AIAA JOURNAL
Vol. 43, No. 11, November 2005

Hybrid Variable Fidelity Optimization by Using a Kriging-Based Scaling Function

Shawn E. Gano* and John E. Renaud†
University of Notre Dame, Notre Dame, Indiana 46556-5637
and
Brian Sanders‡
U.S. Air Force Research Laboratory, Wright–Patterson Air Force Base, Ohio 45433-7531

Solving design problems that rely on very complex and computationally expensive calculations using standard optimization methods might not be feasible given design cycle time constraints. Variable fidelity methods address this issue by using lower-fidelity models and a scaling function to approximate the higher-fidelity models in a provably convergent framework. In the past, scaling functions have mainly been either first-order multiplicative or additive corrections. These are being extended to second order. In this investigation variable metric approaches for calculating second-order scaling information are developed. A kriging-based scaling function is introduced to better approximate the high-fidelity response on a more global level. An adaptive hybrid method is also developed in this investigation. The adaptive hybrid method combines the additive and multiplicative approaches so that the designer does not have to determine which is more suitable prior to optimization. The methodologies developed in this research are compared to existing methods using two demonstration problems. The first problem is analytic, whereas the second involves the design of a supercritical high-lift airfoil. The results demonstrate that the kriging-based scaling methods improve computational expense by lowering the number of high-fidelity function calls required for convergence. The results also indicate the hybrid method is both robust and effective.

Nomenclature

f	=	objective function
$\boldsymbol{g}$	=	inequality constraint
$\boldsymbol{h}$	=	equality constraint
$\boldsymbol{l}$	=	design space lower bounds
$\boldsymbol{u}$	=	design space upper bounds
$\mathcal{W}$	=	hybrid weighting value
$\boldsymbol{x}$	=	design vector
β	=	multiplicative scaling function
γ	=	additive scaling function
Δ	=	trust region size
ϵ_f	=	objective function convergence tolerance
ϵ_x	=	design variable convergence tolerance
ρ	=	trust region ratio
∇	=	gradient operator
∇^2	=	Hessian operator

Subscripts

high	=	high-fidelity model
i, j, k	=	free indices for elements in a vector or matrix
low	=	low-fidelity model
m	=	number of design variables
n	=	current iteration number
scaled	=	scaled low-fidelity value
u	=	unscaled constraint

Superscripts

T	=	transpose operator
$\tilde{}$	=	approximate function

I. Introduction

VARIABLE-FIDELITY and other model management methods have been developed to solve optimization problems that involve simulations with large computational expense. However, these methods can be used to optimize any problem where various fidelity models exist or can be constructed. In this context, the level of fidelity refers to the amount of physics or detail implemented within the model. A higher-fidelity model is one that contains physics or details that do not exist or are not accounted for in a lower-fidelity model. Based on this definition, the real system as it occurs in nature has infinite fidelity with an infinite number of input variables.

A major constraint to engineering design is cost. This cost is directly proportional to computer resources and the time required to run the optimization of simulations to predict system performance. Many times in the design process, various levels of model fidelity are developed, from low-order models for initial conceptual design to the full-physics-based models used for the final optimization analysis. The lower-fidelity models are typically much cheaper to evaluate, but designs produced by using these models neglect important physical effects included in the more expensive higher-fidelity models. The framework presented here attempts to build a scaling function that matches the result of the lower-fidelity models to the higher-fidelity models. The bulk of the optimizer's required function calls will be to the lower-fidelity models, which are updated by the scaling function. The methodology probably converges to the solution of the more expensive models with substantially less calls than would be required if optimization were done solely using the higher-fidelity model. The proof of convergence relies on, at least, a first-order matching between the models and the use of a trust region management scheme.

Early works that used variable-fidelity models for optimization were largely based on heuristics, and the processes were not guaranteed to converge to the high-fidelity solution. Consequently, some cases converged to the low-fidelity solution.[1,2] After realizing that significant differences could arise in the fidelity models, new

Received 26 July 2004; presented as Paper 2004-4460 at the AIAA/ISSMO 10th Multidisciplinary Analysis and Optimization Conference, Albany, NY, 30 August–1 September 2004; revision received 31 May 2005; accepted for publication 2 June 2005. Copyright © 2005 by John E. Renaud. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0001-1452/05 $10.00 in correspondence with the CCC.

*Ph.D. Candidate, Aerospace and Mechanical Engineering; sgano@nd.edu. Student Member AIAA.

†Professor, Aerospace and Mechanical Engineering; John.E.Renaud.2@nd.edu. Associate Fellow AIAA.

‡Program Manager, Advanced Structural Concepts Branch (AFRL/VASA); Brian.Sanders@wpafb.af.mil. Associate Fellow AIAA.

methods were developed that were proven to converge to the high-fidelity solution. Some of these methods used gradient information,[3] and others did not.[4] Haftka[5] devised a multiplicative scaling factor to update the value of lower-fidelity models to match the higher-fidelity models. Chang et al.[6] incorporated sensitivity information into the scaling factor. Alexandrov et al.[7,8] and Alexandrov and Lewis[9] combined this scaling approach and fundamental trust-region methods[10] into an approximation management framework (AMF). The combination of satisfying the first-order consistency condition along with other theoretical assumptions produced a provably convergent methodology. The AMF was demonstrated using various existing optimization routines such as the augmented Lagrangian method or sequential quadratic programming. Giunta and Eldred implemented a similar trust region based method into the DAKOTA project at the Sandia National Laboratory.[11] A pattern search-based framework was also developed by Booker et al.[12]

Response surface-based methods that use variable-fidelity data have been studied extensively by Rodríguez and Renaud[13] and Rodríguez et al.[14,15] In this work, coupled multidisciplinary design optimization problems were solved. Response surfaces were constructed using the fully coupled solutions and decoupled approximate solutions generated from the use of the global sensitivity equations developed by Sobieszczanski-Sobieski.[16] The response surfaces were sequentially optimized using the augmented Lagrangian method, and a trust region model management framework was used to prove convergence.[17]

Successive approximate optimization (SAO) algorithms generate response surfaces with a similar motivation as variable-fidelity methods, namely, to conduct optimization on a inexpensive surrogate model. SAO methods have been successful in reducing computational cost, especially when the sampling needed to create the response surface is on the order of the number of design variables.[18,19] Variable-fidelity methods can further reduce computational cost by using a function similar to a response surface, which is based more on physics—a low-fidelity model updated with a scaling function. The scaling function is essentially a response surface of the error between any two fidelity models.[20] The majority of the work in this investigation is concerned with how this model is constructed and managed.

Variable-fidelity methods require only one high-fidelity function and gradient evaluation per iteration, in comparison to typical SAO methods that use a sampling of high-fidelity calls to produce a low-fidelity surrogate. Variable-fidelity methods have another advantage over other adaptive response surface methods that use only high-fidelity samples: the low-fidelity models provide a global approximations that is locally corrected using scaling functions to produce a better overall approximation of the high-fidelity function.

Previous work dealing with managing the use of various fidelity models have used Taylor-series-based scaling functions to match the models. In this paper, a novel method for matching the various levels of fidelity to further decrease the computational expense of the design process using kriging models is developed and evaluated. A few motivations for using a kriging-based model, as opposed to a Taylor-series or polynomial-based approach, stem from the fact that these other methods use only local information to scale on a global level. The kriging approach allows all past information to be incorporated into the scaling function. Furthermore, polynomial scaling functions tend to diverge as the distance from the current design increases, whereas a kriging model typically behaves less dramatically. Additionally, a kriging model can easily incorporate any previous results obtained from runs completed using the various fidelity models prior to the optimization process.

