11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference
6 - 8 September 2006, Portsmouth, Virginia

AIAA 2006-7114

Multifidelity Optimization for Variable-Complexity Design

T. D. Robinson[‖*], K. E. Willcox[‖†], M. S. Eldred[**‡], R. Haimes[‖§]

[‖] *Aerospace Computational Design Laboratory*
Massachusetts Institute of Technology, Cambridge, MA, 02139

[**] *Optimization and Uncertainty Estimation Department*
Sandia National Laboratories[¶], Albuquerque, NM 87185

Surrogate-based-optimization methods provide a means to minimize expensive high-fidelity models at reduced computational cost, by using a high-fidelity model in combination with a low-fidelity model that is less costly but less accurate. A number of model management techniques have been developed and shown to work well for the case in which both models are defined over the same design space. However, many systems exist with variable fidelity models for which the design variables are defined over different spaces, and a mapping is required between the spaces. Two mapping methods, corrected space mapping and POD mapping, are used in conjunction with trust-region model management. Four constraint-management methods are demonstrated with each of the mapping methods: Lagrangian minimization, direct surrogate optimization, a method based on sequential quadratic programming (SQP), and a composite step approach called MAESTRO. The methods are demonstrated on a fixed-complexity analytical test problem and a variable-complexity wing design problem. Both direct surrogate optimization and the SQP-like method consistently outperformed optimization in the high-fidelity space. On the wing design problem, the combination of direct surrogate optimization and POD mapping achieved 53% savings in high-fidelity function calls over optimization directly in the high-fidelity space.

I. Introduction

As computational capabilities continue to grow, designers of engineering systems have available an increasing range of numerical analysis models. These multifidelity models range from low-fidelity simple-physics models to high-fidelity detailed computational simulation models. The drive towards including higher-fidelity analyses in the design process, for example through the use of computational fluid dynamic (CFD) analyses, leads to an increase in complexity and computational expense. As a result, design optimization, which requires large numbers of analyses of objectives and constraints, becomes prohibitively expensive for many systems of interest. This paper presents a methodology for improving the computational efficiency of high-fidelity design, by exploiting variable fidelity and variable complexity — that is, inexpensive models of lower physical resolution combined with coarser design parameterizations — in a design optimization framework.

Surrogate-based optimization (SBO) methods have been proposed as one method to achieve high-fidelity design optimization at reduced computational cost. In SBO, a surrogate, or less expensive and lower-fidelity,

[*]Graduate Student, Student Member AIAA, robinst@mit.edu
[†]Associate Professor of Aeronautics and Astronautics, Senior Member AIAA
[‡]Principal Member of Technical Staff, Associate Fellow AIAA
[§]Principal Research Engineer, Member AIAA
[¶]Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed-Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

model is used for the majority of the optimization, with recourse to the high-fidelity analysis less frequently. The determination of when to use the high-fidelity model is based on some rigorous procedure for deciding the appropriate level of fidelity. The surrogate model can be developed in a number of ways, including using a simplified-physics model with a different set of governing equations. However, an improvement in a design predicted by a low-fidelity model does not guarantee an improvement in the high-fidelity problem.

Much work has been performed on developing SBO methods that are provably convergent to an optimum of the high-fidelity problem. Ref. 1 reviews a broad spectrum of SBO work. One promising group of methods is based on trust-region model management (TRMM), which imposes limits on the amount of optimization performed using the low-fidelity model, based on a quantitative assessment of that model's predictive capability. These TRMM methods are provably convergent to an optimum of the high-fidelity model, provided the low-fidelity model is corrected to be at least first-order consistent with the high-fidelity model.[2] Correcting to second-order or quasi-second-order consistency provides improved performance.[3] The corrections can be additive or multiplicative; additive has been shown to be more appropriate in a wider variety of problems.[3]

A number of researchers have developed SBO methods for constrained problems. Booker et al. developed a direct-search SBO framework that converges to a minimizer of an expensive objective function subject only to bounds on the design variables, and that does not require derivative evaluations.[4] Audet et al. extended that framework to handle general nonlinear constraints[5] using a filter method for step acceptance.[6] Rodriguez et al. developed a gradient-based TRMM augmented-Lagrangian strategy using response surfaces, and showed that using separate response surfaces for the objective and constraints provided faster convergence than using a single response surface for the augmented Lagrangian.[7] Alexandrov et al. developed the MAESTRO class of methods, which use gradient based optimization and trust region model management, and compared them to a sequential quadratic programming (SQP)-like TRMM method.[8]

The SBO methods developed to date achieve computational gain by performing most of the analysis on the low-fidelity model; however, they require that the high- and low-fidelity models operate with the same set of design variables. For practical design applications, multifidelity models are often defined over different design spaces. For example, the multifidelity supersonic business jet problem considered by Choi et al.[9] has a low-fidelity model that uses classical supersonic aerodynamics and vortex lattice-methods, and a high-fidelity analysis that employs the Euler equations. The low-fidelity model operates on an aircraft defined by 16 design variables: the wing area, aspect ratio, and sweep, the location of the wing root leading edge, the thickness to chord length at three locations on the wing, the minimum cockpit diameter, the minimum cabin diameter, and the fuselage radii at six locations. The high-fidelity model uses 126 design variables: leading and trailing edge droop, twist, and 15 camber Hicks-Henne bumps at each of 7 locations on the wing. Further, combining a low-fidelity model with a coarser parameterization of the design offers the opportunity for additional reduction in computational complexity and cost beyond current SBO methods. To achieve this, new variable-complexity design methodology is required that incorporates variable design parameterizations into SBO methods.

Methods for incorporating models defined over variable design spaces for unconstrained optimization were introduced in Ref. 10. This work extends the work in Ref. 10 to address constrained problems, which is of interest for the majority of practical design problems. Since many algorithms for solving constrained problems are reduced to solving a sequence of unconstrained subproblems, solution methods for unconstrained problems provide a foundation for development of constrained problems. In Section II, an overview of constrained trust-region methods is presented. In Section III, two methods for mapping between different design spaces are presented, and a description of the trust-region approach in the variable-complexity case is given. Section IV and Section V present results for an analytical test case and an engineering design problem, respectively. Finally, conclusions are drawn in Section VI.

II. Constrained Trust-region Methods

We consider a general design problem posed using the following nonlinear optimization formulation:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{c}(\mathbf{x}) \leq 0 \\ & \mathbf{x}_L \leq \mathbf{x} \leq \mathbf{x}_U, \end{aligned} \tag{1}$$

where $f \in \mathbb{R}$ represents the scalar objective to be minimized and $\mathbf{x} \in \mathbb{R}^n$ is the vector of n design variables that describe the design. The vectors $\mathbf{x}_L$ and $\mathbf{x}_U$ define lower and upper bounds on $\mathbf{x}$, respectively. The vector $\mathbf{c} \in \mathbb{R}^m$ contains m constraints, which provide a mathematical description of requirements that the design must satisfy. For realistic design problems of engineering relevance, the complexity of the optimization problem (1) is twofold: first, the simulations required to evaluate $f(\cdot)$ and $\mathbf{c}(\cdot)$ may be computationally expensive, and second, the dimensionality of $\mathbf{x}$ may be large.

One approach to reduce the cost of optimization is to use SBO. Past work has focused on providing surrogate models $\tilde{f}(\mathbf{x})$ and $\tilde{\mathbf{c}}(\mathbf{x})$, for $f(\mathbf{x})$ and $\mathbf{c}(\mathbf{x})$ respectively, that are computationally more efficient to evaluate.[11] Surrogate models can be roughly divided into three categories: data fit surrogates, using interpolation or regression of the high-fidelity model evaluated at one or more sample points;[12–14] reduced-order models, derived using techniques such as proper orthogonal decomposition (POD)[15,16] and modal analysis;[17,18] and hierarchical models, also called multifidelity, variable-fidelity, or variable-complexity models. In this last case, a physics-based model of lower fidelity and reduced computational cost is used as the surrogate in place of the high-fidelity model. The multifidelity case can be further divided based on the means by which the fidelity is reduced in the lower-fidelity model. The low-fidelity surrogate can be the same as the high-fidelity, but converged to a higher residual tolerance;[19] it can be the same model on a coarser grid;[20,21] or it can use a simpler engineering model that neglects some physics modeled by the high-fidelity analysis.[22]

Surrogate models can be rigorously incorporated into design optimization through the use of a formal model management strategy. One such strategy is a TRMM framework.[23] By using corrections to ensure that the surrogate model is at least first-order accurate at the center of the trust region, this method is provably convergent to a local minimum of the high-fidelity function. The TRMM framework is widely used, having been adapted for multi-objective optimization[24] and multidisciplinary optimization,[25] in cases when the design vector is the same between the high-fidelity and surrogate models.

The general approach in TRMM is to solve an optimization subproblem using only the low-fidelity model, with an additional constraint that requires the solution to lie within a specified trust region. The size of the trust region is managed using a merit function to quantitatively assess the predictive capability of the low-fidelity model. There are a number of ways to incorporate constraints in trust-region methods; the specific subproblem solved on each iteration and the merit function depend on the specific constraint management method chosen. The four constraint management methods described in the following subsections are augmented Lagrangian minimization, direct surrogate optimization, an SQP-like method, and MAESTRO.

In this work, we denote the uncorrected low-fidelity models for f and $\mathbf{c}$ by $\hat{f}$ and $\hat{\mathbf{c}}$, respectively, while the respective corrected surrogate models are denoted $\tilde{f}$ and $\tilde{\mathbf{c}}$.

