10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 30 August - 1 September 2004, Albany, New York

Variable Fidelity Optimization Using a Kriging Based Scaling Function

Shawn E. Gano[*] and John E. Renaud [†]

University of Notre Dame, Notre Dame, Indiana, 46556-5637, U.S.A.

Brian Sanders[‡]

Air Force Research Laboratory, WPAFB, Dayton, Ohio, 45433-7531, U.S.A.

Solving design problems that rely on very complex and computationally expensive calculations using standard optimization methods may 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; recently these have been extended to second order. In this work 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 studied. It combines the additive and multiplicative approaches so that the designer doesn't have to determine which is better before optimization. The different methods are theoretically described and then compared using two demonstration problems. The first problem is analytic, while the other is a design of a supercritical high-lift airfoil. The results show that the warm-started kriging based scaling methods have the potential to improve computational expense by lowering the number of high fidelity function calls required for convergence. The results also indicate the hybrid method is effective.

Nomenclature

β	Multiplicative scaling function
Δ	Trust region size
ϵ_f	Objective function convergence tolerance
ϵ_x	Design variable convergence tolerance
γ	Additive scaling function
$\mathbf{l}$	Design space lower bounds
$\mathbf{R}$	Correlation matrix
$\mathbf{r}$	Correlation vector
$\mathbf{u}$	Design space upper bounds
$\mathbf{x}$	Design vector
$\mathcal{W}$	Hybrid weighting value
∇	Gradient operator
∇^2	Hessian operator
ρ	Trust region ratio
σ^2	Variance
$\mathbf{g}$	Inequality constraint

[*] 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.

[‡] AFRL/VASA, Brian.Sanders@wpafb.af.mil, Associate Fellow AIAA.

$\mathbf{h}$	Equality constraint
Θ	Damping factor in damped BFGS method
θ	Correlation parameters
f	Objective function
H	Hessian matrix
I	Identity matrix
MSE	Mean squared error
n_s	Number of sample sites used in kriging model
n_v	Number of variables used in kriging model
s_n	Change in the design vector from the past iteration
y_n	Change in the gradient of the objective function from the past iteration
B	Global kriging trend function
R	Correlation function
y	Function modeled using kriging
Z	Stochastic process model in kriging

Subscripts

$high$	High fidelity model
i, j, k	Free indices for elements in a vector or matrix
low	Low fidelity model
n	Current iteration number
$scaled$	Scaled low fidelity value
u	Denotes an unscaled constraint

Superscripts

$\hat{}$	Predicted value or function
$\tilde{}$	Approximate function
^{T}	Transpose operator

I. Introduction

Variable fidelity and other model management methods have been developed to solve optimization problems that involve simulations with extreme computational expense. However, these methods can be used to optimize any problem where various fidelity models exist or can be built. 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 time needed 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 high predictive full-physics based models used for the final optimization analysis. The lower fidelity models are typically much cheaper to evaluate, but designs produced 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 provably 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 a trust region management scheme is used.

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 the fidelity models before optimization.

Early works that used variable fidelity models for optimization were largely based on heuristics, and the process 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 methods were developed which were proven to converge to the high fidelity solution. Some of these methods used gradient information[3] and others did not.[4] Chang and Haftka *et al.*[5] used a multiplicative scaling approach to update the value of lower fidelity models to match the higher fidelity models. Alexandrov[6–8] built on this work by creating an approximation management framework (AMF) that incorporated this first order scaling into a provably convergent methodology for 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.[9] A pattern search based framework was also developed by Booker *et al.*[10]

Response surface based methods that use variable fidelity data have been extensively studied by Rodríguez *et al.*[11–13] In this work, coupled multidisciplinary design optimization problems were studied. Response surfaces were constructed using the fully coupled solutions and decoupled approximate solutions generated from the use of the global sensitivity equations (GSE) developed by Sobieszczanski-Sobieski.[14] The response surfaces were sequentially optimized using the Augmented Lagrangian method, and a trust region model management framework was used to prove convergence.[15]

Successive approximate optimization (SAO) algorithms generate response surfaces with a similar motivation as variable fidelity methods: to conduct optimization on a inexpensive surrogate model. SAO methods have been used quite successfully in reducing computational cost, especially when the sampling needed to create the response surface is on the order of the number of design variables.[16,17] Variable fidelity methods can further reduce this cost by using a function similar to a response surface, which is based more on physics — namely 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. The majority of the work here is concerned with how this model is constructed and managed.

The are a few significant differences between the existing variable fidelity methodologies and the one presented in this paper. In the existing methods there is an underlying theme that the lower fidelity model is scaled to match the high fidelity model in a local region only. This paper uses the premise that the lower fidelity model approximates the high fidelity model more 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 and also used here is kriging. Further extensions of variable fidelity optimization include the hybrid approach which combines the additive and multiplicative approaches and a study of the second order methods.

The variable fidelity optimization problem can be written in its most general form by the standard nonlinear programming problem:

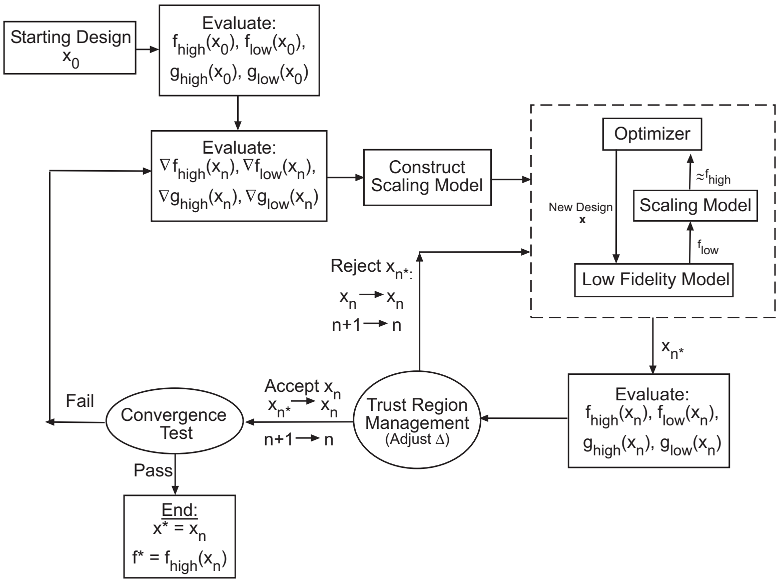
$$\begin{array}{ll} \underset{\mathbf{x}}{\text{minimize}} & f(\mathbf{x}) \\ \text{subject to:} & \mathbf{g}(\mathbf{x}) \le 0 \\ & \mathbf{g}_u(\mathbf{x}) \le 0 \\ & \mathbf{h}(\mathbf{x}) = 0 \\ & \mathbf{h}_u(\mathbf{x}) = 0 \\ & \mathbf{l} \le \mathbf{x} \le \mathbf{u}, \end{array}$$

where f is the objective function, x are the design variables, $\mathbf{g}$ is the vector of inequality constraint, $\mathbf{h}$ is the vector of equality constraint, the subscript u means the constraint is unscaled, and $\mathbf{l}$ and $\mathbf{u}$ 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 may 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 the variable fidelity framework 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 Figure 1 and is based, in part, on work done by Alexandrov.[18] The following process describes the basic steps of the framework:



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
Accept x_n
$x_{n^*} \rightarrow x_n$
$n+1 \rightarrow n$
Convergence Test
Trust Region Management (Adjust Δ)
Evaluate: $f_{high}(x_n)$, $f_{low}(x_n)$, $g_{high}(x_n)$, $g_{low}(x_n)$
Pass
End: $x^* = x_n$ $f^* = f_{high}(x_n)$

Figure 1. Varible fidelity framework flowchart.

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

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, $\mathbf{x}_n$.

Step 3 Construct Scaling Model: A scaling model is constructed to insure 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,[7] three optimizers were compared: augmented Lagrangian method, multilevel algorithms for large-scale constrained optimization (MAESTRO)[6] (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

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

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

Step 6 Trust Region Management: In order to guarantee convergence of the variable fidelity optimization framework, a trust region model management strategy is employed.[19] 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, $\mathbf{x}_n^*$, is found. A trust region ratio, ρ_n, is calculated at this new point:

$$\rho_n = \frac{P(\mathbf{x}_n)_{high} - P(\mathbf{x}_n^*)_{high}}{P(\mathbf{x}_n)_{scaled} - P(\mathbf{x}_n^*)_{scaled}}, \tag{2}$$

where $P()_{high}$ and $P()_{scaled}$ are the l_1 penalty functions for the high and scaled low fidelity models and the point $\mathbf{x}_n$ was the initial point of the optimization. Notice that by definition $P(\mathbf{x}_n)_{scaled} = P(\mathbf{x}_n)_{high}$, 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:[9,15]

$$\Delta_{n+1} = \begin{cases} c_1\Delta_n & : \quad \rho_n \le R_1 \\ \Delta_n & : \quad R_1 < \rho_n < R_2 \\ \Gamma\Delta_n & : \quad R_2 \le \rho_n < R_3 \end{cases}. \tag{3}$$

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 are $R_1 = 0.25$, $R_2 = 0.75$, and $R_3 = 1.25$, while 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 1, 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: The convergence of the entire framework is governed by two inequalities:

$$f_{high}(\mathbf{x}_n) - f_{high}(\mathbf{x}_{n-1}) \quad < \quad \epsilon_f, \tag{4}$$

$$\|\mathbf{x}_n - \mathbf{x}_{n-1}\| \quad < \quad \epsilon_x, \tag{5}$$

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.

III. Scaling Methods

Existing variable fidelity or approximate model management frameworks come in two varieties: multiplicative or additive. Currently, the most common is the multiplicative framework, devised by Alexandrov and Lewis[8] based on Chang's[5] scaling function. The additive method was presented by Lewis and Nash.[20] 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, $f_{high}(\mathbf{x})$ and $f_{low}(\mathbf{x})$, can be matched by multiplying the low fidelity model by an unknown function $\beta(\mathbf{x})$. This is posed mathematically as

$$f_{high}(\mathbf{x}) = \beta(\mathbf{x}) f_{low}(\mathbf{x}). \tag{6}$$

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

$$\beta(\mathbf{x}) = \frac{f_{high}(\mathbf{x})}{f_{low}(\mathbf{x})}. \tag{7}$$

From inspection of Equation 7, it is clear that the function $\beta(\mathbf{x})$ 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.

1. *Multiplicative Scaling Divide-By-Zero Problem*

In this investigation, the multiplicative scaling methodology, as introduced, was found to have a divide-by-zero problem under certain conditions. This issue was not identified by Alexandrov[6,7,21] or Chang.[5] The issue arises when either the constraint or objective values of the low fidelity model approach zero. This causes a problem because the low fidelity value is in the denominator of Equation 12 which defines the multiplicative scaling function. The divide by zero problem can occur as constraints become active or the objective nears zero. To solve this problem, a simple extra offset factor is proposed. This is obtained by adding a constant term, C, to the low fidelity objective and/or constraint values. When this factor is added to low fidelity constraints and/or objective, it prevents the divide-by-zero problem. A strategy for choosing this factor and investigating its effect on the sensitivity of the method is still to be completed. The scaling function β does automatically adjust for this added scalar and, therefore, doesn't introduce any further complications as long as the offset value is *reasonable* in size.