There are a few other significant differences between the existing variable-fidelity methodologies and the ones presented in this research. In the existing first-order methods there is an underlying theme that the lower-fidelity model is updated to match the high-fidelity model using a local scaling function. This paper uses the premise that the lower-fidelity model approximates the high-fidelity model on a global scale, and, therefore, the scaling function to match the two models should be based more on a global scope. A well-known method for constructing global approximations that is suitable for this application is kriging. A second extension of variable-fidelity optimization is the hybrid approach, which combines the additive and multiplicative approaches. The final extension is the use of variable metric methods to reduce the cost of second-order methods.

The variable-fidelity optimization problem can be written in its most general form using the standard nonlinear programming problem.

$$\underset{x}{\text{Minimize:}}$$

$$f(\boldsymbol{x})$$

Subject to:

$$\boldsymbol{g}(\boldsymbol{x}) \leq 0, \qquad \boldsymbol{g}_u(\boldsymbol{x}) \leq 0, \qquad \boldsymbol{h}(\boldsymbol{x}) = 0, \qquad \boldsymbol{h}_u(\boldsymbol{x}) = 0$$

$$\boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}$$

where f is the objective function, x are the design variables, $\boldsymbol{g}$ is the vector of inequality constraints, $\boldsymbol{h}$ is the vector of equality constraints, the subscript u means the constraint is unscaled, and $\boldsymbol{l}$ and $\boldsymbol{u}$ are the upper and lower bounds, respectively, of the design variables. In the general case, unscaled constraints are included in the optimization and are unaffected by the various fidelity models. An example of such constraints is a constraint solely based on the design variables (i.e., geometry of the design). Scaling these constraints can unnecessarily slow convergence. Furthermore, the objective and the constraints can be evaluated using various fidelity models, and the solution is desired of the highest-fidelity model.

In this paper, an overview of variable-fidelity frameworks is described first. Next, the first- and second-order Taylor-series-based approaches are given and discussed. The following section outlines an approach to combine the additive and multiplicative scaling method into a single hybrid method. Then the kriging-based method is introduced and its differences to existing methods highlighted. The kriging-based approach is then compared to the others using two test problems. The paper concludes by comparing the results of the test problems, identifying strengths and weaknesses of using this approach, and commenting on future work in this area.

II. Trust Region Managed Variable-Fidelity Framework

The typical framework for variable-fidelity optimization is depicted in Fig. 1 and is based, in part, on work done by Alexandrov et al.[21] The following process describes the basic steps of the framework:

Step 1, Initialization: The objective and constraints are evaluated using both the high- and low-fidelity models at the starting design point $\boldsymbol{x}_0$. Also an initial l_1 penalty function is evaluated (see step 5).

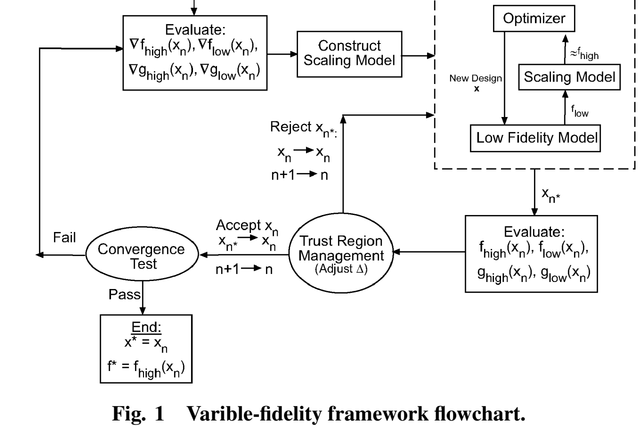

Starting Design x_0
Evaluate: $f_{high}(x_0)$, $f_{low}(x_0)$, $g_{high}(x_0)$, $g_{low}(x_0)$
Evaluate: $\nabla f_{high}(x_n)$, $\nabla f_{low}(x_n)$, $\nabla g_{high}(x_n)$, $\nabla g_{low}(x_n)$
Construct Scaling Model
Optimizer
$\approx f_{high}$
New Design x
Scaling Model
f_{low}
Low Fidelity Model
Reject x_{n*}: $x_n \rightarrow x_n$ $n+1 \rightarrow n$
x_{n*}
Fail
Convergence Test
Accept x_n $x_{n*} \rightarrow x_n$ $n+1 \rightarrow n$
Trust Region Management (Adjust Δ)
Evaluate: $f_{high}(x_n)$, $f_{low}(x_n)$, $g_{high}(x_n)$, $g_{low}(x_n)$
Pass
End: $x^* = x_n$ $f^* = f_{high}(x_n)$

Fig. 1 Varible-fidelity framework flowchart.

Step 2, Gradient evaluation: The gradient of the objective and the Jacobian for the constraints are evaluated using both the high- and low-fidelity models at the current design point $\boldsymbol{x}_n$.

Step 3, Construct scaling model: A scaling model is constructed to ensure matching between the fidelity models. This model can be based on many different methods; additive and multiplicative are the most common. Each method can be modeled as first order, second order, or kriging based. A scaling model is constructed for each constraint as well as for the objective function.

Step 4, Optimize scaled low-fidelity model: The low-fidelity model scaled with the scaling model constructed in step 3 is optimized. The choice of optimizer used is based on preference. In the work done by Alexandrov et al.,[8] three optimizers were compared: augmented Lagrangian method, multilevel algorithms for large-scale constrained optimization (MAESTRO)[7] (used for coupled MDO problems), and sequential quadratic programming (SQP). For typical single discipline problems, Alexandrov found SQP to be the most promising, and it is used in this research. The unscaled constraints are included in this step to ensure that they are always satisfied.

Step 5, Evaluate new design and l_1 penalty function: Using the resulting design point from step 4, the high-fidelity objective and constraints are evaluated. The objective and constraint values are used to calculate a current value of the l_1 penalty function P for the high and scaled low-fidelity models. The penalty function is defined as

$$P(\boldsymbol{x}) = f(\boldsymbol{x}) + \frac{1}{\mu_n}\sum \max[0, g_i(\boldsymbol{x})] + \frac{1}{\mu_n}\sum |h_i(\boldsymbol{x})| \tag{1}$$

where μ is the penalty weight that is typically decreased by a factor of 10 each time a new point is accepted. This penalty weighting drives all of the active constraints to zero as the algorithm converges.

Step 6, Trust region management: To guarantee convergence of the variable-fidelity optimization framework, a trust region model management strategy is employed.[22] This method provides a means for adaptively managing the allowable move limits for the approximate design space. Originally these methods were used to ensure the convergence of Newton-based methods.

A trust region ratio allows the trust region model management framework to monitor how well the approximation matches the high-fidelity design space. After each completed optimization on the scaled low-fidelity model, a new candidate point $\boldsymbol{x}_n^*$ is found. A trust region ratio ρ_n is calculated at this new point:

$$\rho_n = \frac{P(\boldsymbol{x}_n)_{\text{high}} - P\left(\boldsymbol{x}_n^*\right)_{\text{high}}}{P(\boldsymbol{x}_n)_{\text{scaled}} - P\left(\boldsymbol{x}_n^*\right)_{\text{scaled}}} \tag{2}$$

where $P(\,)_{\text{high}}$ and $P(\,)_{\text{scaled}}$ are the l_1 penalty functions for the high and scaled low-fidelity models and the point $\boldsymbol{x}_n$ was the initial point of the optimization. Notice that by definition $P(\boldsymbol{x}_n)_{\text{scaled}} = P(\boldsymbol{x}_n)_{\text{high}}$ because the scaled low-fidelity model matches the high-fidelity model at that point. This is the ratio of the actual change in the function to the predicted change of the function by the scaled lower-fidelity model. Because the constraints are also approximated, the trust region ratio must account for this and converge to a feasible design. That is the reasoning behind using the l_1 penalty function.