A. Lagrangian minimization

One method to solve constrained optimization problems is to formulate a Lagrangian or augmented Lagrangian of the problem and solve the resulting unconstrained problem. The first constraint management method combines this approach with the unconstrained TRMM method of Alexandrov et al.[23] and quasi-second-order additive corrections using the Broyden-Fletcher-Goldfarb-Shanno[26–29] (BFGS) approximation to the Hessian matrix.[3] The algorithm is modified from Ref. 30, with the construction of the response surface in that work replaced by the correction of the low-fidelity model to be consistent with the high-fidelity model. A series of subproblems are solved, which have the form

$$\begin{aligned} \min_{\mathbf{x}} \quad & \tilde{L}^k(\mathbf{x}, \lambda^k, r_p^k) \\ \text{subject to} \quad & \|\mathbf{x} - \mathbf{x}_c^k\|_\infty \leq \Delta^k, \end{aligned} \tag{2}$$

where, for subproblem k, $\tilde{L}^k(\mathbf{x}, \lambda^k, r_p^k)$ is the surrogate augmented Lagrangian, λ^k is the vector of Lagrange multipliers, r_p^k is the penalty parameter, $\mathbf{x}_c^k$ is the center of the trust region, and Δ^k is the trust-region radius. The solution of the k^{th} subproblem (2) is denoted $\mathbf{x}_*^k$ and, if the step is accepted, is used as the center of the trust region for the following subproblem, that is $\mathbf{x}_c^k = \mathbf{x}_*^{k-1}$. The surrogate augmented Lagrangian, $\tilde{L}$, can be defined in two different ways, which are described below.

The size of the trust region, Δ^k, is determined using a merit function. The trust-region ratio is defined as the ratio of the actual improvement in the merit function to the improvement predicted by optimization on the surrogate model of the merit function. For this method, the merit function is the augmented Lagrangian,

and the trust-region ratio is therefore defined as

$$\rho^k = \frac{L(\mathbf{x}_c^k, \lambda^k, r_p^k) - L(\mathbf{x}_*^k, \lambda^k, r_p^k)}{\tilde{L}^k(\mathbf{x}_c^k, \lambda^k, r_p^k) - \tilde{L}^k(\mathbf{x}_*^k, \lambda^k, r_p^k)}, \tag{3}$$

where the high-fidelity augmented Lagrangian is defined by

$$L(\mathbf{x}, \lambda, r_p) = f(\mathbf{x}) + \lambda^T \mathbf{c}(\mathbf{x}) + r_p \sum_{i=1}^{m} \Psi_i(\mathbf{x}, \lambda, r_p)^2, \tag{4}$$

where

$$\Psi_i(\mathbf{x}, \lambda, r_p) = \max\left(c_i(\mathbf{x}), -\frac{\lambda_i}{2r_p}\right), \quad i = 1, \ldots, m. \tag{5}$$

The value for ρ^k defines the acceptance of the k^{th} subproblem solution and the next trust region size, Δ^{k+1}, using the following logic:

1. $\rho^k \leq 0$: The surrogate models are inaccurate. Reject the step and shrink the trust region by half to improve surrogate accuracy.

2. $0 < \rho^k \leq r_1$: The surrogate models are marginally accurate. Accept the step but shrink the trust region size by half.

3. $r_1 < \rho^k < r_2$: The surrogate models are moderately accurate. Accept the step and maintain the current trust region size.

4. $r_2 \leq \rho^k$: The surrogate models are accurate. Accept the step and increase the trust region size by a factor of two.

The parameters r_1 and r_2 are user-selected, and in this work are $r_1 = 0.2$ and $r_2 = 0.9$.

The Lagrange multipliers λ^k in (4) are updated when $\|\nabla L(\mathbf{x}_c^k, \lambda^k, r_p^k)\|_2 < \mu^k$ or there have been three consecutive subproblem step rejections, and $\|c(\mathbf{x}_c^k)\|_\infty \leq \eta^k$, using for the i^{th} Lagrange multiplier the formula

$$\lambda_i^{k+1} = \max\left(0, \lambda_i^k + 2r_p^k c_i(\mathbf{x}_c^k)\right), \tag{6}$$

and the penalty parameter r_p^k is updated when $\|\nabla L(\mathbf{x}_c^k, \lambda^k, r_p^k)\|_2 < \mu^k$ and $\|c(\mathbf{x}_c^k)\|_\infty > \eta^k$ using

$$r_p{}^{k+1} = 5r_p^k. \tag{7}$$

The trust-region size is an indicator of how well the surrogate function is approximating the high-fidelity augmented Lagrangian. Whenever either the Lagrange multipliers or penalty parameter are updated, the augmented Lagrangian changes significantly, so the size of the trust region is reset using

$$\Delta^{k+1} = \kappa \|\nabla L(\mathbf{x}_c^k, \lambda^k, r_p^k)\|_2, \tag{8}$$

where κ is some number greater than one. Since the norm of the gradient of the augmented Lagrangian is a measure of proximity to the optimum solution, (8) ensures that the trust region size will become smaller as the current iterate approaches the optimum. Also, when the Lagrange multipliers and penalty parameter are changed, the parameters μ and η are updated using

$$\mu^{k+1} = \mu^k \min\left(\frac{1}{r_p^k}, 0.1\right) \tag{9}$$

and

$$\eta^{k+1} = \eta^k \min\left(\frac{1}{r_p^k}, \gamma\right). \tag{10}$$

The algorithm parameters μ^0, η^0 and γ are user-selected. The Lagrange multiplier and penalty parameter update rules are illustrated in Figure 1 of Ref. 30.

The surrogate augmented Lagrangian, $\tilde{L}$ is defined in one of the two following ways. These methods are variations on the two methods compared in Ref. 7.

1. Approximating the Lagrangian

In this method, the whole augmented Lagrangian is approximated with a single surrogate. That is, on each iteration, a surrogate $\tilde{L}^k$ is created for L, using a BFGS quasi-second-order additive correction to the augmented Lagrangian of the low-fidelity model, $\hat{L}$, defined by

$$\hat{L}^k(\mathbf{x}, \lambda, r_p) = \hat{f}(\mathbf{x}) + \lambda^T \hat{\mathbf{c}}(\mathbf{x}) + r_p \sum_{i=1}^{m} \hat{\Psi}_i(\mathbf{x}, \lambda, r_p)^2, \tag{11}$$

using

$$\hat{\Psi}_i(\mathbf{x}, \lambda, r_p) = \max\left(\hat{c}_i(\mathbf{x}), -\frac{\lambda_i}{2r_p}\right), \quad i = 1, \ldots, m. \tag{12}$$

The surrogate augmented Lagrangian for each subproblem is then obtained by correcting the low-fidelity augmented Lagrangian,

$$\tilde{L}^k(\mathbf{x}, \lambda, r_p) = \hat{L}(\mathbf{x}, \lambda, r_p) + A^k(\mathbf{x}), \tag{13}$$

where A^k is a quadratic function of $\mathbf{x}$:

$$A^k = A_0^k + \nabla {A^k}^T (\mathbf{x} - \mathbf{x}_c) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_c)^T \nabla^2 A^k (\mathbf{x} - \mathbf{x}_c). \tag{14}$$

Each of the components of A^k is calculated such that at $\mathbf{x}_c$, L and $\tilde{L}^k$ have the same function value, gradient, and BFGS approximation to the Hessian matrix.

2. Approximating objective and constraints separately

In this version of the method, the surrogate augmented Lagrangian is formed using separate approximations to the objectives and the constraints. The surrogate augmented Lagrangian is therefore

$$\tilde{L}^k(\mathbf{x}, \lambda, r_p) = \tilde{f}^k(\mathbf{x}) + \lambda^T \tilde{\mathbf{c}}^k(\mathbf{x}) + r_p \sum_{i=1}^{m} \tilde{\Psi}_i^k(\mathbf{x}, \lambda, r_p)^2, \tag{15}$$

where $\tilde{\Psi}^k$ is defined as in (12) but using the surrogate rather than low-fidelity constraints, and $\tilde{f}^k$ and $\tilde{c}^k$ are obtained using additive corrections,

$$\tilde{f}^k(\mathbf{x}) = \hat{f}(\mathbf{x}) + A^k(\mathbf{x}), \tag{16}$$

and

$$\tilde{c}_i^k(\mathbf{x}) = \hat{c}_i(\mathbf{x}) + B_i^k(\mathbf{x}),\ i = 1 \ldots m. \tag{17}$$

The correction function $A^k(\mathbf{x})$ is defined in equation (14). Each correction function $B_i^k(\mathbf{x})$ is defined similarly, ensuring quasi-second-order consistency between the surrogate constraints and the high-fidelity constraints at the center of the trust region.

B. Direct surrogate optimization

The second method of constraint management solves a series of subproblems that minimize the surrogate objective subject to the surrogate constraints. The subproblem in this case is defined as

$$\begin{aligned} \min_{\mathbf{x}} \quad & \tilde{f}^k(\mathbf{x}) \\ \text{subject to} \quad & \tilde{\mathbf{c}}^k(\mathbf{x}) \leq 0 \\ & \|\mathbf{x} - \mathbf{x}_c^k\|_\infty \leq \Delta^k. \end{aligned} \tag{18}$$

The merit function for this method is the augmented Lagrangian, calculated using the separate approximations to the objectives and the constraints as in Section IIA2. The trust-region ratio is therefore defined by equation (3), and the size of the trust region is managed using the logic given in Section IIA. The Lagrange multipliers and penalty parameters are also updated using equations (6) and (7) in order to calculate the merit function.

