max\left[e^{k/10}, 1/\Delta_k^{1.1}\right]$ |
| $\delta_x$ | Finite difference step | $1 \times 10^{-5}$ |

All parameters used in constructing the radial basis function error model are the same as in March and Willcox (2010). These parameter values are based on recommendations for unconstrained trust-region algorithms and through numerical testing appear to have good performance for an assortment of problems

technique of March and Willcox (2010) and the parameters stated therein.

### 4.2 Single-fidelity derivative-free optimization

For benchmark purposes, we first solve the airfoil optimization problem using three single-fidelity gradient-free optimization methods: the Nelder–Mead simplex algorithm (Nelder and Mead 1965), the global optimization method DIRECT (Jones et al. 1993), and the constrained gradient-free optimizer, COBYLA (Powell 1994, 1998). The test case is to minimize the drag of supersonic airfoil estimated with a panel method, subject to the airfoil having positive thickness everywhere and at least 5% thickness to chord ratio. This test case is a lower-fidelity version of the example in Section 4.3. Nelder–Mead simplex and DIRECT are unconstrained optimizers that use a quadratic penalty function known to perform well on this problem (March and Willcox 2010), while COBYLA handles the constraints explicitly. On this problem, starting from ten random initial airfoils the best observed results were 5,170 function evaluations for the Nelder–Mead simplex algorithm, over 11,000 evaluations for DIRECT, and 6,284 evaluations for COBYLA. Such high numbers of function evaluations mean that these single-fidelity gradient-free algorithms are too expensive for use with an expensive forward solver, such as Cart3D.

### 4.3 Multifidelity objective function results

This section presents optimization results in terms of the number of high-fidelity 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 2.

The average numbers of high-fidelity function evaluations required to find a locally optimal design starting from random initial airfoils are presented in Table 3. The results show that our 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. In addition, Fig. 2 compares the objective function and constraint violation histories verse the number of high-fidelity evaluations for these methods as well as DIRECT and Nelder-Mead simplex from a representative random initial airfoil for the case with the shock-expansion method as the high-fidelity function. For the single-fidelity optimization using Cart3D, the convergence results were highly sensitive to the finite difference step length. The step size required tuning, and the step with the highest success rate was used. A reason for this is that the initial airfoils were randomly generated, and the convergence tolerance

**Table 3** The average number of high-fidelity function evaluations to minimize the drag of a supersonic airfoil with only geometric constraints