B. Additive Scaling

A given set of high and low fidelity models, $f_{high}(\mathbf{x})$ and $f_{low}(\mathbf{x})$, can also be matched by adding the low fidelity model to an unknown function $\gamma(\mathbf{x})$. This is expressed mathematically as

$$f_{high}(\mathbf{x}) = f_{low}(\mathbf{x}) + \gamma(\mathbf{x}). \tag{8}$$

The additive scaling function can be solved for by subtracting the low fidelity function from both sides:

$$\gamma(\mathbf{x}) = f_{high}(\mathbf{x}) - f_{low}(\mathbf{x}). \tag{9}$$

From Equation 9, it is clear that the function $\gamma(\mathbf{x})$ 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 Equations (8) and (9).

IV. First Order Scaling Models

For both of the scaling methods described in the previous section, it is possible to construct various scaling models. In this section the first order method is presented; a higher order method is presented in the next section, followed by the new kriging based approach.

A. First Order Multiplicative Scaling

The first order multiplicative approximation model is found using Chang's[5] scaling function $\beta(\mathbf{x})$. At a given design point, for example the current design, this function is defined as

$$\beta(\mathbf{x}_n) = \frac{f_{high}(\mathbf{x}_n)}{f_{low}(\mathbf{x}_n)}. \tag{10}$$

This scaling factor at any other point can be approximated using a Taylor series to first order:

$$\tilde{\beta}(\mathbf{x}) = \beta(\mathbf{x}_n) + \nabla\beta(\mathbf{x}_n)^T(\mathbf{x} - \mathbf{x}_n). \tag{11}$$

To evaluate this, the gradient information is needed and can be obtained by differentiating Equation 10, resulting in

$$\nabla\beta = \frac{\frac{\partial f_{high}}{\partial x_i} f_{low} - \frac{\partial f_{low}}{\partial x_i} f_{high}}{f_{low}^2}. \tag{12}$$

Therefore, a first order update on the low fidelity model is

$$f_{high} \approx \tilde{\beta}(\mathbf{x}) f_{low}. \tag{13}$$

This model insures that at the initial design point, the updated low fidelity model matches the function and the gradient of the high fidelity model. The identical process is done in order to scale each constraint.

B. First Order Additive Scaling

The first order additive scaling method is similar to the first order multiplicative scaling method because it tries to approximate the high fidelity model by applying a correction to the lower fidelity model. The additive method was used by Lewis and Nash[20] to solve multigrid problems but can be used more generally.

At a given design point, the additive scaling function has the value

$$\gamma(\mathbf{x}_n) = f_{high}(\mathbf{x}_n) - f_{low}(\mathbf{x}_n). \tag{14}$$

This additive scaling factor at any other point can be approximated using a Taylor series to first order:

$$\tilde{\gamma}(\mathbf{x}) = \gamma(\mathbf{x}_n) + \nabla\gamma(\mathbf{x}_n)^T(\mathbf{x} - \mathbf{x}_n). \tag{15}$$

Evaluating this requires gradient information which can be obtained by differentiating Equation 14. This gives

$$\nabla\gamma(\mathbf{x}_n) = \nabla f_{high}(\mathbf{x}_n) - \nabla f_{low}(\mathbf{x}_n). \tag{16}$$

Therefore, a first order update on the low fidelity model is

$$f_{high}(\mathbf{x}) \approx f_{low}(\mathbf{x}) + \tilde{\gamma}(\mathbf{x}). \tag{17}$$

This model insures 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. Nearby points should also approximate the high fidelity response well.

V. Second Order Scaling Models

Using the same idea as in the first order method, an approximation scaling function can be derived to match second order information. This approach hasn't been used until recently by Gano *et al.*[22] and Eldred *et al.*[23] This 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

$$\tilde{\beta}(\mathbf{x}) = \beta(\mathbf{x}_n) + \Delta\mathbf{x}^T \nabla\beta(\mathbf{x}_n) + \frac{1}{2}\Delta\mathbf{x}^T \nabla^2\beta(\mathbf{x}_n)\Delta\mathbf{x}. \tag{18}$$

Using the same gradient result as in Section IV.A, the only remaining term needed is the Hessian of β; this can be found by differentiating again, which simplifies to:

$$\nabla^2\beta(\mathbf{x}_n) = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,n} \\ h_{2,1} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ h_{n,1} & \cdots & \cdots & h_{n,n} \end{bmatrix}, \tag{19}$$

where

$$\begin{aligned} h_{i,j} = \frac{1}{f_{low}^3}\Bigg(2f_{high}\frac{\partial f_{low}}{\partial x_i}\frac{\partial f_{low}}{\partial x_j} + f_{low}^2\frac{\partial^2 f_{high}}{\partial x_i \partial x_j} - f_{low}\Bigg(\frac{\partial f_{low}}{\partial x_j}\frac{\partial f_{high}}{\partial x_i} + \\ \frac{\partial f_{high}}{\partial x_j}\frac{\partial f_{low}}{\partial x_i} + f_{high}\frac{\partial^2 f_{low}}{\partial x_i \partial x_j}\Bigg)\Bigg), \end{aligned} \tag{20}$$

where all the functions and partial derivatives are evaluated at the point $\mathbf{x}_n$ 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

$$\tilde{\gamma}(\mathbf{x}) = \gamma(\mathbf{x}_n) + \Delta\mathbf{x}^T \nabla\gamma(\mathbf{x}_n) + \frac{1}{2}\Delta\mathbf{x}^T \nabla^2\gamma(\mathbf{x}_n)\Delta\mathbf{x}. \tag{21}$$

The first order information was found in Section IV.B. Again, the only remaining information needed is the Hessian of γ; this can be found by taking the gradient of the gradient of γ:

$$\nabla^2\gamma(\mathbf{x}_n) = \nabla^2 f_{high}(\mathbf{x}_n) - \nabla^2 f_{low}(\mathbf{x}_n). \tag{22}$$

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 was readily available.