C. Sequential-Quadratic-Programming-like method

This method is modified from Ref. 8. It is similar to sequential quadratic programming (SQP) in that on each subproblem it minimizes an approximation to the Lagrangian subject to linear approximations to the high-fidelity constraints. The subproblem is

$$\begin{aligned} \min_{\mathbf{x}} \quad & \tilde{\mathcal{L}}^k(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{c}(\mathbf{x}_c^k) + \nabla \mathbf{c}(\mathbf{x}_c^k)^T(\mathbf{x} - \mathbf{x}_c^k) \leq 0 \\ & \|\mathbf{x} - \mathbf{x}_c^k\|_\infty \leq \Delta^k, \end{aligned} \tag{19}$$

where $\tilde{\mathcal{L}}^k$ is the approximation to the Lagrangian of the problem, defined as

$$\tilde{\mathcal{L}}^k(\mathbf{x}, \lambda^k) = \tilde{f}^k(\mathbf{x}) + {\lambda^k}^T \tilde{\mathbf{c}}^k(\mathbf{x}). \tag{20}$$

The merit function for this method is the Lagrangian. Since this method uses the Lagrangian rather than the augmented Lagrangian, there is no penalty parameter and equation (6) cannot be used. The Lagrange multipliers are therefore updated by minimizing the residual in the Karush-Kuhn-Tucker conditions. The vector of Lagrange multipliers for the next subproblem, λ^{k+1}, is determining by solving

$$\begin{aligned} \lambda^{k+1} = \arg\min_{\lambda} \quad & \|\nabla f(\mathbf{x}_c^k) + \textstyle\sum_{i \in S^k} \lambda_i \nabla c_i(\mathbf{x}_c^k)\|_2^2 \\ \text{subject to} \quad & \lambda \geq 0, \end{aligned} \tag{21}$$

where S^k is the set of active constraints for subproblem k. The optimization problem (21) is a nonnegative least-squares constraint problem, which is solved using the method of Ref. 31.

D. MAESTRO

The MAESTRO method was developed by Alexandrov in order to solve multidisciplinary design optimization (MDO) problems that include constraints from a number of disjoint disciplines, and that are possibly calculated on separate nodes of a distributed system. It is described in detail in Ref. 8. The major difference between MAESTRO and the other trust-region methods is that MAESTRO solves two problems on each iteration: one to minimize the constraints and another to minimize the objective. The step used is then the sum of the steps found from the solutions to those two problems. Separate trust regions are maintained for the objectives and for the constraints.

The MAESTRO method requires only equality constraints; therefore the inequality constraints are converted to equality constraints using squared slack variables. The i^{th} inequality constraint, c_i, is converted to an inequality constraint, h_i, using

$$h_i(\mathbf{x}) = c_i(\mathbf{x}) + z_i^2 = 0, \tag{22}$$

where z_i is the corresponding slack variable. The vector of resulting equality constraints is denoted $\mathbf{h}(\mathbf{x}, \mathbf{z})$, where $\mathbf{z}$ is the vector containing the slack variables.

Each subproblem is broken down into two parts. The first part minimizes the constraints, subject to the bounds on the solution provided by the trust region:

$$\begin{aligned} \min_{\mathbf{x},\mathbf{z}} \quad & \|\tilde{\mathbf{h}}^k(\mathbf{x}, \mathbf{z})\|_2 \\ \text{subject to} \quad & \|\mathbf{x} - \mathbf{x}_c^k\|_\infty \leq \Delta_{con}^k, \\ & \|\mathbf{z} - \mathbf{z}_c^k\|_\infty \leq \Delta_{con}^k, \end{aligned} \tag{23}$$

where $\tilde{\mathbf{h}}^k$ is the surrogate model of $\mathbf{h}$ for the k^{th} subproblem, $\mathbf{z}_c^k$ is the vector of slack variable values at the center of the trust region, and Δ_{con}^k is the trust-region size for the k^{th} constraint subproblem. The solution to this problem is denoted $(\mathbf{x}_1^k, \mathbf{z}_1^k)$. The constraint trust-region ratio is calculated as:

$$\rho_{con}^k = \frac{\|\mathbf{h}(\mathbf{x}_c^k, \mathbf{z}_c^k)\|_2 - \|\mathbf{h}(\mathbf{x}_1^k, \mathbf{z}_1^k)\|_2}{\|\tilde{\mathbf{h}}^k(\mathbf{x}_c^k, \mathbf{z}_c^k)\|_2 - \|\tilde{\mathbf{h}}^k(\mathbf{x}_1^k, \mathbf{z}_1^k)\|_2}, \tag{24}$$

and the constraint trust region radius Δ_{con}^k is updated according to the rules in Section IIA using the two-norm of the constraints as the merit function.

The second part of the subproblem minimizes the objective, subject to the conditions that the solution remains within the objective trust region and that improvement in the constraints is maintained:

$$\begin{aligned} \min_{\mathbf{x},\mathbf{z}} \quad & \tilde{\mathbf{f}}^k(\mathbf{x},\mathbf{z}) \\ \text{subject to} \quad & \|\mathbf{x}-\mathbf{x}_1^k\|_\infty \leq \Delta_{obj}^k, \\ & \|\mathbf{z}-\mathbf{z}_1^k\|_\infty \leq \Delta_{obj}^k, \\ & \tilde{\mathbf{h}}^k(\mathbf{x},\mathbf{z}) = \tilde{\mathbf{h}}^k(\mathbf{x}_1^k,\mathbf{z}_1^k). \end{aligned} \tag{25}$$

The solution to the problem (25) is denoted $(\mathbf{x}_2^k, \mathbf{z}_2^k)$. The objective trust-region ratio is calculated as

$$\rho_{obj}^k = \frac{f(\mathbf{x}_1^k,\mathbf{z}_1^k) - f(\mathbf{x}_2^k,\mathbf{z}_2^k)}{\tilde{f}(\mathbf{x}_1^k,\mathbf{z}_1^k) - \tilde{f}^k(\mathbf{x}_2^k,\mathbf{z}_2^k)}, \tag{26}$$

and the objective trust region size Δ_{obj}^k is updated according to the rules in Section IIA.

III. Variable-complexity Design Spaces

Multifidelity surrogate models provide the opportunity to further reduce the complexity of the optimization problem (1) by using a different design vector. That is, a lower-fidelity model of $f(\mathbf{x})$ may be given by $\hat{f}(\hat{\mathbf{x}})$, where $\hat{\mathbf{x}} \in \mathbb{R}^{\hat{n}}$ is a design vector of reduced complexity. For example, the dimension of $\hat{\mathbf{x}}$, $\hat{n}$, may be smaller than the dimension of $\mathbf{x}$. In the case of variable grid size, the optimization design variables may be directly related to the discretization of the grid, leading to fewer design variables on the coarser grid. In the neglected-physics case, the two models may use entirely different sets of design variables.

Constrained SBO methods have until now been applicable only to models in which both the high-fidelity model, $f(\mathbf{x})$, $\mathbf{c}(\mathbf{x})$, and the low-fidelity model, $\hat{f}(\hat{\mathbf{x}})$, $\hat{\mathbf{c}}(\hat{\mathbf{x}})$, are defined over the same space, that is, $\mathbf{x} = \hat{\mathbf{x}}$. In order to use a low-fidelity model with a different set of design variables from the high-fidelity function to be optimized, it is necessary to find a relationship between the two sets of design vectors, that is, $\hat{\mathbf{x}} = P(\mathbf{x})$ or $\mathbf{x} = Q(\hat{\mathbf{x}})$. In Ref. 10, such a mapping was applied to unconstrained problems. This work extends these mapping techniques to constrained problems. In this section, we first describe the variable-complexity case of the constrained trust-region methods and then present two mapping methods.

A. Variable-complexity trust region methods

To extend the TRMM methods described in Section II to the case of variable complexity, the mapping between $\mathbf{x}$ and $\hat{\mathbf{x}}$ must be incorporated. The mapping must be introduced in such a way as to ensure at least first-order consistency between the surrogate model and high-fidelity model at the center of the trust region. In order to enforce this consistency, additive corrections are used.

For some low-fidelity function $\hat{\beta}(\hat{\mathbf{x}})$ that approximates a high-fidelity function $\beta(\mathbf{x})$ and a mapping $\hat{\mathbf{x}} = P(\mathbf{x})$, the k^{th} corrected surrogate function is defined as

$$\tilde{\beta}^k(\mathbf{x}) = \hat{\beta}\left(P(\mathbf{x})\right) + A^k(\mathbf{x}). \tag{27}$$

The correction function, A^k, is a quadratic Taylor series of the difference between the two functions β and $\hat{\beta}$

$$A^k = A_0^k + \nabla A^{k^T}(\mathbf{x}-\mathbf{x}_c) + \frac{1}{2}(\mathbf{x}-\mathbf{x}_c)^T\nabla^2 A^k(\mathbf{x}-\mathbf{x}_c), \tag{28}$$

with each element calculated using

$$A_0^k = \beta^k(\mathbf{x}_c^k) - \hat{\beta}\left(P(\mathbf{x}_c^k)\right), \tag{29}$$

$$\frac{\partial A^k}{\partial x_p} = \frac{\partial \beta}{\partial x_p}(\mathbf{x}_c^k) - \sum_{j=1}^{\hat{n}} \frac{\partial \hat{\beta}}{\partial \hat{x}_j}\left(P(\mathbf{x}_c^k)\right)\frac{\partial \hat{x}_j}{\partial x_p}, \quad p = 1\ldots n, \tag{30}$$

$$\frac{\partial^2 A^k}{\partial x_p \partial x_q} = H_{pq} - \sum_{j=1}^{\hat{n}}\left(\frac{\partial \hat{\beta}}{\partial \hat{x}_j}\left(P(\mathbf{x}_c^k)\right)\frac{\partial^2 \hat{x}_j}{\partial x_p \partial x_q} + \sum_{\ell=1}^{\hat{n}} \hat{H}_{j\ell}\frac{\partial \hat{x}_j}{\partial x_p}\frac{\partial \hat{x}_\ell}{\partial x_q}\right), \quad p = 1\ldots n, \quad q = 1\ldots n, \tag{31}$$

where $\mathbf{H}$ is the BFGS approximation to the Hessian matrix of the high-fidelity function β at $\mathbf{x}_c^k$, $\hat{\mathbf{H}}$ is the BFGS approximation to the Hessian matrix of the low-fidelity function $\hat{\beta}$ at $P(\mathbf{x}_c^k)$, and H_{pq} denotes the pq^{th} element of the matrix $\mathbf{H}$.