| 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%) |

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

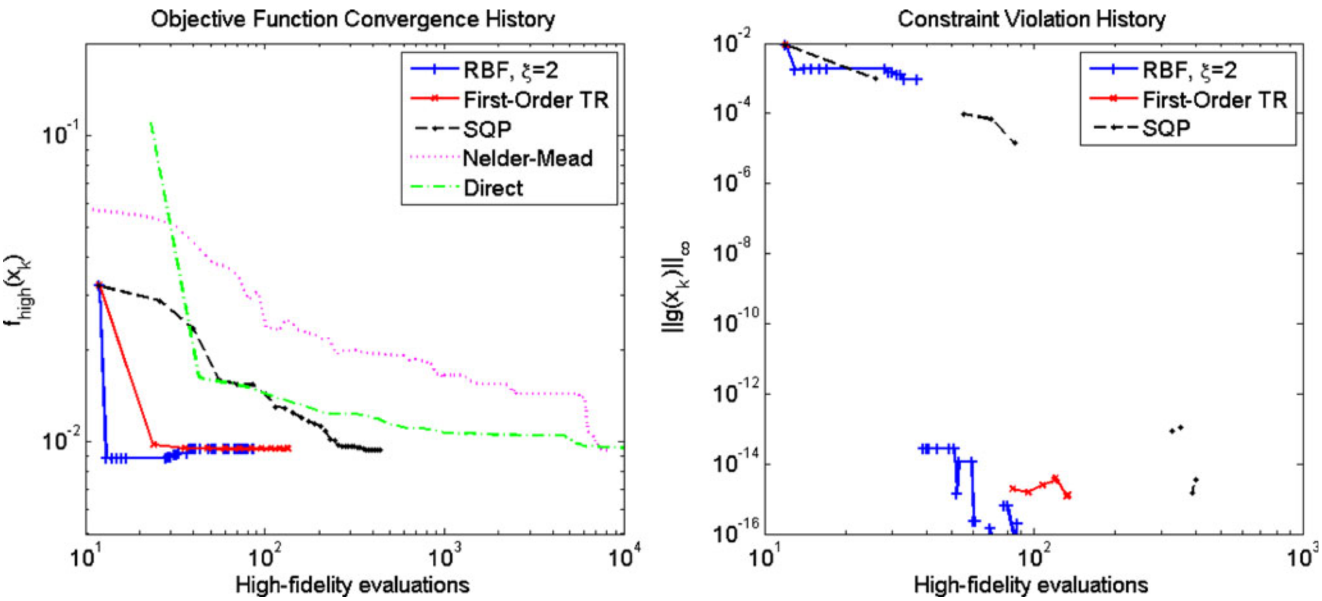
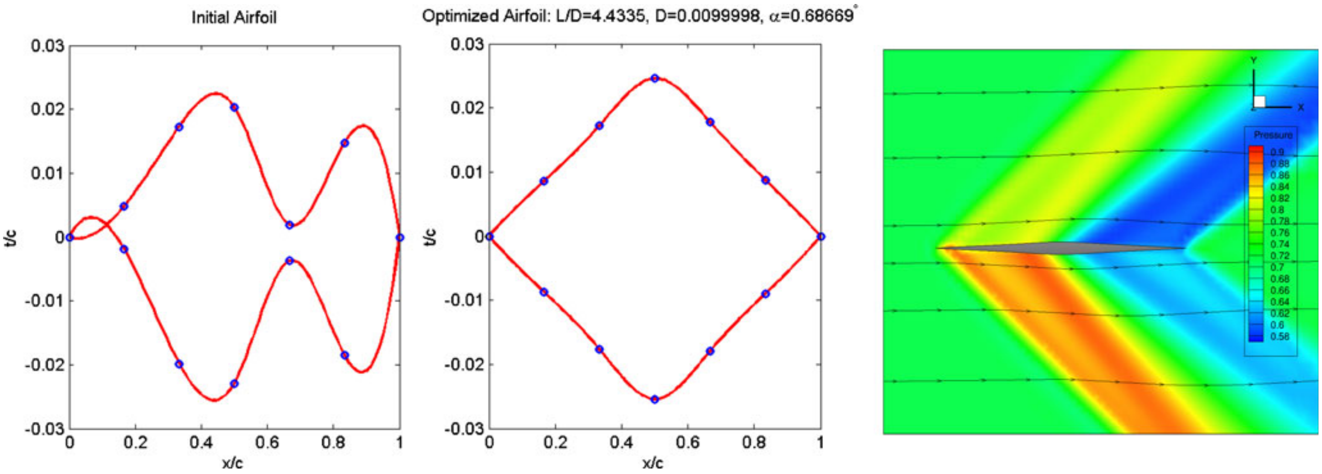


**Fig. 2** Convergence history for minimizing the drag of an airfoil using the shock-expansion theory method as the high-fidelity function subject to only geometric constraints. The methods presented are our calibration approach, a first-order consistent multifidelity trust-region algorithm, sequential quadratic programming, DIRECT, and Nelder-Mead simplex. Both DIRECT and Nelder-Mead simplex use a fixed penalty function to handle the constraints, so only an objective function value is shown. COBYLA and BOBYQA were attempted, but failed to find the known solution to this problem. On the constraint violation plot, missing points denote a feasible iterate, and the sudden decrease in the constraint violation for the RBF calibration approach at 37 high-fidelity evaluations (13th iteration, $\sigma_k = 9.13 \times 10^5$) is when the algorithm switches from solving (12) to solving (11)

**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

| 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%) |

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



(a) Initial airfoil.

(b) Optimal shock-expansion theory airfoil.

(c) Optimal Cart3D flow solution.

**Fig. 3** Initial airfoil and supersonic airfoil with the maximum lift-to-drag ratio having drag less than 0.01 and 5% thickness at Mach 1.5

**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

| | 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%) |
| Objective: | Cart3D | Panel method | 1168* (−) | 97 (−92%) | 104 (−91%) | 112 (−90%) |
| Constraint: | Cart3D | Panel method | 2335* (−) | 97 (−96%) | 115 (−95%) | 128 (−94%) |

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

of Cart3D for airfoils with sharp peaks and negative thickness was large compared with the airfoils near the optimal design. This sensitivity of gradient-based optimizers to the finite difference step length highlights the benefit of gradient-free approaches, especially when constraint gradient estimates become poor.

### 4.4 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 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 3. Two cases are examined: in the first the shock-expansion method models the high-fidelity constraint and the panel method models the low-fidelity constraint; in the second case, Cart3D models the high-fidelity constraint and the panel method models 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 multifidelity techniques developed in this paper. A significant decrease (almost 90%) in the number of high-fidelity function evaluations is observed when compared with SQP. Performance is almost the same as the first-order consistent trust-region algorithm.