The trust region size is governed by the following standard rules[11,17]:

$$\Delta_{n+1} = \begin{cases} c_1\Delta_n : \rho_n \leq R_1 \\ \;\Delta_n \;\, : R_1 < \rho_n < R_2 \\ \Gamma\Delta_n : R_2 \leq \rho_n < R_3 \end{cases} \tag{3}$$

where $\Gamma = c_2$ if $\|x_k^* - x_{c_k}\|_\infty = \Delta_k$ otherwise $\Gamma = 1$. A typical set of values for the range limiting constants is $R_1 = 0.25$, $R_2 = 0.75$, and $R_3 = 1.25$, whereas the trust region multiplication factors are typically $c_1 = 0.25$ and $c_2 = 3$. Physically, ρ represents how good of an approximation our scaled low-fidelity model is compared to the high-fidelity model. If ρ is near one, the approximation is quite good. If ρ is near zero, the approximation is not as good, but still captures the minimization trend. If ρ is negative, then the point is a worse design. In this case the point is rejected, the trust region size is reduced by the factor c_1, and the algorithm returns to step 4. As long as $\rho > 0$, the point is accepted, and the algorithm proceeds to step 7.

Step 7, Convergence test: For the implementation used in this research, convergence was determined by the following stopping criterion:

$$f_{\text{high}}(\boldsymbol{x}_n) - f_{\text{high}}(\boldsymbol{x}_{n-1}) < \epsilon_f \tag{4}$$

$$\|\boldsymbol{x}_n - \boldsymbol{x}_{n-1}\| < \epsilon_x \tag{5}$$

where ϵ_f and ϵ_x are tolerances supplied by the user and n is the current iteration counter. If any of the two inequalities at the current point is true, the algorithm is considered converged. If the convergence test is true, then the final design is found; otherwise, the algorithm returns to step 2.

Convergence to a Karush–Kuhn–Tucker (KKT) point can tested at the final design point by evaluating the projected gradient as is done by Rodríguez and Renaud. If the projected gradient at the final design is zero, then the design satisfies the KKT conditions.

III. Scaling Methods

Variable-fidelity or approximate model management frameworks can be derived from an infinite number of scaling functions; however, existing methods have generally used two varieties: multiplicative or additive. Currently, the most common is the multiplicative framework, devised by Alexandrov and Lewis[9] based on Haftka's[5] scaling function. The additive method was presented by Lewis and Nash.[23] Both methods are based on constructing an unknown function to update the lower-fidelity model, which, in turn, will approximate the higher-fidelity model.

A. Multiplicative Scaling

A given set of high- and low-fidelity models $f_{\text{high}}(\boldsymbol{x})$ and $f_{\text{low}}(\boldsymbol{x})$ can be matched by multiplying the low-fidelity model by an unknown function $\beta(\boldsymbol{x})$. This is posed mathematically as

$$f_{\text{high}}(\boldsymbol{x}) = \beta(\boldsymbol{x}) f_{\text{low}}(\boldsymbol{x}) \tag{6}$$

This scaling model was first proposed and used for approximating structural response by Haftka.[5] Solving for the unknown multiplicative scaling function results in

$$\beta(\boldsymbol{x}) = \frac{f_{\text{high}}(\boldsymbol{x})}{f_{\text{low}}(\boldsymbol{x})} \tag{7}$$

From inspection of Eq. (7), it is clear that the function $\beta(\boldsymbol{x})$ is the scaling ratio of the high-fidelity model to the low-fidelity model, and when it is multiplied by the low-fidelity model the response of the high-fidelity model is achieved.

The multiplicative scaling model is unknown and must be approximated. A standard way to do this is to expand Eq. (7) about a current design point $\boldsymbol{x}_n$ by using a Taylor series to first order:

$$\tilde{\beta}(\boldsymbol{x}) = \beta(\boldsymbol{x}_n) + \nabla\beta(\boldsymbol{x}_n)^T(\boldsymbol{x} - \boldsymbol{x}_n) \tag{8}$$

To evaluate this, the gradient of β is needed and can be obtained by differentiating Eq. (7), resulting in

$$\nabla\beta(\boldsymbol{x}_n) = \begin{bmatrix} \dfrac{f_{\text{low}}(\boldsymbol{x}_n)(\partial f_{\text{high}}/\partial x_1)|_{x=x_n} - f_{\text{high}}(\boldsymbol{x}_n)(\partial f_{\text{low}}/\partial x_1)|_{x=x_n}}{f_{\text{low}}(\boldsymbol{x}_n)^2} \\ \vdots \\ \dfrac{f_{\text{low}}(\boldsymbol{x}_n)(\partial f_{\text{high}}/\partial x_m)|_{x=x_n} - f_{\text{high}}(\boldsymbol{x}_n)(\partial f_{\text{low}}/\partial x_m)|_{x=x_n}}{f_{\text{low}}(\boldsymbol{x}_n)^2} \end{bmatrix} \tag{9}$$

This first-order model ensures that at the design point $\boldsymbol{x}_n$ the updated low-fidelity model matches the high-fidelity model to first order. The identical process is done to scale each constraint.

B. Additive Scaling

A given set of high- and low-fidelity models $f_{\text{high}}(\boldsymbol{x})$ and $f_{\text{low}}(\boldsymbol{x})$ can also be matched by adding the low-fidelity model to an unknown function $\gamma(\boldsymbol{x})$. The additive method was used by Lewis and Nash[23] to solve multigrid problems but can be used more generally. The additive correction is expressed mathematically as

$$f_{\text{high}}(\boldsymbol{x}) = f_{\text{low}}(\boldsymbol{x}) + \gamma(\boldsymbol{x}) \tag{10}$$

The additive scaling function $\gamma(\boldsymbol{x})$ is found by subtracting the low-fidelity function from both sides:

$$\gamma(\boldsymbol{x}) = f_{\text{high}}(\boldsymbol{x}) - f_{\text{low}}(\boldsymbol{x}) \tag{11}$$

From Eq. (11), it is clear that the function $\gamma(\boldsymbol{x})$ is the additive scaling of the high-fidelity model to the low-fidelity model, or the error between them. When this function is added to the low-fidelity model, the response of the high-fidelity model is produced. A similar function for the constraints can be developed in the same manner as Eqs. (10) and (11).

A first-order approximation of the additive scaling model is found using the same approach as in the multiplicative case. The additive scaling function is approximated about a current design point using a first-order Taylor-series expansion:

$$\tilde{\gamma}(\boldsymbol{x}) = \gamma(\boldsymbol{x}_n) + \nabla\gamma(\boldsymbol{x}_n)^T(\boldsymbol{x} - \boldsymbol{x}_n) \tag{12}$$

Evaluating this requires gradient information, which can be obtained by differentiating Eq. (11) to get

$$\nabla\gamma(\boldsymbol{x}_n) = \nabla f_{\text{high}}(\boldsymbol{x}_n) - \nabla f_{\text{low}}(\boldsymbol{x}_n) \tag{13}$$

The first-order additive model again ensures that at the current design point the updated low-fidelity model matches both the function and the gradient of the high-fidelity model exactly, which is required for proof of convergence of the overall variable-fidelity optimization method.