A. Approximate Second Order Scaling

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. The second order information can, therefore, be obtained at no additional cost, in terms of function calls, compared to the first order scaling methods from Sections IV.A and IV.B. The two most prevalent methods used are Broyden-Fletcher-Goldfarb-Shanno (BFGS) and symmetric-rank-1 (SR1).

1. *Initial Scaling of the Hessian*

Typically, the initial guess for the Hessian information, H, for the objective and each constraint function is chosen to be the identity matrix. The scaling of this initial guess may be undesirable, so a scaling method has been implemented; it is discussed by Nocedal and Wright.[24] The scaling is derived from the approximate values of the eigenvalues of the Hessian. This scaling factor is given as

$$H_o = \frac{y_n^T s_n}{y_n^T y_n} I, \tag{23}$$

where

$$\begin{aligned} y_n &= \nabla f_n - \nabla f_{n-1} && (24) \\ s_n &= x_n - x_{n+1}. && (25) \end{aligned}$$

In the above equations, f is the function to which the approximate second order information is desired.

2. *Damped BFGS*

Broyden,[25] Fletcher,[26] Goldfarb,[27] and Shanno[28] developed a rank-2 update for the second order information; this method is referred to as BFGS. The BFGS method can become unstable at certain points, so a more robust implementation is the damped BFGS.[24] This method is defined by the following:

$$r_n = \Theta_n y_n + (1 - \Theta_n) H_n s_n, \tag{26}$$

where the scalar Θ_n is defined as

$$\Theta_n = \begin{cases} 1 & : \; s_n^T y_n \geq 0.2 s_n^T H_n s_n \\ \frac{0.8 s_n^T H_n s_n}{s_n^T H_n s_n - s_n^T y_n} & : \; s_n^T y_n < 0.2 s_n^T H_n s_n \end{cases}. \tag{27}$$

The Hessian update H_n is

$$H_{n+1} = H_n - \frac{H_n s_n s_n^T H_n}{s_n^T H_n s_n} + \frac{r_n r_n^T}{s_n^T r_n}. \tag{28}$$

This update also has an important feature such that if it is applied to a positive definite matrix, the update will remain positive definite. This is especially useful when using approximate line searches.

3. *Symmetric-Rank-1*

The symmetric-rank-1 (SR1) method does not guarantee that the updated matrix maintains positive definiteness, unlike the BFGS update. This property could be beneficial to the variable fidelity framework because it can capture the true nature of the second order information without relying on the assumption that the design space is positive definite. A downside is that this method is only a rank one update. The SR1 update is

$$H_{n+1} = H_n + \frac{(y_n - H_n s_n)(y_n - H_n s_n)^T}{(y_n - H_n s_n)^T s_n}. \tag{29}$$

VI. Adaptive Hybrid Scaling - Combining Additive and Multiplicative Methods

Up to this point, we 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

$$f_{high}(\mathbf{x}) = \mathcal{W} f_{low}(\mathbf{x})\beta(\mathbf{x}) + (1 - \mathcal{W})\left(f_{low}(\mathbf{x}) + \gamma(\mathbf{x})\right). \tag{30}$$

To determine the value of $\mathcal{W}$, a further condition must be enforced. Concurrent work being done by Eldred *et al.*[23] proposes to use 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

$$\mathcal{W} = \frac{f_{high}(\mathbf{x}_{pp}) - (f_{low}(\mathbf{x}_n) + \gamma(\mathbf{x}_n))}{f_{low}(\mathbf{x}_n)\beta(\mathbf{x}_n) - (f_{low}(\mathbf{x}_n) + \gamma(\mathbf{x}_n))}. \tag{31}$$

In Equation 31 the current additive and multiplicative scaling functions are used along with any previous point, $\mathbf{x}_{pp}$, 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 $\beta(\mathbf{x})$ and $\gamma(\mathbf{x})$ 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.

VII. 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. Therefore, a global scaling function may be better at approximating the high fidelity response. This is the motivation behind the use of an iteratively updated kriging model to estimate the scaling between the different fidelity models. This approach allows the use of all information calculated throughout the course of the optimization, even when the trust region test fails.

The kriging model can be constructed to generate any scaling function: for example, the additive or multiplicative methods already discussed. These models are generated using all the design points where both the high and low fidelity models were evaluated throughout the optimization process. Optimization of the updated or scaled low fidelity model results in a new design point $\mathbf{x}_n^*$. At this design, both the objective and constraints are evaluated using the high fidelity models. Once this information is obtained, the trust region ratio (Equation 2) is calculated using the current scaling functions. If the design is rejected, the exact value of the scaling function is found using available information and is saved along with all past information. If the design is accepted, the gradient information is evaluated for both fidelity models. Then, based on how the gradients are calculated, the exact scaling function information is saved. If the gradients are found via finite differences, then the exact scaling function is stored at each perturbed point. If the gradient information was obtained directly, perhaps by an adjoint approach, then the exact scaling function and its derivative is saved. Once all available information is stored for the Kriging model, the updated scaling function is constructed and used in the next iteration of optimization.

The kriging model gives exact responses at sample points, as it is an interpolating function. This insures that at least zero order matching is obtained. With the inclusion of gradient information, first order matching is achieved. First order matching combined with the trust region model management strategy provides for a provably convergent framework.[18]

Building and rebuilding the kriging models takes extra time and memory storage. This added computational time and resource expenditure may be negligent 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 unseing 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. 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.*[29]

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.[30] Martin and Simpson conducted a study on using kriging models to approximate deterministic computer models and discussed the applicability of various kriging variants.[31,32]

Kriging begins by modeling an unknown function,[33,34] y, with the form

$$y(\mathbf{x}) = B + Z(\mathbf{x}). \tag{32}$$

The term B is an unknown constant or global trend function, typically linear or quadratic, which is estimated based on the known n_s, number sample points, and $Z(x)$ is the modeling of a stochastic process which has a mean of zero, a variance σ^2, and a co-variance that is not zero. The covariance of $Z(x)$ is given by

$$Cov[Z(x^i), Z(x^j)] = \sigma^2 \mathbf{R}[R(\mathbf{x}^i, \mathbf{x}^j)], \tag{33}$$

where is $\mathbf{R}$ is the correlation matrix, R is the correlation function which is selected by the user, and i and j run from 1 to n_s. It is important to notice that $\mathbf{R}$ is symmetric and has unit values along the diagonal.