When using the Lagrangian-approximating approach of Section IIA1, the correction uses the augmented Lagrangian for β and $\tilde{\beta}$. When using the other four multifidelity constraint-management approaches, corrections are applied separately to the objective, setting $\beta = f$ and $\tilde{\beta} = \tilde{f}$, and to each of the constraints, setting $\beta = c_i$ and $\tilde{\beta} = \tilde{c}_i$. For each subproblem k, equation (29) computes the difference between the value of the high-fidelity function and the low-fidelity function at the center of the trust region. Using the chain rule, equation (30) computes the difference between the gradient of the high-fidelity function and the gradient of the low-fidelity function at the center of the trust region, where the gradients are computed with respect to the high-fidelity design vector $\mathbf{x}$. The second term in (30) therefore requires the Jacobian of the mapping, $\frac{\partial \hat{x}_j}{\partial x_p}$. Similarly, equation (31) computes the difference between the BFGS approximation of the Hessian matrices of the high-fidelity and low-fidelity functions at the center of the trust region. Once again, derivatives are required with respect to $\mathbf{x}$ and are computed using the chain rule.

The correction function A used is defined as a function of $\mathbf{x}$, that is, the set of high-fidelity design variables. In this implementation, the steps taken by the optimization method are in the high-fidelity space. One avenue of current research is to develop mappings from the high-fidelity space to the low-fidelity space that are constrained to contain the step directions of interest, thus allowing the optimization method to work in the low-fidelity space.

It is also interesting to note that, in the case of a low-fidelity design vector $\hat{\mathbf{x}}$ of lower dimension than the high-fidelity design vector $\mathbf{x}$, the mapping identifies a subspace of the high-fidelity space to be spanned by the low-fidelity design vectors. If only the mapped low-fidelity objective and constraints were used, optimization progress would be limited to that subspace. That is, no change would be observed in those high-fidelity design variables (or combinations thereof) not represented in the low-fidelity space. However, since the correction function (28) is defined in the high-fidelity space in such a way that the surrogate models are BFGS-consistent with the high-fidelity functions in all the high-fidelity dimensions, the solutions to the trust-region optimization subproblems can be in any direction in the high-fidelity space.

B. Corrected space mapping

Space mapping[32, 33] is a method of linking variable-fidelity models developed in the microwave circuit design community. In that application area, it is often appropriate to make corrections to the input of a model, rather than to its output. In space mapping, a particular form is assumed for the relationship P between the high- and low-fidelity design vectors. This form is described by some set of parameters, contained here in a vector $\mathbf{p}$, that are found by minimizing the difference between the high- and low-fidelity outputs. However, this standard space mapping method does not enforce the consistency required for a convergent trust-region optimization algorithm.

In order to obtain a surrogate model that satisfies the consistency conditions for provable convergence, we use the corrected space mapping method presented in Ref. 10. This method performs the space mapping and correction steps simultaneously, by embedding the correction within the space mapping optimization problem. That is, it incorporates a correction, and with the remaining degrees of freedom, performs the best match possible to the control points by varying the input mapping.

In the unconstrained case, the corrected space mapping optimization problem is given by

$$\mathbf{p}^k = \arg\min_{\mathbf{p}} \sum_{i=1}^{q} ||\beta(\mathbf{x}^i) - \tilde{\beta}^k(\mathbf{x}^i, \mathbf{p})||_2, \tag{32}$$

where β is chosen to be the high-fidelity objective function f and $\tilde{\beta}^k$ is the corresponding corrected surrogate function $\tilde{f}^k$. The optimization problem (32) seeks to minimize the difference between the high-fidelity and surrogate objective functions over a set of q sample points $\mathbf{x}^i$, where $\mathbf{x}^i$ denotes the i^{th} sample (or control) point. Both the choice of sample points and the particular form of the mapping P is left to the implementation.

Here, we consider the extension of corrected space mapping to constrained problems. When using the Lagrangian-approximating approach of Section IIA1, the corrected space mapping optimization problem uses the augmented Lagrangian for β and $\tilde{\beta}$. That is, the space mapping parameters $\mathbf{p}$ are determined by solving

an optimization problem of the form (32) that minimizes the difference over a set of sample points between the high-fidelity Lagrangian L, as defined in equation (4), and the surrogate Lagrangian $\tilde{L}$, as defined in equation (13). When using the other four multifidelity constraint-management approaches, space mapping is applied separately to the objective, setting $\beta = f$ and $\tilde{\beta} = \tilde{f}$, and to each of the constraints, setting $\beta = c_i$ and $\tilde{\beta} = \tilde{c}_i$. This results in $m+1$ space mapping optimization problems of the form (32), each with its own correction function and vector of space mapping parameters. The elements of the correction function are calculated using equations (29) to (31).

In the implementation employed in this work, the sample points used in equation (32) are the previous q subproblem iterates, at which high-fidelity objective function and constraint values are already available, and a linear relationship was chosen for the mapping P:

$$\hat{\mathbf{x}} = P(\mathbf{x}) = \mathbf{M}\mathbf{x} + \mathbf{b}, \tag{33}$$

where $\mathbf{M}$ is a matrix with $\hat{n} \times n$ elements and $\mathbf{b}$ is a vector of length $\hat{n}$ for a total of $(n+1) \times \hat{n}$ space-mapping parameters contained in $\mathbf{p}$.

It should be noted that when using corrected space mapping, the correction function must be re-evaluated with each new value of $\mathbf{p}$, since the low-fidelity function values, the low-fidelity gradient, and the first- and second-order derivatives of the mapping needed in equations (29) – (31) change with the space-mapping parameters, through their dependence on $\hat{x}$. Since the resulting corrected function is at least first-order accurate at the center of the trust region, the resulting trust-region optimization is provably convergent to a local minimum of the high-fidelity problem.

C. POD mapping

The second mapping methodology is based on the gappy proper orthogonal decomposition (POD) method of reconstructing data sets. This, in turn, is based on the POD method, also known as principal components analysis and Karhunen-Loève expansions, which yields a set of basis vectors that provides the least-squares optimal representation of a given data set.

The POD method of snapshots, developed by Sirovich,[16] finds the basis vectors empirically. In this method, a set of q snapshots $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^q$, or column vectors describing different states of a system, is computed. The POD basis vectors, $\phi^j,\ j = 1, 2, \ldots, q$, can then be computed as the left singular vectors of the matrix $\mathbf{X}$, defined as

$$\mathbf{X} = \begin{bmatrix} \left[\mathbf{x}^1 - \bar{\mathbf{x}}\right] & \left[\mathbf{x}^2 - \bar{\mathbf{x}}\right] & \cdots & \left[\mathbf{x}^q - \bar{\mathbf{x}}\right] \end{bmatrix}, \tag{34}$$

where $\bar{\mathbf{x}}$ is the mean of the snapshots and the i^{th} column of X contains the i^{th} snapshot minus the mean.

The singular values of $\mathbf{X}$ indicate the relative importance of the corresponding basis vector in representing the snapshots. Therefore, only the q basis vectors corresponding to the largest singular values are used. A low-dimensional representation of a solution $\mathbf{x}$ is thus given by

$$\mathbf{x} \approx \bar{\mathbf{x}} + \sum_{j=1}^{r} \nu_i \phi^i, \tag{35}$$

where ν_i is the coefficient describing the contribution of the i^{th} POD mode ϕ^i to the solution $\mathbf{x}$.

The gappy POD method allows one to reconstruct data from a "gappy" data set, that is, a set in which some of the data are unknown or missing.[34,35] The first step is to define a mask vector, which describes for a particular solution vector where data are available and where data are missing. For the solution $\mathbf{x}$, the corresponding mask vector $\mathbf{n}$ is defined as follows:

$$\begin{aligned} n_i &= 0 \text{ if } x_i \text{ is missing} \\ n_i &= 1 \text{ if } x_i \text{ is known,} \end{aligned}$$

where x_i denotes the i^{th} element of the vector $\mathbf{x}$. Pointwise multiplication is defined as $(\mathbf{n}, \mathbf{x})_i = n_i x_i$. Then the gappy inner product is defined as $(\mathbf{u}, \mathbf{v})_n = ((\mathbf{n}, \mathbf{u}), (\mathbf{n}, \mathbf{v}))$, and the induced norm is $(\|\mathbf{v}\|_n)^2 = (\mathbf{v}, \mathbf{v})_n$.

For a vector $\mathbf{x}$ that has some unknown components, it is assumed that the repaired vector $\check{\mathbf{x}}$ can be represented by the expansion (35). In this representation, the POD coefficients ν_i are chosen to minimize the error between the available and reconstructed data. This error can be defined as

$$\epsilon = \|\mathbf{x} - \check{\mathbf{x}}\|_n^2 \tag{36}$$

using the gappy norm so that only the original existing data elements in $\mathbf{u}$ are compared. The coefficients ν_i that minimize the error ϵ can be found by differentiating (36) with respect to each of the ν_i in turn. This leads to the linear system of equations

$$\mathbf{E}\nu = \mathbf{r}, \tag{37}$$

where the ij^{th} component of $\mathbf{E}$ is given by

$$E_{ij} = \left(\phi^i, \phi^j\right)_n \tag{38}$$

and the i^{th} component of $\mathbf{r}$ is given by

$$r_i = \left(\mathbf{u}, \phi^i\right)_n. \tag{39}$$

Solving equation (37) for ν, the missing elements of $\mathbf{u}$ can be obtained using the expansion (35).