IV. Second-Order and Approximate Second-Order Scaling Models

Using the same idea as in the first-order methods, an approximation scaling function can be derived to match second-order information. Higher-order models were mentioned by Chang et al.[6] but were not used because of the cost of obtaining higher-order information. Second-order scalings were first used by Gano et al.[24] and Eldred et al.[25] using both second-order information and approximate second-order information. The approach is analogous to the first-order method except the Taylor-series approximation is expanded to include the second-order terms as the name implies. The result for the multiplicative method is

$$\tilde{\beta}(\boldsymbol{x}) = \beta(\boldsymbol{x}_n) + \Delta\boldsymbol{x}^T\nabla\beta(\boldsymbol{x}_n) + \tfrac{1}{2}\Delta\boldsymbol{x}^T\nabla^2\beta(\boldsymbol{x}_n)\Delta\boldsymbol{x} \tag{14}$$

where $\Delta\boldsymbol{x} = \boldsymbol{x} - \boldsymbol{x}_n$. Using the same gradient result as in Sec. III.A, the only remaining term needed is the Hessian of β; this can be found by differentiating Eq. (9) again, which simplifies to

$$\nabla^2\beta(\boldsymbol{x}_n) = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,m} \\ h_{2,1} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ h_{m,1} & \cdots & \cdots & h_{m,m} \end{bmatrix} \tag{15}$$

where

$$h_{i,j} = \frac{1}{f_{\text{low}}^3}\left[2f_{\text{high}}\frac{\partial f_{\text{low}}}{\partial x_i}\frac{\partial f_{\text{low}}}{\partial x_j} + f_{\text{low}}^2\frac{\partial^2 f_{\text{high}}}{\partial x_i \partial x_j} - f_{\text{low}}\left(\frac{\partial f_{\text{low}}}{\partial x_j}\frac{\partial f_{\text{high}}}{\partial x_i} + \frac{\partial f_{\text{high}}}{\partial x_j}\frac{\partial f_{\text{low}}}{\partial x_i} + f_{\text{high}}\frac{\partial^2 f_{\text{low}}}{\partial x_i \partial x_j}\right)\right] \tag{16}$$

where all of the functions and partial derivatives are evaluated at the point $\boldsymbol{x}_n$ and i and j are the indices for the Hessian matrix, which run from 1 to the number of design variables.

For the additive method, the second-order expansion is

$$\tilde{\gamma}(\boldsymbol{x}) = \gamma(\boldsymbol{x}_n) + \Delta\boldsymbol{x}^T\nabla\gamma(\boldsymbol{x}_n) + \tfrac{1}{2}\Delta\boldsymbol{x}^T\nabla^2\gamma(\boldsymbol{x}_n)\Delta\boldsymbol{x} \tag{17}$$

The first-order information was found in Sec. III.B. Again, the only remaining information needed is the Hessian of γ; this can be found by taking the derivative of Eq. (13):

$$\nabla^2\gamma(\boldsymbol{x}_n) = \nabla^2 f_{\text{high}}(\boldsymbol{x}_n) - \nabla^2 f_{\text{low}}(\boldsymbol{x}_n) \tag{18}$$

The scaling functions have the same form as those in the first-order methods and can similarly be computed for the constraints as well as the objective function. Computing the symmetric full-rank Hessian matrices of either of the second-order methods would be quite expensive, even if gradient information were readily available.

The second-order information that is needed in both second-order techniques can be very costly to compute. There exist techniques to approximate the second-order information from first-order information, which is calculated at each iteration of the variable-fidelity optimization process. In this investigation the second-order information can, therefore, be obtained at no additional cost in terms of function calls per iteration, compared to the first-order scaling methods. The two most prevalent methods used are the Broyden–Fletcher–Goldfarb–Shanno[26–29] (BFGS) update and the symmetric-rank-1 (SR1) update.[30] The main difference between these two methods is that the BFGS maintains a positive-definite approximation of the Hessian, whereas SR1 does not.

V. Adaptive Hybrid Scaling: Combining Additive and Multiplicative Methods

In this investigation a new adaptive hybrid scaling method is developed. Previous studies have used either a multiplicative or additive scaling function to update the low-fidelity response to approximate the high-fidelity model. In general, some suites of fidelity models are matched better using one method or the other, and there is no way to know this a priori. This section presents a novel methodology that could be more robust by including both types of scaling.

Combining both the multiplicative and additive scaling functions such that they still properly scale the low-fidelity model to match the high-fidelity model requires the use of a weighted average of the two methods. Using a weighting term $\mathcal{W}$, this sum is

$$f_{\text{high}}(\boldsymbol{x}) = \mathcal{W}f_{\text{low}}(\boldsymbol{x})\beta(\boldsymbol{x}) + (1 - \mathcal{W})[f_{\text{low}}(\boldsymbol{x}) + \gamma(\boldsymbol{x})] \tag{19}$$

To determine the value of $\mathcal{W}$, a further condition must be enforced. Concurrent work being done by Eldred et al.[25] proposes using a previously evaluated point to adjust the value of $\mathcal{W}$ such that the model passes through that point as well. The weighting function then takes the value

$$\mathcal{W} = \frac{f_{\text{high}}(\boldsymbol{x}_{\text{pp}}) - [f_{\text{low}}(\boldsymbol{x}_{\text{pp}}) + \gamma(\boldsymbol{x}_{\text{pp}})]}{f_{\text{low}}(\boldsymbol{x}_{\text{pp}})\beta(\boldsymbol{x}_{\text{pp}}) - [f_{\text{low}}(\boldsymbol{x}_{\text{pp}}) + \gamma(\boldsymbol{x}_{\text{pp}})]} \tag{20}$$

In Eq. (20) the current additive and multiplicative scaling functions are used along with any previous point $\boldsymbol{x}_{\text{pp}}$, where the high-fidelity model was evaluated. There is some freedom in choosing the past point. One option is to simply use the last accepted design point. However, for this work the nearest point is used. The advantage of using the nearest point is that it could have been a design that was evaluated but rejected; this would help keep the next iteration from moving in this undesired direction. A new weighting value can be computed both for the objective and each constraint at each iteration. Updating these weights at each iteration allows the framework to *adapt* to the best model for the current area of the design space.

Both $\beta(\boldsymbol{x})$ and $\gamma(\boldsymbol{x})$ can be of any form: for example, first-order, second-order, or kriging models. The weighted averaging maintains the Taylor-series matching for this model and, therefore, retains the convergence properties.

The adaptive hybrid method has more overhead in memory usage and CPU time in computing both scaling functions, though this extra computational cost is negligible compared to a high-fidelity function call. The extra expense of this method is offset substantially if just one high-fidelity call is saved.

VI. Kriging-Based Scaling Models

The scaling models developed in the first- and second-order approaches are only local to the current design point and do not use past information, except for the construction of the second-order information using the BFGS or SR1 methods. When using variable-fidelity physics-based models, the low-fidelity model typically is a global model.[31] Likewise, a global scaling function might be better at approximating the high-fidelity response. In this investigation a new kriging-based scaling function is developed to improve the scaling between the different fidelity models on a more global scale. This approach allows the use of all information calculated throughout the course of the optimization, even when the trust region test fails. Figure 2 depicts the difference between the Taylor-series and kriging-based approaches. The figure explicitly shows how past information can be used to generate a better scaling model. The kriging model can be constructed for any type of scaling function, for example, the additive or multiplicative methods already discussed.

It is important to reiterate that the savings of modeling the error, as opposed to simply modeling the high-fidelity function itself, comes from the fact that the error model can be applied to a physics-based low-fidelity model. This scaled low-fidelity model provides a better approximation of the high-fidelity model than simply using a mathematically constructed interpolation or data-fitting model. Furthermore, creating a model of the error, which is used as a correction, requires fewer high-fidelity samples as compared to building a response surface from high-fidelity samples alone.

The kriging model gives exact responses at sample points, as it is an interpolating function. This ensures that at least zero-order matching is obtained. With the inclusion of gradient information, first-order matching is achieved. Combining first-order matching, a trust region model management strategy, and an optimizer that converges to a KKT point of the scaled low-fidelity problem provides for a provably convergent framework.[21]

Building and rebuilding the kriging models take extra time and memory storage. This added computational time and resource expenditure might be negligent, however, compared to the evaluation of the high-fidelity model. The method used to build the kriging models is described in the following subsection.

Another benefit of using a kriging-based scaling approach is that past data can be easily incorporated into the scaling model to further increase the convergence rate. Often, a model is evaluated for various purposes before an optimization is performed. These results can be included in the kriging model to improve its matching capabilities.