The selection of the correlation function is chosen by the user when generating the kriging model. In the statistical and engineering literature[35,36] the Gaussian function is by far the most popular and is also used in this work. It is defined as

$$R(x^i, x^j) = e^{-\sum_{k=1}^{n_v} \theta_k |x_k^i - x_k^j|^2}, \tag{34}$$

where θ_k is the vector of unknown correlation parameters which is of length n_v, the number of design variables. Also, x_k^i and x_k^j are the kth elements of the sample points $\mathbf{x}^i$ and $\mathbf{x}^j$.

The kriging model estimates values of $y(\mathbf{x})$ at untried values of $\mathbf{x}$ based on a given set of expected values or sample points. The notation in this paper that distinguishes between the true value and an estimated value uses a hat to denote the estimated values, such as y is the true response value and $\hat{y}$ is the estimated value. The mean squared error is defined as the square of the expected value of the difference between the real response and the approximated one at any point. Mathematically this is stated as

$$MSE = E(y(\mathbf{x}) - \hat{y}(\mathbf{x}))^2. \tag{35}$$

Because Kriging is a interpolation process, the model will have no MSE at a sample point. If the MSE is minimized, then the kriging predictor is

$$\hat{y} = \hat{B} + \mathbf{r}^T(\mathbf{x})\mathbf{R}^{-1}(\mathbf{y} - \mathbf{f}\hat{B}), \tag{36}$$

where $\mathbf{y}$ is the vector of responses (objective values) to the sample locations $\{\mathbf{x}^1, \ldots, \mathbf{x}^{n_s}\}$ and $\mathbf{f}$ is a constant vector of all ones with length n_s. In Equation 36 the correlation vector, $\mathbf{r}(\mathbf{x})$, is the correlation between the value at a new location $\mathbf{x}$ and the values at the sampled locations. To use this predictor, $\hat{B}$ and θ_k must both be found, as $\mathbf{r}$ and $\mathbf{R}$ depend on θ_k. The correlation vector is

$$\mathbf{r}(\mathbf{x})^T = [R(\mathbf{x}, \mathbf{x}^1), \ldots, R(\mathbf{x}, \mathbf{x}^{n_s})]. \tag{37}$$

The unknown parameters θ_k are found using maximum likelihood estimation.[36] This approach calculates $\hat{B}$ and the estimated variance using generalized least squares with the following results:

$$\hat{B} = (\mathbf{f}^T\mathbf{R}^{-1}\mathbf{f})^{-1}\mathbf{f}^T\mathbf{R}^{-1}\mathbf{y} \tag{38}$$

$$\hat{\sigma^2} = \frac{1}{n_s}\left((\mathbf{y} - \hat{B}\mathbf{f})^T\mathbf{R}^{-1}(\mathbf{y} - \hat{B}\mathbf{f})\right). \tag{39}$$

Finally, θ_k is found by maximizing the maximum likelihood estimator:

$$\begin{array}{ll} \underset{\theta_k}{\text{Maximize}} & -\frac{n_s \ln(\hat{\sigma}^2) + \ln|\mathbf{R}|}{2} \\ \text{subject to:} & 0 < \theta_k \leq \infty. \end{array}$$

This is a n_v dimensional optimization problem which is well posed. Notice that a kriging model can be built for any values of θ_k; this optimization insures the best choice. In practice when solving for θ_k, $\mathbf{R}$ may become badly scaled; this is overcome using various numerical techniques as described by Lophaven *et al.*[37,38]

B. Direct Cokriging and Indirect Cokriging

The inclusion of zero and first order information at a point into a kriging model is referred to as cokriging. There are two basic types of cokriging, indirect and direct. In the direct method, the kriging model is modified to include gradient information directly.[30,39] Indirect cokriging uses gradient information to calculate extra data points near the sample point via a Taylor series expansion or from the perturbed finite difference locations. Then the ordinary Kriging model is used with the inclusion of the extra samples. Both methods are compared in the results by Liu and Batill.[40]

The kriging method used is determined mainly based on how the gradients have been evaluated. If finite differences are used, then indirect cokriging is preferred since the perturbation points can be directly included into the model. If the gradients are obtained from other methods, then direct cokriging could be used with a slight modification to allow points that don't have derivatives to be included. However, both methods are similar in nature, and the indirect method allows for automatic inclusion of points with and without gradient information and is used exclusively in this work.

C. Warm Starting the Kriging Method

A possible downside to using this kriging method to model the scaling between the variable fidelity models is the fact that during the first few iterations, the kriging model may not have enough points to accurately model this scaling function. In order to accelerate this possible dilemma a warm-start kriging based method is used. As illustrated in Figure 2, 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 built initially from many more sample points and, therefore, with higher precision. The number of initial iterations needed before switching methods remains a research topic. This warm-start approach is not limited by the scaling function used.

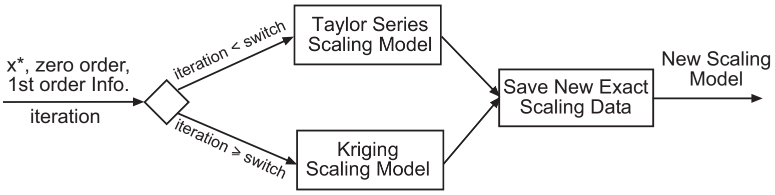


Figure 2. Scaling function generation using the warm-start kriging based method.