As described in Ref. 10, the gappy POD method provides a way to map between high- and low-fidelity design space data: the high-fidelity vector is treated as the known data, and the low-fidelity as the unknown data, or vice versa. In the mapping application, the POD basis vectors must span both low- and high-fidelity design space. This is achieved by generating a set of r training pairs, for which the low- and the high-fidelity vectors describe the same physical system. These training pairs are combined in the following way to form the snapshot matrix:

$$\mathbf{X} = \begin{bmatrix} \left[\hat{\mathbf{x}}_1 - \bar{\hat{\mathbf{x}}}\right] & \left[\hat{\mathbf{x}}_2 - \bar{\hat{\mathbf{x}}}\right] & \cdots & \left[\hat{\mathbf{x}}_r - \bar{\hat{\mathbf{x}}}\right] \\ - & - & & - \\ \left[\mathbf{x}_1 - \bar{\mathbf{x}}\right] & \left[\mathbf{x}_2 - \bar{\mathbf{x}}\right] & \cdots & \left[\mathbf{x}_r - \bar{\mathbf{x}}\right] \end{bmatrix}, \tag{40}$$

where now the i^{th} column of $\mathbf{X}$ contains both the i^{th} low- and the i^{th} high-fidelity snapshots, and $\bar{\hat{\mathbf{x}}}$ denotes the mean of the low-fidelity snapshot set.

The left singular vectors of this snapshot matrix provide the corresponding POD basis vectors, which are partitioned in the same way as the snapshot vectors. Therefore, equation (35) can be decomposed into two equations

$$\mathbf{x} = \bar{\mathbf{x}} + \sum_{i=1}^{q} \nu_i \phi^i \tag{41}$$

$$\hat{\mathbf{x}} = \bar{\hat{\mathbf{x}}} + \sum_{i=1}^{q} \nu_i \hat{\phi}^i, \tag{42}$$

where ϕ^i is the portion of the i^{th} POD basis vector corresponding to $\mathbf{x}$ and $\hat{\phi}^i$ is the portion corresponding to $\hat{\mathbf{x}}$.

Using the gappy POD formulation, equation (41) can be solved in a least squares sense in order to find the coefficients ν that best represent a given high-fidelity vector $\mathbf{x}$. Those coefficients can then be used in equation (42) to calculate the low-fidelity vector. Alternatively, if a mapping is desired from the low-fidelity space to the high-fidelity space, the coefficients are found from equation (42) and used in (41).

Unlike for space mapping, a single POD mapping is used for the objective and all constraints in all constraint-management methods. When incorporating this method into the TRMM framework, an additive or multiplicative correction must be used to ensure at least first-order consistency. These corrections are applied as shown in the beginning of this section.

IV. Barnes Problem

The first example considers a simple analytic test case. This problem was originally developed by G.K. Barnes as part of an M.S. thesis.[36] It was then used as a demonstration problem in a textbook,[37] and has since been used a number of times to demonstrate optimization approaches.[7,38–40] The objective function is

$$\begin{aligned} f &= -75.196 + 3.81x_1 - 0.126x_1^2 + 2.5056 \times 10^{-3}x_1^3 - 1.034 \times 10^{-5}x_1^4 6.83x_2 - 0.0302x_1x_2 \\ &+ 1.281 \times 10^{-3}x_2x_1^2 - 3.525 \times 10^{-5}x_2x_1^3 + 2.266 \times 10^{-7}x_2x_1^4 - 0.256x_2^2 3.46 \times 10^{-3}x_2^3 \\ &- 1.35 \times 10^{-5}x_2^4 + \frac{28.106}{x_2 + 1} + 5.237 \times 10^{-6}x_1^2x_2^2 + 6.3 \times 10^{-8}x_1^3x_2^2 + 1.663 \times 10^{-6}x_1x_2^3 + 2.867e^{0.0005x_1x_2} \end{aligned} \tag{43}$$

and the constraints are

$$c_1 = -(x_1 x_2/700 - 1) \leq 0, \quad (44)$$
$$c_2 = -(x_2/5 - x_1^2/625) \leq 0, \quad (45)$$
$$c_3 = -(x_2/50 - 1)^2 - (x_1/500 - 0.11) \leq 0. \quad (46)$$

In addition, the following bounds are imposed on the two design variables:

$$0 \leq x_1 \leq 80, \quad (47)$$
$$0 \leq x_2 \leq 80. \quad (48)$$

The problem has a global minimum at $\mathbf{x} = (80, 80)$ with an objective value of $f = -132.876$ and a local minimum at $\mathbf{x} = (49.526, 19.622)$ with an objective value of $f = -31.6372$. At that local minimum only the second constraint is active. Contours of the objective and lines showing the constraints are shown in Figure 1. The Barnes problem was chosen because it is two-dimensional and therefore easy to visualize, it has a number of interesting starting points, demonstrating tradeoffs between minimizing the objective and satisfying the constraints, and it has been used in previous studies.

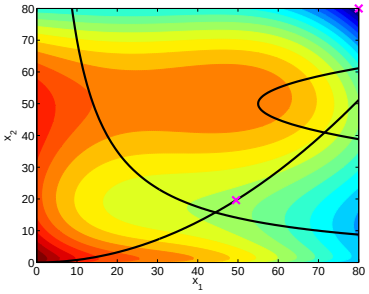

Figure 1. Objective function contours and constraints of the Barnes problem. Local and global optima are indicated with an 'x'.

The low-fidelity objective function is a third-order Taylor series of the high-fidelity function about the point $\mathbf{x} = (30, 40)$. The low-fidelity constraints are linear functions chosen to be in approximately the same location as the quadratic curves of the exact constraints over the global space. The equations for these low-fidelity constraints are

$$\hat{c}_1 = (-\hat{x}_1 - \hat{x}_2 + 50)/10 \leq 0, \quad (49)$$
$$\hat{c}_2 = (0.64\hat{x}_1 - \hat{x}_2)/6 \leq 0, \quad (50)$$
$$\hat{c}_3 = \begin{cases} 0.006\hat{x}_1 - 0.0134\hat{x}_2 + 0.34 \leq 0 & \text{if } \hat{x}_2 > 50, \\ 0.006\hat{x}_1 + 0.0134\hat{x}_2 - 1 \leq 0 & \text{if } \hat{x}_2 \leq 50. \end{cases} \quad (51)$$

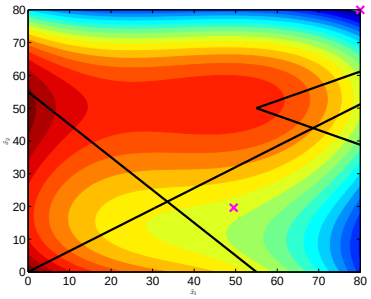

Figure 2. Objective function contours and constraints of the low-fidelity approximation to the Barnes problem. Optima of the high-fidelity problem are indicated with an 'x'.

The objective function contours and constraints used for the low-fidelity model are shown in Figure 2.

Three starting points were used. The point $\mathbf{x}^0 = (30, 40)$ is a feasible starting point, while $\mathbf{x}^0 = (65, 5)$ and $\mathbf{x}^0 = (10, 20)$ are infeasible starting points. At $\mathbf{x}^0 = (10, 20)$, the directions towards feasibility and optimality are near-orthogonal. The algorithm parameters used were $r_p^0 = 10$, $\eta_0 = 1$, $\kappa = 1$, $\gamma = 0.01$, $r_1 = 0.2$ and $r_2 = 0.9$.

Although for this case one could choose the trivial mapping $\mathbf{x} = \hat{\mathbf{x}}$, the corrected space mapping and POD mapping methods were used to generate mappings as described in Section III. For the POD method, 100 snapshots were generated on a uniform grid within the bounds. Two POD basis vectors were used. In this case, where the actual mapping is linear, linear combinations of those two vectors capture the entire mapping. For space mapping, the eight previous iterates were used as the control points. The linear space mapping of equation (33) was used.

Each of the constraint methods were run with each of the mapping methods, for a total of ten method combinations. In addition, for a benchmark, SQP was applied directly to the high-fidelity model in a single-fidelity approach. For those methods requiring Lagrange multipliers, an initial guess of $\lambda_i^0 = 0,\ i = 1, 2, 3$, was used for each multiplier. For MAESTRO, which requires initial slack variables, $z_i^0 = 1,\ i = 1, 2, 3$ was used. Table 1 shows the number of high-fidelity function evaluations needed by each method to reach a function value within 10^{-2} of the known local optimum with a constraint violation of no more than 10^{-3}. As can be seen in Table 1, direct surrogate optimization was able to converge more quickly than high-fidelity SQP for all cases considered, with an average reduction in the number of high-fidelity function calls of almost a factor of two. The SQP-like method also performed at least as well as SQP in the high-fidelity space, though the gains are more modest than direct surrogate optimization.

Approximating the Lagrangian with a single surrogate converged slowly, although it did converge from all starting points. It uses less information then any of the other methods, since it has only one correction

function. Consequently, it also has slightly less overhead than the other methods. However, in design problems of interest, overhead is insignificant relative to high-fidelity function evaluations, and is not considered a basis for comparison in these results. The Lagrangian with separate approximations was on average superior to the first method, but converged more slowly in two cases. The best approach between the two Lagrangian minimization methods presented in Section IIA can therefore be dependent on the initial guess and the problem. MAESTRO did not converge in the allowed 100 iterations for one of the starting points; for the others it was comparable to the Lagrangian method with separate surrogates for the objective and constraints. Although the relative performance of the two mapping methods varies among the starting points, on average corrected space mapping converged more quickly than POD mapping for three of the four constraint-management methods.