A. Kriging Overview

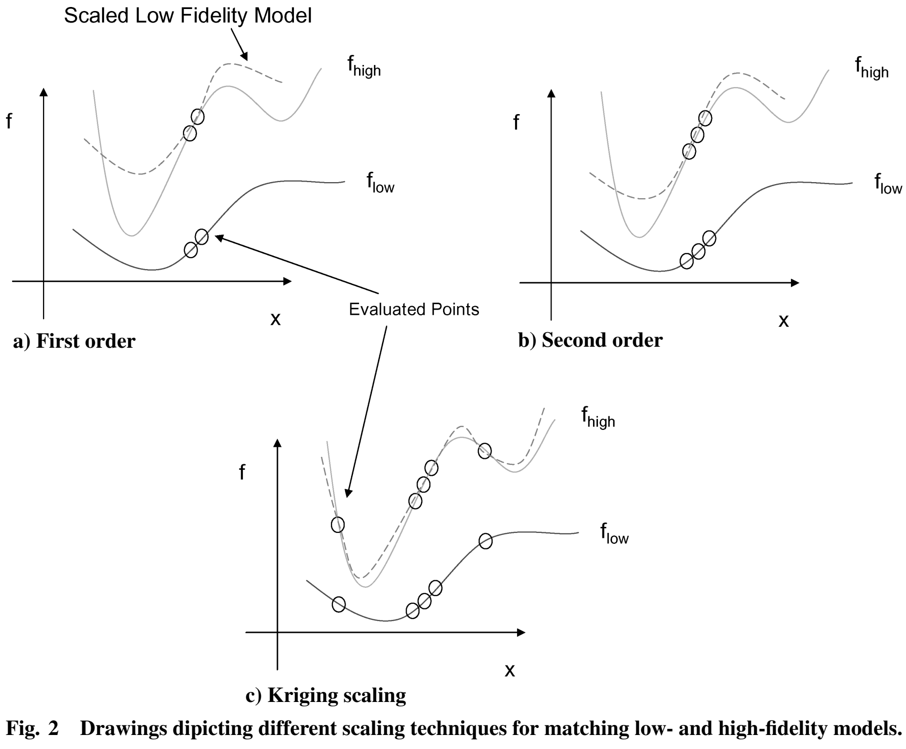
Kriging was developed for use in the field of geological statistics and was used in estimating unknown values from known values obtained via a semivariogram. This helped give a best-guess global view of desired parameters from a set of known values. The term *kriging* was named after D. G. Krige from South Africa, who used this method with much success (see Cressie[32] for details). The method was originally developed for use in the computer science and engineering fields as design and analysis of computer experiments or DACE modeling by Sacks et al.[33]

Kriging methods have been used to model the response of many engineering systems, including the design of a low-boom business jet by Chung and Alonso.[34] Martin and Simpson conducted a study on using kriging models to approximate deterministic computer models and discussed the applicability of various kriging variants.[35,36]

The kriging scaling functions are used in the same manner as the Taylor-series-based methods: to scale the low-fidelity model to match the high-fidelity model. However, the kriging model is built using all of the points at which the models have been evaluated, including gradient information or finite difference samples. The details of constructing kriging models have been thoroughly described by past researchers.[37–43]

B. Warm Starting the Kriging Method

A possible downside to using a kriging method to model the scaling between the variable-fidelity models is the fact that during the first few iterations the kriging model might not have enough points to accurately model this scaling function. To accelerate this possible dilemma, a warm-started kriging-based method is developed. As illustrated in Fig. 3, this method starts using a Taylor-series-based method, either first or second order, and then switches to the kriging model. When the Taylor-series-based method is used, all information is saved, which allows the kriging model to be constructed



a) First order

b) Second order

c) Kriging scaling

Fig. 2 Drawings dipicting different scaling techniques for matching low- and high-fidelity models.

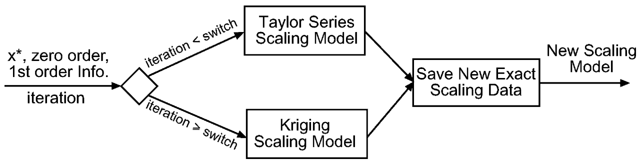


Fig. 3 Scaling function generation using the warm-started kriging-based method.

initially from many more sample points and, therefore, with higher precision. This warm-started approach is not limited by the scaling function used. One other item that should be noted about this method is that a switch between methods only occurs if the point has been accepted in the previous iteration; otherwise, the methodology shrinks the trust region space and uses the same model until a new point is accepted, and then the model is switched.

VII. Supplying Gradient Information

If low-fidelity gradient information is readily available and a gradient-based inner optimizer is used, finding the gradients of the scaled low-fidelity model is a straightforward process. Having the scaled low-fidelity gradients allows for a decrease in function calls by the optimizer because finite differencing techniques are not required. This is especially important if the low-fidelity model has a significant computational cost.

To find the gradients of the scaled low-fidelity models, the appropriate scaling method, additive or multiplicative for example, must be differentiated. Then either the first-order, second-order, or kriging models must be differentiated with respect to their expansion variables. The kriging model is a special case; its gradients can be found by other means as described by Lophaven et al.[40]

VIII. Optimization Demonstrations

To demonstrate the savings of the approximate second-order scaling, kriging-based scaling, and hybrid methods developed in this research, two problems are solved. The first problem is a simple two-dimensional analytic problem that is computationally inexpensive and is useful for understanding how the various methods are different. The second problem is a realistic engineering design problem that uses complex computational models. In this problem, an energy-efficient-transport (AEET) high-lift airfoil is designed by placing the slats and flap components to maximize its lifting capability in slow flight.

The goal of these demonstrations is to compare and contrast the various variable-fidelity scaling methods. To compare the various methods, we will take a look primarily at the number of high-fidelity function calls. Other aspects of the methods that will be compared are the number of low-fidelity function calls, the final trust region size, and the number of iterations performed. The number of low-fidelity function calls might be important, depending on the computational expense of the low-fidelity model relative to the high-fidelity model. The trust region size at the end of the optimization gives an insight into how well the scaling function performed: the larger the region the better. For comparison purposes, the number of function calls needed for a standard SQP optimization performed on the high-fidelity model alone is also presented for each problem. The SQP solver used was MATLAB's fmincon,[44] which is included in the optimization toolbox.

A. Two-Dimensional Analytic Problem

This test problem serves two main purposes: it first allows for verification that the implementations converge and work properly, and second it demonstrates the improvement of the proposed methods over the existing methods. The problem is analytic and two dimensional, making it easy to create contour plots of the design space for a given iteration. These contour plots can be used to visually compare the high-fidelity space as well as the scaled low-fidelity space. The problem is given in Eqs. (21–24). The initial design point was $[1.5, 1.5]^T$, and both design variables were bounded between 0.1 and 10. The low-fidelity model adds linear and nonlinear noise factors to the high-fidelity model to change the shape of the design space and location of the optima. For this problem all gradient information was obtained by using finite differencing:

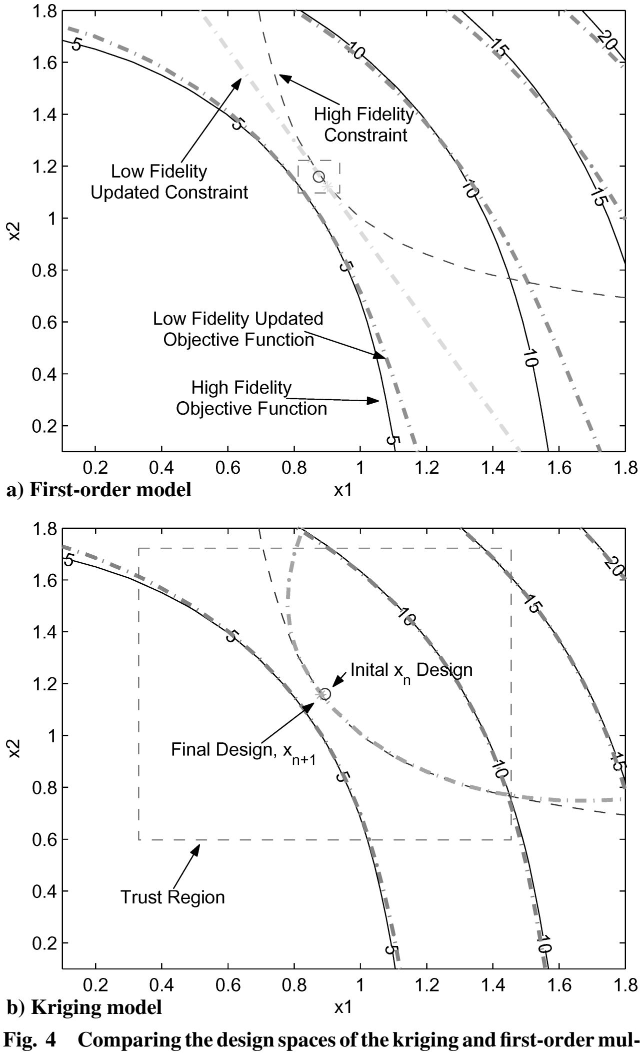
$$f_{\text{high}} = 4x_1^2 + x_2^3 + x_1 x_2 \tag{21}$$