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. Additional proposed research involving this method should be done to identify if switching methods, even if the current iterate was rejected, would be more efficient.

VIII. 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 since finite differencing techniques are not required. This is important especially 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[38] *et al.*

IX. Optimization Demonstrations

To demonstrate the savings of the variable fidelity methods presented in this paper, 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. In order 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 may be important, depending on its computational expense realtime 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 sequential quadratic programming (SQP) optimization performed on the high fidelity model alone is also presented for each problem.

A. 2D Analytic Problem

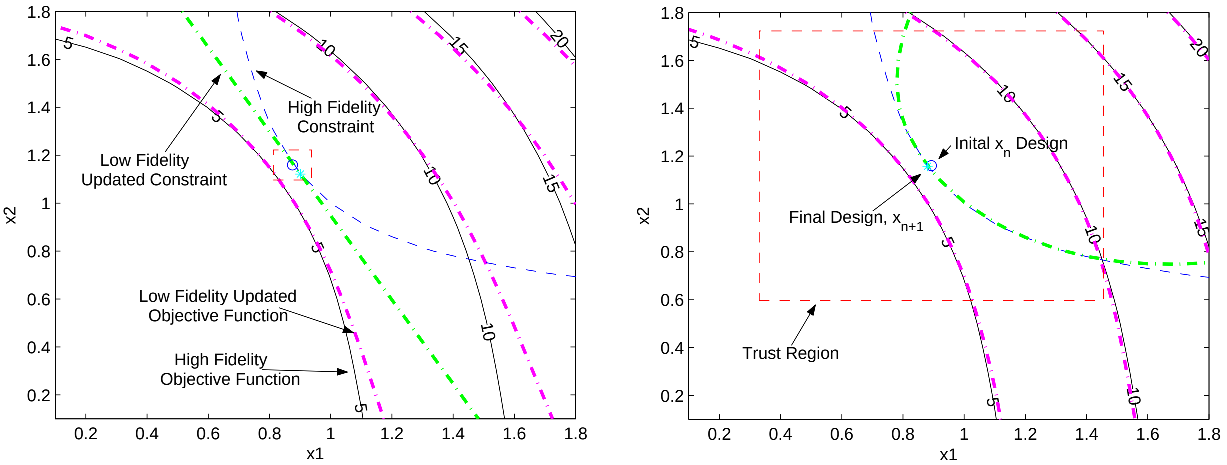
This test problem serves two main purposes: it first allows for verification that the implementations converge and work properly, and second it demonstrates the improvement of the proposed methods over the existing methods. The problem is analytic and two dimensional, making it easy to create contour 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 Equations 40 through 43. 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 using finite differencing.

$$
\begin{aligned}
f_{high} &= 4x_1^2 + x_2^3 + x_1x_2 && (40)\\
g_{high} &= \frac{1}{x_1} + \frac{1}{x_2} - 2 && (41)\\
f_{low} &= 4(x_1 + 0.1)^2 + (x_2 - 0.1)^3 + x_1x_2 + 0.1 && (42)\\
g_{low} &= \frac{1}{x_1} + \frac{1}{x_2 + 0.1} - 2 - 0.001 && (43)
\end{aligned}
$$

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 Figure 3 for the multiplicative first order and kriging methods. 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 in the table for each method is the warm-start 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 of the table shows the savings, in percent of high fidelity function calls, of using the warm-start kriging method. The SQP optimization results are also given at the end of the table.

The plots in Figure 3 depict some of the differences between using a Taylor series approach and kriging based one. The plots also help one visualize various aspects of the framework. 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.

Many observations can be made from comparing the results in Table 1. A significant trend observed is that the warm-start 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 general, the BFGS and SR1 second order methods performed better than the first order methods. And there was no significant difference between the BFGS and SR1 methods for this problem. The full Hessian second order method had mixed results in terms of having fewer high fidelity function calls than the first order model, but with an increase in the number of design variables, the second order method would become very expensive. It is also important to note that the kriging method used alone didn't show any significant savings over the second order methods. This is



(a) 1st order model

(b) kriging model

Figure 3. Comparing the design spaces of the kriging and 1st order multiplicative scaling method after the same number of iterations. The kriging model better approximates the objective and constraints over a larger range, resulting in a bigger trust region.

Table 1. Variable Fidelity Optimization Results for the Analytic Test Problem.

Method	High	Low	Iter.	Final Δ	High	Low	Iter.	Final Δ	Savings
Multiplicative Scaling					Warm-Started Kriging (≥2 Iter.)				
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					Warm-Started Kriging (≥2 Iter.)				
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					Warm-Started Kriging (≥2 Iter.)				
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	-	-	-	-	-	

because the kriging model doesn't 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-start mode.

B. Energy Efficient Transport High-Lift Airfoil Design

For about two decades, starting in the mid 1970s, the National Aeronautics and Space Administration (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 maximum lift for takeoff or landing configurations. The problem was developed from the experimental and numerical work done at NASA Langley.[41–43] The problem consists of nine design variables which control the horizonal, vertical, and rotational orientation of the slat vane and flap relative to their cruise configuration. The rotation is measured positive counter-clockwise about each control surface's leading edge. Figure 4 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.