### 4.5 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 3. 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. The optimal airfoils are shown in Fig. 3. Table 5 presents the number of high-fidelity 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 (about 90%) 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.

## 5 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. For the supersonic airfoil design example considered here, the multifidelity methods resulted in approximately 90% reduction in the number of high-fidelity function evaluations compared to solution with a single-fidelity sequential quadratic programming method. In addition, the multifidelity methods performed similarly to a first-order consistent trust-region algorithm with gradients estimated using finite difference approximations. This shows that derivative-free multifidelity methods provide significant opportunity for optimization of computationally expensive functions without available gradients.

The behavior of the gradient-free algorithms presented is slightly atypical of nonlinear programming methods. For example, the convergence rate of these gradient-free algorithms is rapid initially and then slows when close to an optimal solution. In contrast, convergence for a gradient-based method is often initially slow and then accelerates when close to an optimal solution (e.g., as approximate Hessian information becomes more accurate in a quasi-Newton approach). Also, gradient-based optimizers typically find the local optimal solution nearest the initial design. Although by virtue of the physics involved, the presented examples have unique optimal solutions, in a general problem the local optimum to which these gradient-free algorithms converge may not be the one in the immediate vicinity of the initial iterate. An example of this behavior is the case when the initial iterate is itself a local optimum, the surrogate model may not capture this fact and the iterate may move to a different point in the design space with a lower function value.

In the case of hard constraints, or when the objective function fails to exist if the constraints are violated, it is still possible to use Algorithm 2. 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 $\mathbb{R}^n$. When the objective function can be evaluated outside the feasible region, the constraints do not influence the construction of the surrogate model. However, when the objective function does not exist where the constraints are violated, then the points used to construct the surrogate model must all be feasible and this restricts the shape of the feasible region. Specifically, 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.

Lastly, we comment on the applicability of the proposed multifidelity approach. Though this paper presents no formal convergence theory, and at best that theory will only apply if many restrictive assumptions hold (for example, assumptions on smoothness, constraint qualification, and always using a fully linear surrogate) our numerical experiments indicate the robustness of the approach in application to a broader class of problems. For example, the Cart3D CFD model employed in our case studies does not satisfy the Lipschitz continuity requirements, due to finite convergence tolerances in determining the CFD solution; however, with the aid of the smooth calibrated surrogates combined with the trust-region model management, our multifidelity method is successful in finding locally optimal solutions. Another example is a high-fidelity optimal solution for which constraint qualification conditions are not satisfied. In the algorithms presented, the design vector iterate 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 may not be satisfied.

In summary, this paper has presented a multifidelity optimization algorithm that does not require estimating gradients of high-fidelity functions, enables the use of multiple low-fidelity models, enables optimization of functions with hard constraints, and exhibits robustness over a broad class of optimization problems, even when non-smoothness is present in the objective function and/or constraints. For airfoil design problems, this approach has been shown to perform similarly in terms of the number of function evaluations to finite-difference-based multifidelity optimization methods. This suggests that the multifidelity derivative-free approach is a promising alternative for the wide range of problems where finite-difference gradient approximations are unreliable.

**Acknowledgments** The authors gratefully acknowledge support from NASA Langley Research Center contract NNL07AA33C, technical monitor Natalia Alexandrov, and a National Science Foundation graduate research fellowship. In addition, we wish to thank Michael Aftosmis and Marian Nemec for support with Cart3D.

## References

Aftosmis MJ (1997) Solution adaptive cartesian grid methods for aerodynamic flows with complex geometries. In: 28th computational fluid dynamics lecture series, von Karman Institute for Fluid Dynamics, Rhode-Saint-Genése, Belgium, lecture series 1997-02

Alexandrov N, Lewis R, Gumbert C, Green L, Newman P (1999) Optimization with variable-fidelity models applied to wing design. Tech. Rep. CR-209826, NASA

Alexandrov N, Lewis R, Gumbert C, Green L, Newman P (2001) Approximation and model management in aerodynamic optimization with variable-fidelity models. AIAA J 38(6):1093–1101

Audet C, Dennis Jr JE (2004) A pattern search filter method for nonlinear programming without derivatives. SIAM J Optim 14(4):980–1010

Bertsekas DP (1999) Nonlinear programming, 2nd edn. Athena Scientific

Boggs PT, Dennis Jr JE (1976) A stability analysis for perturbed nonlinear iterative methods. Math Comput 30(134):199–215