$$g_{\text{high}} = 1/x_1 + 1/x_2 - 2 \tag{22}$$

$$f_{\text{low}} = 4(x_1 + 0.1)^2 + (x_2 - 0.1)^3 + x_1 x_2 + 0.1 \tag{23}$$

$$g_{\text{low}} = 1/x_1 + 1/(x_2 + 0.1) - 2 - 0.001 \tag{24}$$

Each of the methods used converged to the same solution, the optima of the high-fidelity problem, and all used the same convergence criteria of $\epsilon_f = \epsilon_x = 0.0001$. The contour plots at an intermediate iteration of the optimization are given in Fig. 4 for the multiplicative first-order and kriging methods. The kriging model better approximates the objective and constraints over a larger range. Table 1 shows the results of the variable-fidelity optimization using combinations of first-order, second-order BFGS, second-order SR1, second-order full Hessian updates, and kriging models for the various methods. Also included for each method is the warm-started kriging approach. For these trials, the corresponding method was used for the first two iterations. Then a kriging scaling model was used until convergence. The results all compare the the number of high- and low-fidelity function calls, total number of iterations, and final trust region size Δ. The last column shows the savings, in percent of high-fidelity function calls, of using the warm-started kriging method. The SQP optimization results are also given at the end of the table.



a) First-order model

b) Kriging model

Fig. 4 Comparing the design spaces of the kriging and first-order multiplicative scaling method after the same number of iterations.

Table 1 Variable-fidelity optimization results for the analytic test problem

					Warm-started kriging (>2 iter.)				
Method	High	Low	Iter.	Final Δ	High	Low	Iter.	Final Δ	Kriging savings
Multiplicative scaling									
1st order	35	187	14	0.000244	25	152	8	0.015625	29%
2nd order, BFGS	18	145	7	0.062500	14	137	5	0.062500	22%
2nd order, SR1	18	147	7	0.187500	14	137	5	0.062500	22%
2nd order, FULL	25	115	4	1.000000	14	99	4	0.062500	44%
Kriging	29	244	12	0.026367	——	——	——	——	——
Additive scaling									
1st order	13	89	4	0.250000	13	112	4	0.25000	0%
2nd order, BFGS	13	94	4	1.000000	13	112	4	0.25000	0%
2nd order, SR1	16	110	5	1.000000	13	112	4	0.25000	19%
2nd order, FULL	25	88	4	1.000000	17	97	5	0.25000	32%
Kriging	13	92	4	1.000000	——	——	——	——	——
Adaptive hybrid scaling									
1st order	20	159	9	0.000977	17	190	6	0.250000	15%
2nd order, BFGS	13	96	4	1.000000	10	69	3	0.250000	23%
2nd order, SR1	13	93	4	1.000000	13	93	5	0.250000	0%
2nd order, FULL	25	97	4	1.000000	20	129	6	0.015625	20%
Kriging	22	238	7	1.000000	——	——	——	——	——
SQP	44	——	8	——	——	——	——	——	——

The plots in Fig. 4 depict some of the differences between using a Taylor-series approach and kriging-based approach. The most noticeable difference between these plots is that the kriging model does a much better job of approximating the objective function and the constraint boundary; this would imply a faster convergence rate. Another major difference is that the trust region size is much larger for the kriging method. This larger trust region is a better qualitative indication that the kriging method is doing a superior job scaling the low-fidelity model to match the high-fidelity model. This trend is not observed in Table 1 but is observed in the next problem. However, in most cases for this problem the trust region size has only been decreased once using the kriging methods. This reduction generally occurred during the first use of the kriging models, when only a small sample set had accumulated.

Many observations can be made from comparing the results in Table 1. A significant trend observed is that the warm-started kriging method improved the performance of most of the scaling methods. For this problem, the additive method worked much better than the multiplicative method; however, the hybrid scaling method did very well adapting to to match the additive method. In this problem the BFGS and SR1 second-order methods generally performed better than the first-order methods. And there was no significant difference between the BFGS and SR1 methods. The second-order method using the full Hessian calculation had mixed results in terms of having fewer high-fidelity function calls than the first-order model. However, with an increase in the number of design variables the second-order information would become much more expensive to compute. The kriging method used alone did not show any significant savings over the second-order methods. This is because the kriging model does not include any second-order information. Future work could be done to try to add second-order information into the kriging model to improve its performance when used alone or in the warm-started mode.

B. Energy-Efficient-Transport High-Lift Airfoil Design

For about three decades, starting in the mid-1970s, NASA conducted research to improve the efficiency of jet transport aircraft. Part of this research effort included the energy-efficient-transport program, which developed supercritical airfoils with larger section thickness-to-chord ratios, higher aspect ratios, higher cruise lift coefficients, and less swept wings. Because these wings had higher lift at cruise, they could be smaller and more fuel efficient. With the reduced wing area, these new wings needed a high-lift flap system to ensure that takeoff and landing requirements could be met.

The problem solved here is: given a high-lift airfoil, find the optimal placement of its slat, vane, and flap to provide a maximum lift coefficient for takeoff or landing configurations. The problem was developed from the experimental and numerical work done at NASA Langley Research Center.[45–47] The problem consists of nine design variables that control the horizonal, vertical, and rotational orientation of the slat vane and flap, relative to their cruise configuration. The rotation is measured positive counterclockwise about each control surface's leading edge. Figure 5 shows the layout of the supercritical airfoil and the control surface movements. The only constraints placed on the system were that the gaps between control surfaces must be positive in value for gridding purposes.

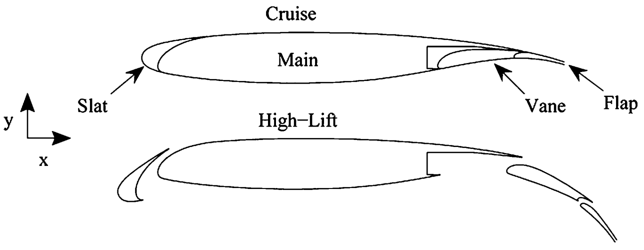


Fig. 5 Cruise and high-lift configurations for EET airfoil.