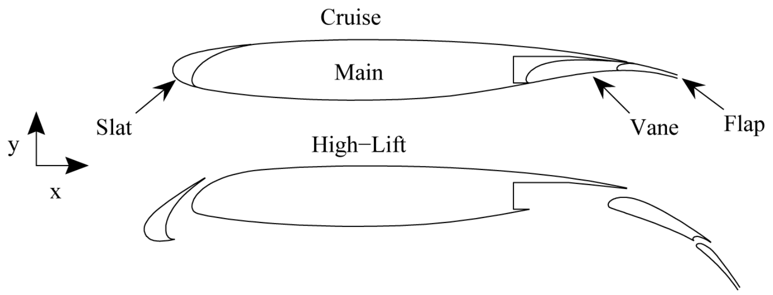
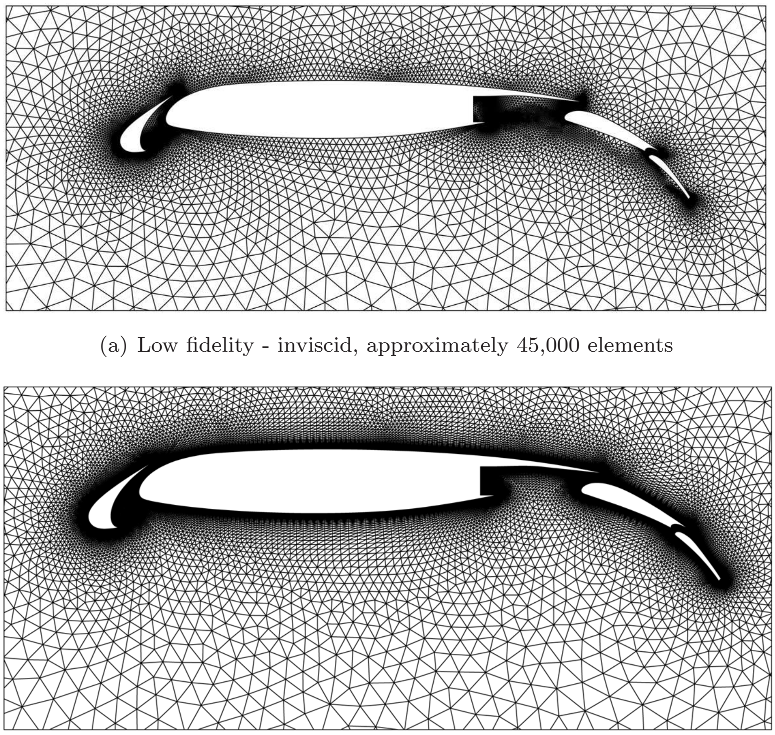


Figure 4. Cruise and high-lift configurations for EET airfoil.

The flow conditions for the problem consisted of a Reynolds number of 9 million, Mach number of 0.25, and an angle of attack 3 degrees. The flow was solved using the inviscid Euler's equations for the low fidelity model. The grid consisting of about 45,000 elements, as seen in Figure 5a, which 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 Figure 5b. The CFD runs took approximately 15 minutes and 2.5 hours 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. It should also be noted that 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.[7]

Both fidelity models were solved using a computational fluid dynamics package developed at NASA Langley called FUN2D.[44,45] This package uses fully unstructured mesh, which was generated using the advancing-front local-reconnection method described by Marcum.[46,47]

The results of the optimization trials are presented Table 2. Based on the results from the analytic case, only the adaptive hybrid method was used on this problem. Using the same convergence criteria as before, all of the trials converged to the same solution. For this problem, gradients were supplied via the surrogate models, simulating an adjoint approach.



(a) Low fidelity - inviscid, approximately 45,000 elements

(b) High fidelity - viscid, approximately 100,000 elements

Figure 5. 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-start kriging based approach can save a significant number of function calls. Each variable fidelity method also used less 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 may be that the scaled kriging model becomes very flat near the optima.

X. Conclusions and Future Work

In this work, many different scaling techniques are presented for use in variable fidelity optimization. The focus is kriging based techniques and adaptive hybrid methods which are given here for the first time. The

Table 2. Variable Fidelity Optimization Results for the AEET Problem.

Method	High	Low	Iter.	Final Δ	High	Low	Iter.	Final Δ	Savings
Adaptive Hybrid Scaling					Warm-Started Kriging (≥3 Iter.)				
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	-	-	-	-	-	

methods were presented theoretically and then applied to two demonstration problems. The results of the problems indicate that the warm-started kriging based method can save substantial computational cost by lowering the number of high fidelity function calls required for optimization. The adaptive hybrid method showed 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.

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 for this. Also, the kriging models may be expensive to build and update after each iteration, depending on the number of design variables and sample points used; though, this cost may be insignificant, depending on the relative cost of the high fidelity model. The only drawback observed at this point for the use of the hybrid method is that it requires more memory to store both the additive and multiplicative models; again however, this is more than likely offset by the savings in high fidelity calls required.

Acknowledgments

This research effort was supported in part by the following grants and contracts: AFRL / DARPA / Anteon Corporation Contract F33615-98-D-3210, ONR Grant N00014-02-1-0786, and NSF Grant DMI-0114975. The authors would also like to 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, 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 & Optimization Symposium*, No. AIAA 96-4102, Bellevue, WA, 4-6 September 1996, pp. 1056–1059.

[4] Dennis, J. and Torczon, V., "Managing approximation models in optimization," 1996, In Multidisciplinary Design Optimization: State of the Art. Alexandrov, N. and Hussaini, M. Y., editors.

[5] 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(2), March-April 1993, pp. 283–288.

[6] 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," No. AIAA paper 2000-4886, September 2000.

[7] 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," *Proceedings of the 38th AIAA Aerospace Sciences Meeting and Exhibit*, No. AIAA 2000-0841, Reno, Nevada, 10-13 January 2000.

[8] Alexandrov, N. M. and Lewis, R. M., "First-Order Approximation and Model Management in Optimization, in Large-Scale PDE-Constrained Optimization," *Springer-Verlag*, 2001.

[9] Giunta, A. A. and Eldred, M. S., "Implementation of a Trust Region Model Management Strategy in the DAKOTA Optimization Toolkit," No. AIAA-2000-4935, 2000.

[10] Booker, A. J., Dennis, J. E., Frank, P. D., Serafini, D. B., Torczon, V., and Trosset, M. W., "A Rigorous Framework for Optimization of Expensive Function by Surrogates," No. CRPC-TR98739-S, April 1998.

[11] 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.