Constraint Management	Mapping Method	From (30,40) Calls	From (30,40) Ratio	From (10,20) Calls	From (10,20) Ratio	From(65,5) Calls	From(65,5) Ratio	**Average ratio**
SQP in high-fidelity space	N/A	10	1	14	1	10	1	**1**
Approx. whole Lagrangian	POD	26	2.6	62	4.429	12	1.2	**2.743**
Approx. whole Lagrangian	SM	33	3.3	21	1.5	17	1.7	**2.167**
Lagrangian with sep. approx.	POD	16	1.6	20	1.429	17	2.125	**1.718**
Lagrangian with sep. approx.	SM	23	2.3	23	1.643	14	1.4	**1.781**
Direct Surrogate Approximation	POD	4	0.4	8	0.571	8	0.8	**0.591**
Direct Surrogate Approximation	SM	7	0.7	7	0.5	5	0.5	**0.567**
SQP-like	POD	8	0.8	12	0.857	10	1	**0.886**
SQP-like	SM	8	0.8	8	0.571	8	0.8	**0.724**
MAESTRO	POD	12	1.2	N/A	N/A	17	1.7	**1.450**
MAESTRO	SM	12	1.2	N/A	N/A	17	1.7	**1.450**

Table 1. Convergence of variable complexity optimization methods. "Calls" refers to the number of high-fidelity function calls, both objective and constraints, along with their gradients, while "Ratio" gives the ratio of high-fidelity function calls required compared to SQP in the high-fidelity space.

Figures 3 and 4 show all the constraint management methods discussed, using POD mapping from the point $\mathbf{x}^0 = (10, 20)$. While the Lagrangian method with the single correction function takes small steps steadily towards the solution, both the separate approximations and direct surrogate optimization take larger steps, overshooting the optimum but returning to converge more quickly than the first method. Note that direct surrogate optimization first finds a feasible point, and then moves towards optimality. MAESTRO initially makes very good progress, but does not find the optimum in the allowed 100 iterations.

Figure 5 shows the most promising method, direct surrogate optimization, with both space mapping and POD mapping. This shows that both mapping methods have very similar performance, although they take different paths to the optimum.

V. Wing design problem

The second example considers incidence angle design of a wing. The wing is rectangular with an aspect ratio of 10. The objective function is the coefficient of induced drag, and the wing is constrained to have a coefficient of lift of at least 0.2. The optimization problem is given by

$$\begin{aligned} \min_{\alpha} \quad & 200 C_{Di}(\alpha) \\ \text{subject to} \quad & 10\,(0.2 - C_L(\alpha)) \leq 0 \\ & -10^\circ \leq \alpha_i \leq 10^\circ, \quad i = 1 \ldots 5, \end{aligned} \tag{52}$$

where scaling has been applied to both objective and constraint in order to improve numerical conditioning.

The high-fidelity code uses a vortex lattice method, using a zero thickness, constant collocation, doublet lattice model.[41] The method is similar to a vortex lattice method composed of horseshoe line vortices. By imposing Neumann boundary conditions and a zero-spanwise-vorticity trailing edge Kutta condition, the wing and wake surface doublet strengths can be uniquely determined. The discretization is performed using quadrilateral panels with uniform chordwise spacing and cosine spanwise refinement. A standard Trefftz

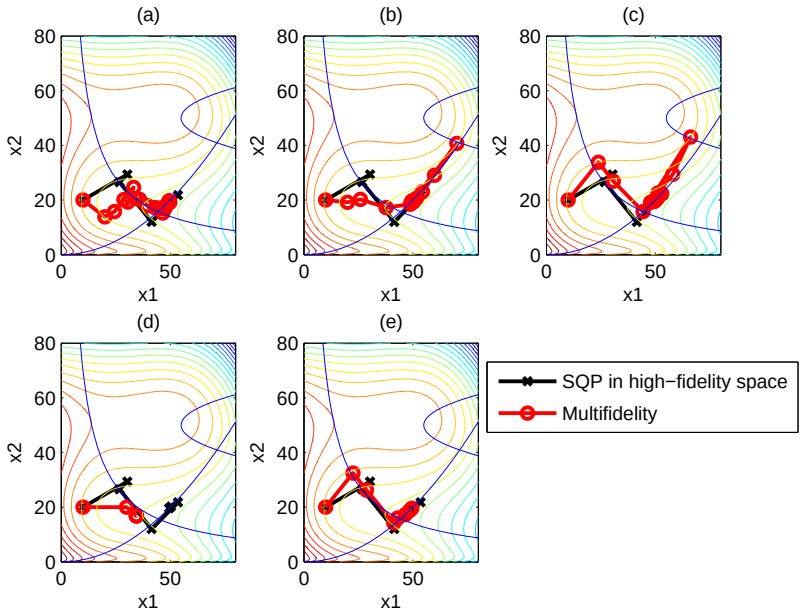

Figure 3. Paths of various constraint-management methods, using POD mapping from $\mathbf{x}^0 = (10, 20)$. (a) Lagrangian with single approximation, (b) Lagrangian with separate approximations for objective and constraints, (c) Direct surrogate optimization (d) MAESTRO (e) SQP-like variable complexity method. For comparison, the path of SQP working directly in the high-fidelity space is shown in black.

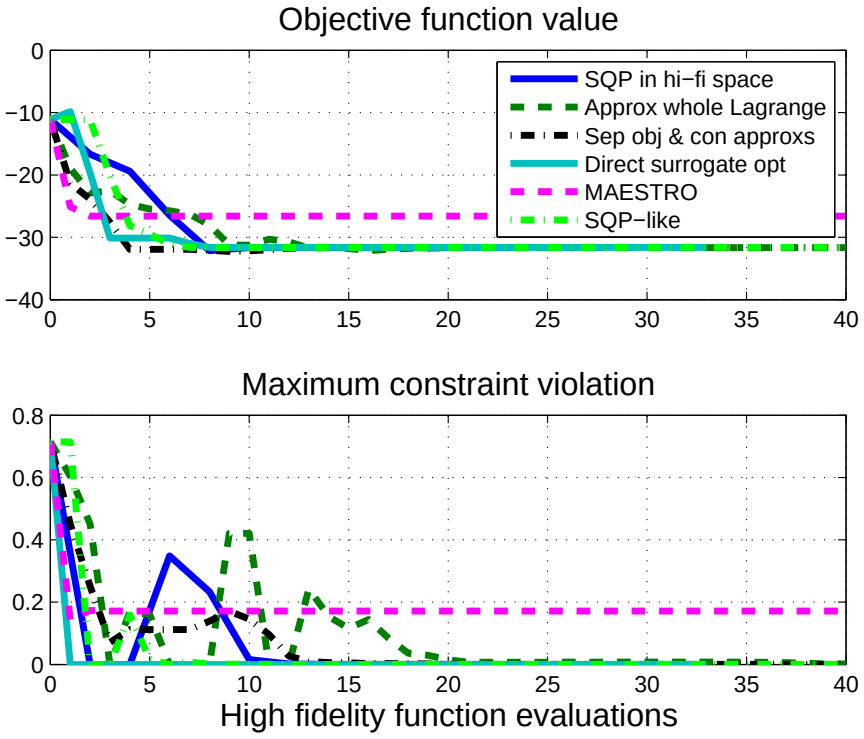

Figure 4. Function value and maximum constraint violation of various constraint-management methods, using POD mapping from $\mathbf{x}^0 = (10, 20)$.

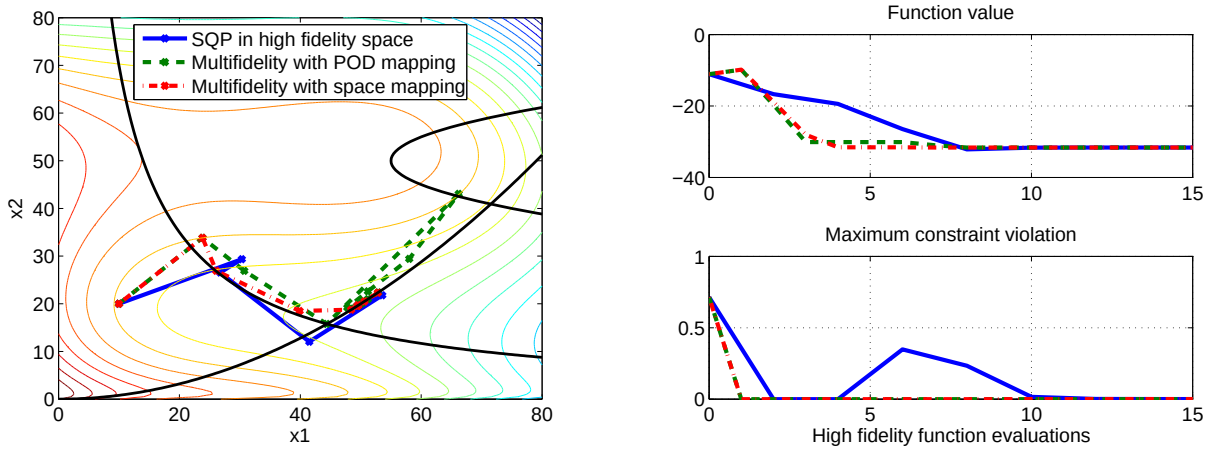

Figure 5. Paths, function value, and maximum constraint violation of direct surrogate optimization method, using space mapping and POD mapping, from $\mathbf{x}^0 = (10, 20)$.

Plane analysis[41] is used to compute both the lift and induced drag. The high-fidelity design variables are the angle of attack of the wing at each of five points, equally spaced from the root to the tip. A single analysis of the high-fidelity code takes approximately 90-100 seconds on a 2.4 GHz desktop workstation.

The low-fidelity code uses lifting-line theory. A standard lifting line method following Ref. 42 has been implemented to compute both the lift and induced drag. By assuming that the wing bound vorticity can be approximated by a single spanwise line vortex, the lifting line method is less effective for low aspect ratio wings than it is for higher aspect ratio wings. In its most general form, variations in both the incidence angle and chord with respect to spanwise position can be prescribed. The low-fidelity design variables are also angles of attack, but in this case at only three points, again equally spaced from the root to the tip. This problem is therefore an example of a variable complexity design: the low-fidelity design space is of lower dimension than the high-fidelity design space. A single analysis of the low-fidelity code takes approximately 30 milliseconds on a 2.4 GHz desktop workstation.