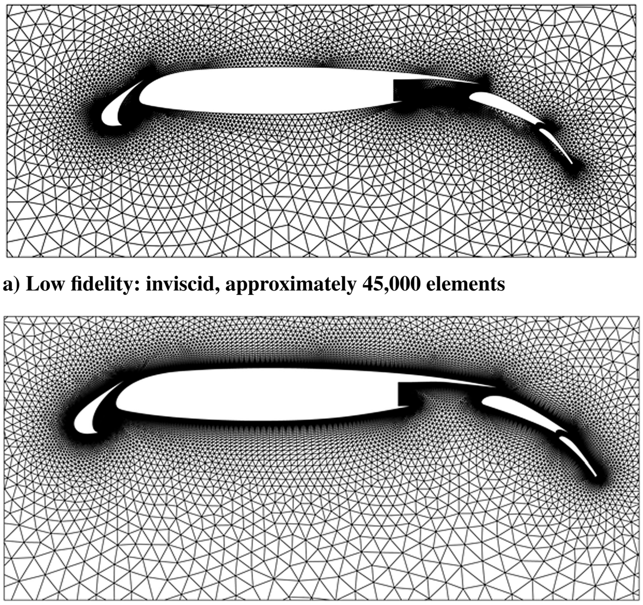
The flow conditions for the problem consisted of a Reynolds number of 9×10^6, Mach number of 0.25, and an angle of attack 3 deg. The flow was solved by using the inviscid Euler's equations for the low-fidelity model. The grid consisting of about 45,000 elements, as seen in Fig. 6a, was extended to 30 times the chord length in each direction. For the high-fidelity model, a full Navier–Stokes solution was used; the grid consisted of about 100,000 elements as seen in Fig. 6b. The computational fluid dynamics (CFD) runs took approximately 10 min and 2.5 h for the low- and high-fidelity models, respectively. To further reduce the computation expense of the low-fidelity model, a kriging surrogate was created reducing the expense to approximately a second. Also, a surrogate was created and used for the high-fidelity model for the purpose of reducing the cost to a tractable level for debugging and testing, which follows from the work done by Alexandrov et al.[8]

Both fidelity models were solved using a computational-fluid-dynamics package developed at NASA Langley Research Center called FUN2D.[48,49] This package uses fully unstructured mesh, which was generated using the advancing-front local-reconnection method described by Marcum.[50,51]

The results of the optimization trials are presented in Table 2. Based on the results from the analytic case, only the adaptive hybrid method was used on this problem. With the same convergence criteria as in the two-dimensional analytic problem, all of the trials converged to the same solution.

Table 2 Variable-fidelity optimization results for the AEET problem

	Warm-started kriging (>3 iter.)								
Adaptive hybrid scaling	High	Low	Iter.	Final Δ	High	Low	Iter.	Final Δ	Savings
1st order	35	279	34	0.000220	14	352	13	0.004944	60%
2nd order, BFGS	21	295	20	0.000103	12	295	11	0.002930	43%
2nd order, SR1	15	298	14	0.000412	12	378	11	0.002930	20%
Kriging	15	462	14	0.187500	——	——	——	——	——
SQP	61	——	20	——	——	——	——	——	——



a) Low fidelity: inviscid, approximately 45,000 elements

b) High fidelity: viscid, approximately 100,000 elements

Fig. 6 Close-up view of the high- and low-fidelity unstructured CFD grids used in the AEET problem.

The results from this larger, more complicated problem further support the claim that the warm-started kriging-based approach can save a significant number of function calls. Each variable-fidelity method also used fewer high-fidelity function calls than did the standard SQP optimizer. For this problem, the second-order and kriging methods outperformed the first-order methods, and SR1 was the best of the second-order methods. One interesting side effect observed when using the kriging-based methods was an increase in low-fidelity function calls. A possible reason for this might be that the scaled kriging model becomes very flat near the optima.

IX. Summary

In this work, many different scaling techniques were presented for use in variable-fidelity optimization including approximate second-order methods, kriging-based scaling methods, and an adaptive hybrid method. The focus was on kriging-based techniques and the adaptive hybrid methods, which were given here for the first time. The methods were presented theoretically and then applied to two demonstration problems: a two-dimensional analytic problem and a high-lift airfoil design problem.

The results of both demonstration problems indicated that the warm-started kriging-based method can save substantial computational cost by lowering the number of high-fidelity function calls required for optimization. In the analytic problem the warm-started method used 19% fewer high-fidelity function calls on average for the various scaling methods. For the high-lift example the warm-started method reduced the number of expensive function calls by 60% using first-order information and by an average of over 30% for the different second order scaling methods.

It was demonstrated that the adaptive hybrid method has the ability to adjust to an appropriate weighting between the additive and multiplicative methods so as to eliminate the need to determine a priori which method should be used. The hybrid method did not improve performance over using either the multiplicative or additive scaling methods; it just alleviated the need for the designer to select the appropriate method. The adaptive hybrid method, therefore, can be used to save computational resources prior to performing the optimization.

A few drawbacks were identified with the proposed kriging models. They tended to increase the number of low-fidelity function calls, which would be an issue if the low-fidelity model required significant computational time. More work is needed to identify why this occurs and possible corrections. Also, the kriging models might be expensive to build and update after each iteration, depending on the number of design variables and sample points used; though, this cost can be insignificant, depending on the relative cost of the high-fidelity model. A possible scaling limitation is that the adaptive hybrid method requires more memory to store both the additive and multiplicative models. With sufficient memory, the extra processing times for handling both models is offset by the savings in high-fidelity function calls.

Acknowledgments

This research effort was supported in part by the following grants and contracts: U.S. Air Force Research Laboratory/Defense Advanced Research Project Agency/Anteon Corporation Contract F33615-98-D-3210, Office of Naval Research Grant N00014-02-1-0786, and National Science Foundation Grant DMI-0114975. The authors also thank Victor Pérez, Mike Eldred, and Jay Martin for their insightful discussions.

References

[1] Braibant, V., and Fleury, C., "An Approximation-Concepts Approach to Shape Optimal Design," *Computer Methods in Applied Mechanics and Engineering*, Vol. 53, Nov. 1985, pp. 119–148.

[2] Hajela, P., "Geometric Programming Strategies in Large-Scale Structural Synthesis," *AIAA Journal*, Vol. 24, No. 7, 1986, pp. 1173–1178.

[3] Alexandrov, N., "Robustness Properties of a Trust Region Framework for Managing Approximations in Engineering Optimization," *Proceedings of the 6th AIAA/NASA/USAF Multidisciplinary Analysis and Optimization Symposium*, AIAA, Reston, VA, 1996, pp. 1056–1059.

[4] Dennis, J., and Torczon, V., "Managing Approximation Models in Optimization," *Multidisciplinary Design Optimization: State of the Art*, edited by N. Alexandrov and M. Y. Hussaini, Society for Industrial and Applied Mathematics, Philadelphia, 1996.

[5] Haftka, R. T., "Combining Global and Local Approximations," *AIAA Journal*, Vol. 29, No. 9, 1991, pp. 1523–1525.

[6] Chang, K. J., Haftka, R. T., Giles, G. L., and Kao, P.-J., "Sensitivity-Based Scaling for Approximating Structural Response," *Journal of Aircraft*, Vol. 30, No. 2, 1993, pp. 283–288.

[7] Alexandrov, N. M., Nielsen, E. J., Lewis, R. M., and Anderson, W. K., "First-Order Model Management with Variable-Fidelity Physics Applied to Multi-Element Airfoil Optimization," AIAA Paper 2000-4886, Sept. 2000.

[8] Alexandrov, N. M., Lewis, R. M., Gumbert, C. R., Green, L. L., and Newman, P. A., "Optimization with Variable-Fidelity models Applied to Wing Design," AIAA Paper 2000-0841, Jan. 2000.

[9] Alexandrov, N. M., and Lewis, R. M., "First-Order Approximation and Model Management in Optimization," *Large-Scale PDE-Constrained Optimization*, Lecture Notes in Computational Science and Engineering, Vol. 30, edited by L. T. Biegler, O. Ghattas, M. Heinkenschloss, and B. v. Bloemen Waanders, Springer-Verlag, 2003, pp. 65–81.

[10] Conn, A. R., Gould, N. I. M., and Toint, P. L., *Trust-Region Methods*, Society for Industrial and Applied Mathematics, Philadelphia, 2000.

[11] Giunta, A. A., and Eldred, M. S., "Implementation of a Trust Region Model Management Strategy in the DAKOTA Optimization Toolkit," AIAA Paper 2000-4935, Sept. 2000.

[12] Booker, A. J., Dennis, J. E., Frank, P. D., Serani, D. B., Torczon, V., and Trosset, M. W., "A Rigorous Framework for Optimization of Expensive Function by Surrogates," Center for Research on Parallel Computation, CRPC-TR98739-S, Houston, TX, April 1998.