[12] Rodríguez, J. F., Renaud, J. E., and Watson, L. T., "Trust Region Augmented Lagrangian Methods for Sequential Response Surface Approximation and Optimization," *Transactions of the ASME*, Vol. 120, 1998, pp. 58–66.

[13] 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, 2000, pp. 139–154.

[14] Sobieszczanski-Sobieski, J., "Sensitivity of Complex, Internally Coupled Systems," *AIAA Journal*, Vol. 28, No. 1, 1990.

[15] 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, 2001, pp. 24–34, Published by Springer-Verlag.

[16] Pérez, V. M., Renaud, J. E., and Watson, L. T., "Interior Point Sequential Approximate Optimization Methodology," *Proceedings of the 10th AIAA/USAF/NASA/ISSMO Symposium on Multidsciplinary Analysis & Optimization*, No. AIAA 2002-5505, Atlanta, GA, 4-6 September 2002.

[17] Pérez, V. M., Renaud, J. E., and Watson, L. T., "Reduced Sampling for Construction of Quadratic Response Surface

Approximations Using Adaptive Experimental Design," *Proceedings of the 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, No. AIAA 2002-1587, Denver, CO, 22-25 April 2002.

[18]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," No. ICASE Report 97-50, NASA/CR-201745, 2001.

[19]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.

[20]Lewis, R. M. and Nash, S. G., "A Mulitgrid Approach to the Optimization of Systems Governed by Differential Equations," No. AIAA-2000-4890, 2000.

[21]Alexandrov, N. M. and Lewis, R. M., "First-Order Frameworks for Managing Models in Engineering Optimization," *First International Workshop on Surrogate Modelling and Space Mapping for Engineering Optimization*, Lyngby, Denmark, 16-18 November 2000.

[22]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 & Materials Conference*, No. AIAA 2002-1587, Palm Springs, CA, 19 - 22 Apr 2004 2002, pp. AIAA 2004–1763.

[23]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 Multidsciplinary Analysis & Optimization Conference*, No. AIAA 2004-4457, Albany, NY, 30 August - 1 September 2004.

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

[25]Broyden, C. G., "The Convergence of a Class of Double-rank Minimization Algorithms," *J. Inst. Maths. Applics.*, Vol. 6, 1970, pp. 76–90.

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

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

[28]Shanno, D. F., "Conditioning of Quasi-Newton Methods for Function Minimization," *Mathematics of Computing*, Vol. 24, 1970, pp. 647–656.

[29]Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P., "Design and Analysis of Computer Experiments," *Statistical Science*, Vol. 4, 1989, pp. 409–423.

[30]Chung, H.-S. and Alonso, J. J., "Using Gradients to Construct Cokriging Approximation Models for High-Dimensional Design Optimization Problems," *40th AIAA Aerospace Sciences Meeting and Exhibit*, No. AIAA 2002-0317, Reno, NV, 14-17 January 2002.

[31]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*, No. DET2003/DAC-48762, Chicago, IL, 2-6 September 2003.

[32]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*, No. DETC2004/DAC-57300, Salt Lake City, Utah, 28 September - 2 October 2004.

[33]Denison, D. G. T., Holmes, C. C., Mallick, B. K., and Smith, A. F. M., *Bayesian Methods for Nonlinear Classification and Regression*, John Wiley & Sons, Inc., 2002.

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

[35]Simpson, T. W., Mauery, T. M., Korte, J. J., and Mistree, F., "Comparison of Response Surface and Kriging Models for Multidisciplinary Design Optimization," No. AIAA-98-4755, 1998.

[36]Giunta, A. A. and Watson, L. T., "A Comparison of Approximation Modeling Techniques: Polynomial Versus Interpolating Models," No. AIAA-98-4758, 1998.

[37]Lophaven, S. N., Nielsen, H. B., and Søndergaard, J., "DACE a Matlab Kriging Toolbox," Tech. Rep. IMM-REP-2002-12, Technical University of Denmark, DK-2800 Kongens Lyngby - Denmark, 1 August 2002, version 2.0.

[38]Lophaven, S. N., Nielsen, H. B., and Søndergaard, J., "Aspects of the Matlab Toolbox DACE," Tech. Rep. IMM-REP-2002-13, Technical University of Denmark, DK-2800 Kongens Lyngby - Denmark, 2002.

[39]Chung, H.-S. and Alonso, J. J., "Design of a Low-Boom Supersonic Business Jet Using Cokriging Approximation Models," *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, No. AIAA 2002-5598, Atlanta, GA, 4-6 September 2002.

[40]Liu, W. and Batill, S., "Gradient-Enhanced Response Surface Approximations Using Kriging Models," *Proceedings of the 9th AIAA/ISSMO Symposium and Exhibit on Multidsciplinary Analysis and Optimization*, No. AIAA 2002-5456, Atlanta, GA,, 4-6 September 2002.

[41]Morgan, H. L. J., "Experimental Test Results of Energy Efficient Transport (EET) High-Lift Airfoil in Langley Low-Turbulance Pressure Tunnel," Tech. Rep. TM-2002-211780, NASA, Langley Research Center, Hampton Virginia, 2002.

[42]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.

[43]Gatlin, G. M. and McGhee, R. J., "Study of Semi-Span Model Testing Techiniques," *Proceedings of the 14th Applied Aerodynamics Conference*, No. AIAA 96-2386, New Orleans, LA, 17-20 June 1996.

[44]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. pp. 1–21.

[45]Anderson, W. K., Rausch, R. D., and Bonhaus, D. L., "Implicit/Multigrid Algorithms for Incompressible Turbulent Flows on Unstructured Grids," *J. Comp. Phys.*, Vol. 128, 1996, pp. 391–408.

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

[47]Marcum, D. L., "Advancing-Front/Local-Reconnection (AFLR) Unstructured Grid Generation," *Computational Fluid Dynamics Review*, 1998.