For both the high-fidelity and low-fidelity models, gradients were calculated using centered finite differences with a step size of 10^{-4} degrees. The evaluations required for these finite differences are included in the count of the number of function calls. The initial design used for optimization has a uniform angle of attack of three degrees. This point is feasible using both the high- and low-fidelity constraints. For the POD mapping method, 100 snapshots were generated using uniform Latin hypercube sampling[43] in the low-fidelity space, over the angle of attack range from -5 to 5 degrees. The high-fidelity portions of the training pairs were then created using linear interpolation. Two POD basis vectors were used. For space mapping, the eight previous iterates were used as the control points. The linear space mapping of equation (33) was used.

The Lagrangian TRMM method with separate approximations to objective function and constraints, direct surrogate optimization, and the SQP-like method were run, with each of the mapping methods. These three methods were chosen because they proved more efficient than the other methods on the Barnes problem. Again, SQP was applied directly in the high-fidelity space to provide a benchmark. The initial Lagrange multiplier was set to $\lambda^0 = 0$.

Table 2 shows the number of high-fidelity function evaluations needed by each method to reach a function value within 10^{-3} of the minimum function value found, with a constraint violation of no more than 10^{-3}. Five of the six variable complexity method combinations outperformed SQP applied directly in the high-fidelity space. The best method, direct surrogate optimization combined with POD mapping, has a 53% savings on the number of function evaluations required. On this problem, that is a savings of more than two and a half hours on a 2.4 GHz desktop workstation.

The lift distribution that yields minimum induced drag is elliptical.[44] Figure 6 shows the resulting lift distributions for SQP applied directly in the high-fidelity space and for direct surrogate optimization with both POD and space mapping. SQP in the high-fidelity space and direct surrogate optimization with POD have near-elliptical distributions. Direct surrogate optimization with corrected space mapping has a slight

Constraint Management	Mapping Method	Calls	Ratio
SQP in high-fidelity space	N/A	209	1
Lagrangian with sep. approx.	POD	340	1.627
Lagrangian with sep. approx.	SM	148	0.708
Direct surrogate optimization	POD	99	0.474
Direct surrogate optimization	SM	158	0.756
SQP-like	POD	113	0.541
SQP-like	SM	102	0.488

Table 2. Convergence of variable complexity optimization methods. "Calls" refers to the number of high-fidelity function calls, including those required to compute finite-difference gradients, while "Ratio" gives the ratio of high-fidelity function calls required compared to SQP in the high-fidelity space.

depression in lift near the root. However, according to the high-fidelity code, the coefficient of induced drag for this distribution is very close to the coefficient of induced drag for the elliptical distribution (0.12% higher). This is an indication that the high-fidelity code may have flat areas or multiple local minima, and is an example of the well-known tendency of optimization methods to find flaws in the analysis code. Figure 7 shows the objective function value and constraint violation of SQP in the high-fidelity space and direct surrogate optimization with both POD and corrected space mapping. It can be seen that for both mapping methods the objective is reduced at a similar rate, but the POD mapping reduces the magnitude of constraint violations more quickly than space mapping.

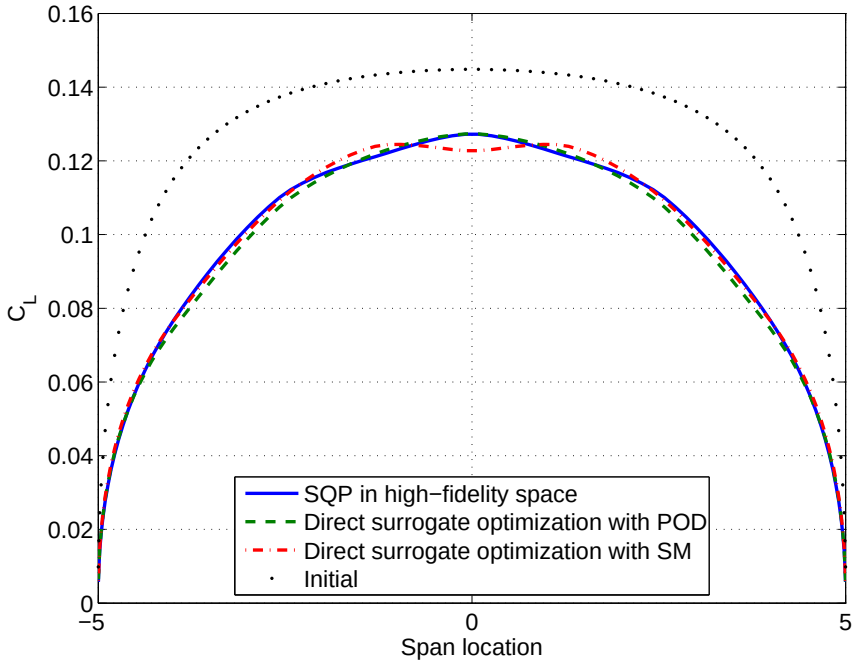

Figure 6. Lift distribution of optima found by three optimization methods: applying SQP in the high-fidelity space and applying direct surrogate optimization with each of space mapping and POD mapping. The initial lift distribution is also shown.

Figure 8 shows the three constraint-management methods with corrected space mapping, along with the benchmark SQP method. Again, while all three methods reduce the objective at a similar rate, they vary in the speed of reducing the constraint violations.

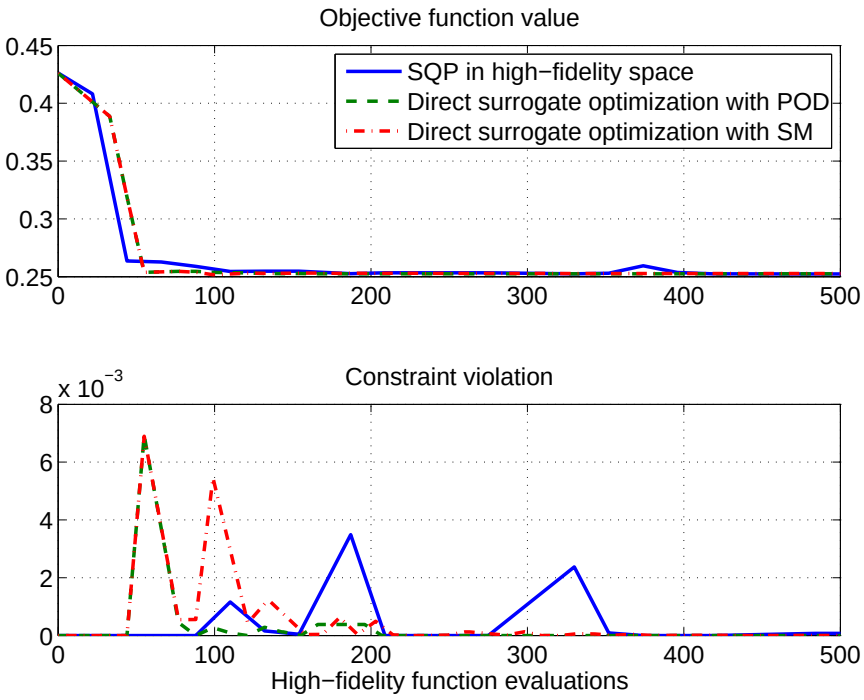

Figure 7. Function value ($200C_{Di}$) and maximum constraint violation $\max[0, 10(0.2 - C_L)]$ of applying SQP in the high-fidelity space and applying the direct surrogate optimization with each of space mapping and POD mapping.

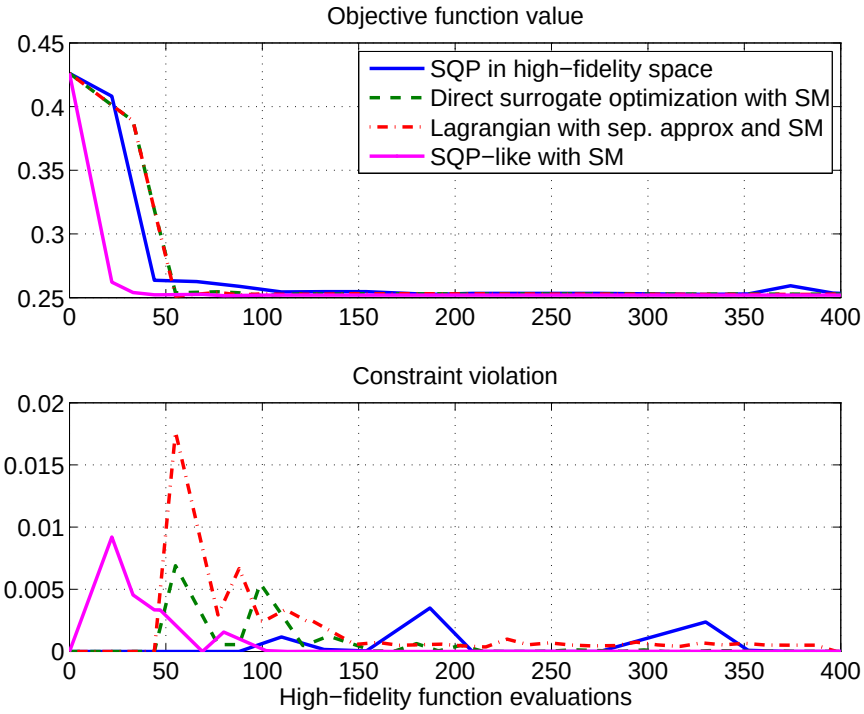

Figure 8. Function value ($200C_{Di}$) and maximum constraint violation $\max[0, 10(0.2 - C_L)]$ of applying SQP in the high-fidelity space and three multi-complexity methods with corrected space mapping.

VI. Conclusion

New methodology has been presented that extends surrogate-based optimization methods to variable-complexity constrained design problems. This methodology can manage a hierarchy of design models in the case where the design vector varies between the high-fidelity and low-fidelity models. Trust-region model-management methods using direct surrogate optimization or an SQP-like method were found to be the most efficient approach to manage constraints. When combined with POD mapping and applied to a variable-complexity wing-design optimization problem, direct surrogate optimization achieved 53% savings in high-fidelity function calls. These savings translated to a reduction in computational cost of more than two and a half hours.