[13] Rodríguez, J. F., and Renaud, J. E., "Convergence of Trust Region Augmented Lagrangian Methods Using Variable Fidelity Approximation Data," *Structural Optimization*, Vol. 15, 1998, pp. 141–156.

[14] Rodríguez, J. F., Renaud, J. E., and Watson, L. T., "Trust Region Augmented Lagrangian Methods for Sequential Response Surface Approximation and Optimization," *Journal of Mechanical Design*, Vol. 120, No. 1, 1998, pp. 58–66.

[15] Rodríguez, J. F., Renaud, J. E., Wujek, B. A., and Tappeta, R. V., "Trust Region Model Management in Multidisciplinary Design Optimization," *Computational and Applied Mathematics*, Vol. 124, Nos. 1–2, 2000, pp. 139–154.

[16] Sobieszczanski-Sobieski, J., "Sensitivity of Complex, Internally Coupled Systems," *AIAA Journal*, Vol. 28, No. 1, 1990, pp. 153–160.

[17] Rodriguez, J. F., Pérez, V. M., Padmanabhan, D., and Renaud, J. E., "Sequential Approximate Optimization Using Variable Fidelity Response Surface Approximations," *Structural and Multidisciplinary Optimization*, Vol. 22, No. 1, 2001, pp. 24–34.

[18] Pérez, V. M., Renaud, J. E., and Watson, L. T., "Interior Point Sequential Approximate Optimization Methodology," AIAA Paper 2002-5505, Sept. 2002.

[19] Pérez, V. M., Renaud, J. E., and Watson, L. T., "Reduced Sampling for Construction of Quadratic Response Surface Approximations Using Adaptive Experimental Design," AIAA Paper 2002-1587, April 2002.

[20] Leary, S. J., Bhaskar, A., and Keane, A. J., "A Constraint Mapping Approach to the Structural Optimization of an Expensive Model Using Surrogates," *Optimization and Engineering*, Vol. 2, 2001, pp. 385–398.

[21] Alexandrov, N. M., Dennis, J. E., Lewis, R. M., and Torczon, V., "A Trust Region Framework for Managing the Use of Approximation Models in Optimization," ICASE, Rept. 97-50, NASA CR-201745, Oct. 2001.

[22] Conn, A. R., Gould, N. I. M., and Toint, P. L., "Global Convergence of a Class of Trust Region Algorithms for Optimization with Simple Bounds," *SIAM Journal of Numerical Analysis*, Vol. 25, No. 2, 1988, pp. 433–464.

[23] Lewis, R. M., and Nash, S. G., "A Mulitgrid Approach to the Optimization of Systems Governed by Differential Equations," AIAA Paper 2000-4890, Sept. 2000.

[24] Gano, S. E., Pérez, V. M., and Renaud, J. E., "Multi-Objective Variable-Fidelity Optimization of a Morphing Unmanned Aerial Vehicle," *Proceedings of the 45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference* [CD-ROM], AIAA, Reston, VA, 2004.

[25] Eldred, M. S., Giunta, A. A., Collis, S. S., Alexandrov, N. A., and Lewis, R. M., "Second-Order Corrections for Surrogate-Based Optimization with Model Hierarchies," *Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference* [CD-ROM], AIAA, Reston, VA, 2004.

[26] Broyden, C. G., "The Convergence of a Class of Double-Rank Minimization Algorithms," *Journal of the Institute of Mathematics and Its Application*, Vol. 6, 1970, pp. 76–90.

[27] Fletcher, R., "A New Approach to Variable Metric Algorithms," *Computer Journal*, Vol. 13, No. 3, 1970, pp. 317–322.

[28] Goldfarb, D., "A Family of Variable Metric Updates Derived by Variational Means," *Mathematics of Computing*, Vol. 24, 1970, pp. 23–26.

[29] Shanno, D. F., "Conditioning of Quasi-Newton Methods for Function Minimization," *Mathematics of Computing*, Vol. 24, No. 111, 1970, pp. 647–656.

[30] Nocedal, J., and Wright, S. J., *Numerical Optimization*, Springer Series in Operations Reseach, Springer-Verlag, New York, 1999.

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

[32] Cressie, N. A. C., *Statistics for Spatial Data*, Wiley, New York, 1993.

[33] Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P., "Design and Analysis of Computer Experiments," *Statistical Science*, Vol. 4, No. 4, 1989, pp. 409–423.

[34] Chung, H.-S., and Alonso, J. J., "Using Gradients to Construct Cokriging Approximation Models for High-Dimensional Design Optimization Problems," AIAA Paper 2002-0317, Jan. 2002.

[35] Martin, J. D., and Simpson, T. W., "A Study on the Use of Kriging Models to Approximate Deterministic Computer Models," *Proceedings of ASME 2003 Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers, New York, 2003.

[36] Martin, J. D., and Simpson, T. W., "On the Use of Kriging Models to Approximate Deterministic Computer Models," *Proceedings of the ASME 2004 International Design Engineering Technical Conference*, American Society of Mechanical Engineers, New York, 2004.

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

[38] Giunta, A. A., and Watson, L. T., "A Comparison of Approximation Modeling Techniques: Polynomial Versus Interpolating Models," AIAA Paper 98-4758, Sept. 1998.

[39] Lophaven, S. N., Nielsen, H. B., and Søndergaard, J., "DACE a Matlab Kriging Toolbox," Technical Univ. of Denmark, Tech. Rept. IMM-REP-2002-12, Ver. 2.0, Kongens Lyngby, Aug. 2002.

[40] Lophaven, S. N., Nielsen, H. B., and Søndergaard, J., "Aspects of the Matlab Toolbox DACE," Technical Univ. of Denmark, Tech. Rept. IMM-REP-2002-13, Kongens Lyngby, Aug. 2002.

[41] Denison, D. G. T., Holmes, C. C., Mallick, B. K., and Smith, A. F. M., *Bayesian Methods for Nonlinear Classication and Regression*, Wiley, New York, 2002.

[42] Stein, M. L., *Interpolation of Spatial Data: Some Theory for Kriging*, Springer-Verlag, New York, 1999.

[43] 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, 1998, pp. 455–492.

[44] *Optimization Toolbox for Use with MATLAB: User's Guide Version 3*, The MathWorks, Inc., Natick, MA, 2004.

[45] Morgan, H. L. J., "Experimental Test Results of Energy Efficient Transport (EET) High-Lift Airfoil in Langley Low-Turbulance Pressure Tunnel," NASA TM-2002-211780, Dec. 2002.

[46] Lin, J. C., and Dominik, C. J., "Parametric Investigation of a High Lift Airfoil at High Reynolds Numbers," *Journal of Aircraft*, Vol. 34, No. 4, 1997, pp. 485–491.

[47] Gatlin, G. M., and McGhee, R. J., "Study of Semi-Span Model Testing Techniques," *Proceedings of the 14th Applied Aerodynamics Conference*, AIAA, Reston, VA, 1996.

[48] Anderson, W. K., and Bonhaus, D. L., "An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids," *Computers and Fluids*, Vol. 23, No. 1, 1994, pp. 1–21.

[49] Anderson, W. K., Rausch, R. D., and Bonhaus, D. L., "Implicit/Multigrid Algorithms for Incompressible Turbulent Flows on Unstructured Grids," *Journal of Computational Physics*, Vol. 128, No. 2, 1996, pp. 391–408.

[50] Marcum, D. L., "Adaptive Unstructured Grid Generation for Viscous Flow Applications," *AIAA Journal*, Vol. 34, No. 11, 1996, p. 2440.

[51] Marcum, D. L., "Advancing-Front/Local-Reconnection (AFLR) Unstructured Grid Generation," *Computational Fluid Dynamics Review*, edited by M. Hafez and K. Oshima, Vol. 1, World Scientific, 1998.

A. Messac
Associate Editor