Booker AJ, Dennis Jr JE, Frank PD, Serafini DB, Torczon V, Trosset MW (1999) A rigorous framework for optimization of expensive functions by surrogates. Struct Optim 17(1):1–13

Castro J, Gray G, Giunta A, Hough P (2005) Developing a computationally efficient dynamic multilevel hybrid optimization scheme using multifidelity model interactions. Tech. Rep. SAND2005-7498, Sandia

Conn A, Gould N, Toint P (2000) Trust-region methods. MPS/SIAM series on optimization. Society for Industrial and Applied Mathematics, Philadelphia

Conn A, Scheinberg K, Vicente L (2008) Geometry of interpolation sets in derivative free optimization. Math Program 111(1–2):141–172

Conn A, Scheinberg K, Vicente L (2009a) Global convergence of general derivative-free trust-region algorithms to first- and second-order critical points. SIAM J Optim 20(1):387–415

Conn AR, Scheinberg K, Vicente LN (2009b) Introduction to derivative-free optimization. MPS/SIAM series on optimization. Society for Industrial and Applied Mathematics, Philadelphia

Jones D (2001) A taxonomy of global optimization methods based on response surfaces. J Glob Optim 21:345–383

Jones D, Perttunen CD, Stuckmann BE (1993) Lipschitzian optimization without the lipschitz constant. J Optim Theory Appl 79(1):157–181

Jones D, Schonlau M, Welch W (1998) Efficient global optimization of expensive black-box functions. J Glob Optim 13:455–492

Kennedy M, O'Hagan A (2000) Predicting the output from a complex computer code when fast approximations are available. Biometrika 87(1):1–13

Kolda TG, Lewis RM, Torczon V (2003) Optimization by direct search: new perspectives on classical and modern methods. SIAM Rev 45(3):385–482

Kolda TG, Lewis RM, Torczon V (2006) A generating set direct search augmented lagrangian algorithm for optimization with a combination of general and linear constraints. Tech. Rep. SAND2006-5315, Sandia

Lewis RM, Torczon V (2010) 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

Liuzzi G, Lucidi S (2009) A derivative-free algorithm for inequality constrained nonlinear programming via smoothing of an $l_\infty$ penalty function. SIAM J Optim 20(1):1–29

March A, Willcox K (2010) A provably convergent multifidelity optimization algorithm not requiring high-fidelity gradients. In: 6th AIAA multidisciplinary design optimization specialist conference, Orlando, FL, AIAA 2010-2912

MathWorks, Inc. (2010) Constrained nonlinear optimization. Optimization toolbox user's guide, v. 5

Moré JJ, Wild SM (2010) Estimating derivatives of noisy simulations. Tech. Rep. Preprint ANL/MCS-P1785-0810, Mathematics and Computer Science Division

Nelder JA, Mead RA (1965) A simplex method for function minimization. Comput J 7:308–313

Nocedal J, Wright S (2006) Numerical optimization, 2nd edn. Springer, New York

Papalambros PY, Wilde DJ (2000) Principles of optimal design, 2nd edn. Cambridge University Press, Cambridge

Powell MJD (1994) A direct search optimization method that models the objective and constraint functions by linear interpolation. In: Gomez S, Hennart J-P (eds) Advances in optimization and numerical analysis, vol 7. Kluwer Academic, Dordrecht, pp 51–67

Powell MJD (1998) Direct search algorithms for optimization calculations. Acta Numer 7:287–336

Rajnarayan D, Haas A, Kroo I (2008) A multifidelity gradient-free optimization method and application to aerodynamic design. In: 12th AIAA/ISSMO multidisciplinary analysis and optimization conference, Victoria, British Columbia, AIAA 2008-6020

Rvachev VL (1963) On the analytical description of some geometric objects. Tech. Rep. 4, Reports of Ukrainian Academy of Sciences (in Russian)

Sasena MJ, Papalambros P, Goovaerts P (2002) Exploration of metamodeling sampling criteria for constrained global optimization. Eng Optim 34(3):263–278

Wild S (2009) Derivative-free optimization algorithms for computationally expensive functions. PhD thesis, Cornell University

Wild S, Shoemaker CA (2009) Global convergence of radial basis function trust-region algorithms. Tech. Rep. Preprint ANL/MCS-P1580-0209, Mathematics and Computer Science Division

Wild S, Regis R, Shoemaker C (2008) ORBIT: optimization by radial basis function interpolation in trust-regions. SIAM J Sci Comput 30(6):3197–3219