Acknowledgement

The first two authors acknowledge the funding support of the Computational Engineering program of the Singapore-MIT Alliance. The authors also gratefully acknowledge Dr. Willis of MIT for his help with the lifting line and vortex lattice methods.

References

[1]Queipo, N. V., Haftka, R. T., Shyy, W., Goel, T., Vaidyanathan, R., and Tucker, P. K., "Surrogate-based analysis and optimization," *Progress in Aerospace Sciences*, Vol. 41, No. 1, January 2005, pp. 1–28.

[2]Conn, A. R., Gould, N. I., and Toiunt, P. L., *Trust-Region Methods*, MPS/SIAM Series on Optimization, Society for Interactive and Applied Mathematics, Philadelphia, 2000.

[3]Eldred, M., Giunta, S., and Collis, S., "Second-order corrections for surrogate-based optimization with model hierarchies," *Proceedings of the 10th AIAA/ISSMO Multidisciplinary analysis and optimization conference*, AIAA, Albany, New York, Aug 30. - Sept. 1 2004.

[4]Booker, A., Dennis, J., Frank, P., Serafini, D., Torczon, V., and Trosset, M., "A rigorous framework for optimization of expensive functions by surrogates," *Structural Optimization*, Vol. 17, 1999, pp. 1–13.

[5]Audet, C., Dennis, J., Moore, D. W., Booker, A., and Frank, P. D., "A surrogate-model-based method for constrained optimization," *Proceedings of the 8th AIAA/USAF/NASA/ASSMO Symposium on Multidisciplinary Analysis and Optimization*, Long Beach, CA, Sept. 6-8 2000, AIAA-2000-4891.

[6]Audet, C. and Dennis, J., "A pattern search filter method for nonlinear programming without derivatives," *SIAM Journal of Optimization*, Vol. 14, No. 4, 2004, pp. 980–1010.

[7]Rodriguez, J., Renaud, J., and Watson, L., "Trust region augmented Lagrangian methods for sequential response surface approximation and optimization," *Journal of Mechanical Design*, Vol. 120, No. 1, March 1998, pp. 58–66.

[8]Alexandrov, N., Lewis, R., Gumbert, C., Green, L., and Newman, P., "Optimization with Variable-Fidelity Models Applied to Wing Design," *Proceedings of the 38th Aerospace Sciences Meeting and Exhibit*, Reno, NV, January 2000, AIAA 2000-0841.

[9]Choi, S., Alonso, J., Kim, S., and Kroo, I., "Two-Level Multi-Fidelity Design Optimization Studies for Supersonic Jets," *43rd AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, January 2005, AIAA-2005-531.

[10]Robinson, T., Eldred, M., Willcox, K., and Haimes, R., "Strategies for Multi-fidelity Optimization with Variable Dimensional Hierarchical Models," AIAA paper 2006-1819, presented at the 2nd MDO Specialists Conference, Newport, RI, 2006.

[11]Barthelemey, J.-F. M. and Haftka, R., "Approximation Concepts for Optimum Structural Design – A Review," *Structural Optimization*, Vol. 5, No. 3, 1993, pp. 129–144.

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

[13]Simpson, T. W., Mauery, T. M., Korte, J. J., and Mistree, F., "Kriging Models for Global Approximation in Simulation-Based Multidisciplinary Design Optimization," *AIAA Journal*, Vol. 39, No. 12, December 2001, pp. 2233–2241.

[14]Jones, D. R., Schonlau, M., and Welch, W. J., "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, Vol. 13, No. 4, December 1998, pp. 455–492.

[15]Holmes, P. J., Lumley, J. L., Berkooz, G., Mattingly, J. C., and Wittenberg, R. W., "Low-dimensional models of coherent structures in turbulence," *Physics Reports*, Vol. 287, No. 4, 1997, pp. 337–384.

[16]Sirovich, L., "Turbulence and the Dynamics of Coherent Structures. Part 1 : Coherent Structures," *Quarterly of Applied Mathematics*, Vol. 45, No. 3, October 1987, pp. 561–571.

[17]Karpel, M., Moulin, B., and Love, M., "Modal-Based Structural Optimization with Static Aeroelastic and Stress Constraints," *Journal of Aircraft*, Vol. 34, No. 3, May-June 1997, pp. 433–440.

[18]Banerjee, J., "Exact modal analysis of an idealised whole aircraft using symbolic computation," *Aeronautical Journal*, Vol. 104, May 2000, pp. 247–255.

[19]Forrester, A. I., Bressloff, N. W., and Keane, A. J., "Optimization using surrogate models and partially converged computational fluid dynamics simulations," *Proceedings of the Royal Society A*, Vol. 462, March 2000, pp. 2177–2204.

[20]Lewis, R. M. and Nash, S. G., "A Multigrid Approach to the Optimization of Systems Governed by Differential Equations," *Proceedings of the 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Optimization*, Long Beach, CA, September 2000, AIAA Paper 2000-4890.

[21]Alexandrov, N., Lewis, R., Gumbert, C., Green, L., and Newman, P., "Approximation and model management in aerodynamic optimization with variable-fidelity models," *Journal of Aircraft*, Vol. 38, No. 6, November-December 2001, pp. 1093–1101.

[22]Alexandrov, N. M., Nielsen, E., Lewis, R., and Anderson, W., "First-Order Model Management with Variable-Fidelity Physics Applied to Multi-Element Airfoil Optimization," *Proceedings of the 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Long Beach, CA, September 2000, AIAA Paper 2000-4886.

[23]Alexandrov, N., Dennis, J., Lewis, R., and Torczon, V., "A Trust-Region Framework for Managing the Use of Approximation Models in Optimization," *Structural and Multidisciplinary Optimization*, Vol. 15, No. 1, February 1998, pp. 16–23.

[24]Yang, B., Yeun, Y., and Ruy, W., "Managing Approximation Models in Multiobjective Optimization," *Structural and Multidisciplinary Optimization*, Vol. 24, No. 2, September 2002, pp. 141–156.

[25]Rodriguez, J., Renaud, J., Wujek, B., and Tappeta, R., "Trust Region Model Management in Multidisciplinary Design Optimization," *Journal of Computational and Applied Mathematics*, Vol. 124, No. 1-2, December 2000, pp. 139–154.

[26]Broyden, C., "The convergence of a class of double rank minimization algorithms: Parts I and II," *Journal of the Institute of Mathematics and its Applications*, Vol. 6, 1970, pp. 76–90, 222–231.

[27]Fletcher, R., "A New Approach to Variable Metric Algorithms," *Computer Journal*, Vol. 13, 1970, pp. 317–322.

[28]Goldfarb, D., "A family of variable metric methods derived by variational means," *Mathematics of Computation*, Vol. 24, 1970, pp. 23–26.

[29]Shanno, D., "Conditioning of quasi-Newton methods for function minimization," *Mathematics of Computation*, Vol. 24, 1970, pp. 647–656.

[30]Rodriguez, J., Renaud, J., and Watson, L., "Convergence of trust region augmented Lagrangian methods using variable fidelity approximation data," *Structural Optimization*, Vol. 15, 1998, pp. 121–156.

[31]Lawson, C. and Hanson, R., *Solving Least-Squares Problems*, Prentice-Hall, 1974, p. 161.

[32]Bandler, J., Biernacki, R., Chen, S., Grobelny, P., and Hemmers, R., "Space Mapping Technique for Electromagnetic Optimization," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 42, 1994, pp. 2536–2544.

[33]Bandler, J., Cheng, Q., Dakroury, S., Mohamed, A., Bakr, M., Madsen, K., and Søndergaard, J., "Space Mapping: The State of the Art," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 52, No. 1, January 2004, pp. 337–361.

[34]Everson, R. and Sirovich, L., "The Karhunen-Loève Procedure for Gappy Data," *Journal of the Optical Society of America*, Vol. 12, No. 8, 1995, pp. 1657–1664.

[35]Bui-Thanh, T., Damodaran, M., and Willcox, K., "Aerodynamic Data Reconstruction and Inverse Design Using Proper Orthogonal Decomposition," *AIAA Journal*, Vol. 42, No. 8, August 2004, pp. 1505–1516.

[36]Barnes, G., Master's thesis, The University of Texas, Austin, Texas, 1967.

[37]Himmelblau, D., *Applied Nonlinear Programming*, McGraw Hill, 1972.

[38]Perez, V. M., Renaud, J. E., and Watson, L. T., "Adaptive Experimental Design for Construction of Response Surface Approximations," *Proceedings of the 42nd AIAA/ASME/ASCE/AHS/ASC Structures, Sttructural Dynamics, and Materials Conference and Exhibit*, Seattle, WA, April 16-19 2001, AIAA=2001-1622.

[39]Perez, V., Renaud, J., and Watson, L., "Homotopy curve tracking in approximate interior point optimization," *Proceedings of the 44th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Norfolk, VA, April 2003, AIAA-2003-1670.

[40]Perez, V., Eldred, M., and Renaud, J., "Solving the Infeasible Trust-Region Problem Using Approximations," *Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, New York, August 2004, AIAA-2004-4312.

[41]Katz, J. and Plotkin, A., *Low-Speed Aerodynamics*, Cambridge University Press, Cambridge, 2nd ed., 2001.

[42]Ashley, H. and Landahl, M., *Aerodynamics of Wings and Bodies*, Dover Publications, New York, 1985.

[43]McKay, M., Beckman, R., and Conover, W., "A comparison of three methods for selecting values of output variables in the analysis of output from a computer code," *Technometrics*, Vol. 21, 1979, pp. 239–245.

[44]Anderson, J., *Fundamentals of Aerodynamics, 2nd ed.*, McGraw-Hill, 1991.