45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference
19 - 22 April 2004, Palm Springs, California

AIAA 2004-1763

Multilevel Variable Fidelity Optimization of a Morphing Unmanned Aerial Vehicle

Shawn E. Gano* Victor M. Pérez † John E. Renaud ‡ Stephen M. Batill §

University of Notre Dame, Notre Dame, Indiana, 46556

Brian Sanders¶

Air Force Research Laboratory, Wright-Patterson AFB, Ohio, 45433

The Morphing Aircraft Structures (MAS) program has grown substantially in technology and has received more attention in recent years. One main initiative of this project is to develop aerial vehicles that are capable of radical shape change. Such shape changes should enable the vehicle to efficiently perform single missions that normally would require two or more different aircraft. Many design optimization issues arise for such systems. Morphing aircraft have multiple configurations which create multiobjective and possibly multilevel optimization problems. The problems are multiobjective because of the performance trade-offs occurring between each morphed state. In many morphing concepts one configuration must be defined in order to design a second state; this produces a multilevel design problem. The complexity of the simulation model and the nested formulation of the multilevel design problem results in a very computationally intensive optimization problem. This paper includes a discussion of solution strategies for these multiobjective, multilevel morphing aircraft design problems. Two design tools are explored to combat the described issues of the optimization problem: conversion to a single-level design problem and the use of variable fidelity optimization. A study is performed comparing the results obtained from the optimization processes of both a multi-level design problem and its corresponding single level problem. Finally a variable fidelity optimization framework is discussed and applied to design a morphing concept; the two fidelity models include a high fidelity computational fluid dynamics simulation and a low fidelity panel method.

Nomenclature

α	Angle of attack
γ	Additive scaling function, error between high and low fidelity models
Γ	Circulation distribution
$\tilde{\gamma}$	Approximated additive scaling function
Δ	Trust region size
ϵ_f	Convergence tolerance for objective function
ϵ_x	Convergence tolerance for design variables
μ	l_1 penalty function weight
ρ	Trust region ratio
ϕ	Velocity potential
∇	Hessian operator
c_d	Sectional drag coefficient

*Graduate Research Assistant, Aerospace and Mechanical Engineering, e-mail: sgano@nd.edu, Student Member AIAA
†Postdoctoral Research Associate, Aerospace and Mechanical Engineering, e-mail: vperez@nd.edu, Member AIAA
‡Professor, Aerospace and Mechanical Engineering, e-mail: jrenaud@nd.edu, Associate Fellow AIAA.
§Department Chair, Aerospace and Mechanical Engineering, e-mail: sbatill@nd.edu, Associate Fellow AIAA.
¶AFRL/VASA, e-mail: Brian.Sanders@wpafb.af.mil, Associate Fellow AIAA.

c_l	Sectional lift coefficient
f_{high}	High fidelity objective function
f_{low}	Low fidelity objective function
H	Hessian matrix
$\mathbf{l}$	Lower variable bounds
m	Number of panels used to describe an airfoil
$\hat{n}_j$	Outward unit normal of the jth panel
P	l_1 penalty function
s_j	Distance from the front edge of the jth panel
s_k	Change in design vector between iterations, $\mathbf{x}_{k+1} - \mathbf{x}_k$
S_j	Length of the jth panel
T_{top}, T_{bot}	Thickness of the top and bottom buckled airfoils
$\mathbf{u}$	Upper variable bounds
V_∞	Free stream velocity
w_1, w_2	Weights for multiobjective function
$\mathbf{x}$	Design Vector
$\mathbf{x}_c$	Current design Vector
y_k	Change in gradient between iterations, $\nabla f_{k+1} - \nabla f_k$

I. Introduction

DESIGNING morphing aircraft that can undergo drastic shape change presents many technological and computational challenges. Typical morphing aircraft problems tend to be multilevel in nature because of their geometrical state changes. Combining this multilevel problem with high fidelity analysis creates a tremendous computational expense. The focus of this work is to explore options and tools available to reduce this computational burden, focusing mainly on reposing the problem to a single level formulation and using variable fidelity optimization techniques.

The shape change that the vehicle undergoes should be relatively inexpensive in terms of actuation energy, which means the cost of the shape change in terms of power and physical hardware needed to morph should be much less than the resulting performance gain.[1] These morphing vehicles could be used for missions that are not achievable by current aircraft standards or for completely new missions that were not thought possible by previous flying machines. Such missions could include: reconnaissance and attack (as a single mission, instead of two separate missions), hunt and rescue, and biological or nuclear weapon detection.

The buckle wing unmanned aerial vehicle (UAV), which is being developed at the University of Notre Dame, fits the morphing paradigm well and will be used to demonstrate the computational tools discussed in this paper. The buckle wing in cruise conditions looks like a typical UAV with a standard high aspect ratio wing; though the wing may be slightly thicker. However, the wing actually is composed of two thinner wings which are fused together to form this high aspect ratio wing, and when the need for maneuverability arises the aircraft can, via a buckling load applied at the wing tips, morph into a configuration that somewhat resembles a biplane with its outboard wing tips joined together. When the UAV is in the buckled state, it can generate much more lift and becomes more agile. Figure 1 depicts the front view of the buckle wing in its buckled state and shows the UAV in the same state but from an isoparametric view.

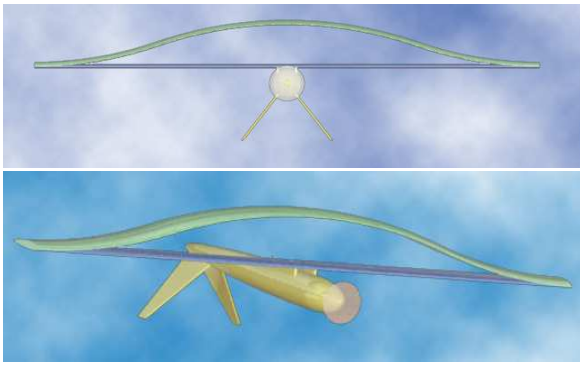

Figure 1. Front view (top) and isoparametric view (bottom) of the Buckle Wing UAV.

The issues associated with this morphing concept concern how to best design the airfoil shapes. The airfoil of the fused configuration must be a good long range or long endurance shape; this shape must also allow for a good parting line, such that when the wing buckles into the buckled or split wing mode, the two airfoils are maneuverable. A first attempt at solving this conforming airfoil problem was addressed by Gano et al.[2] In that work, a full description of this conforming airfoil problem was given and was partially solved using a multiobjective multilevel formulation. However, it only addressed half of the problem; the paper only solved for the best separation line for a given fused shape. The focus of this paper is to solve the full optimization of the conforming airfoil problem, which is a multiobjective problem. Because this problem requires computing the aerodynamic properties of many different airfoil shapes and since computational fluid dynamics (CFD) methods are quite expensive, variable fidelity aerodynamic analysis is utilized to decrease the computational burden. In this paper, a single level and a multilevel approach are discussed and compared. Then the single level formulation is optimized using a variable fidelity framework.

II. Multiobjective Multilevel Problem Formulations

When solving the wing or airfoil design problem of a morphing vehicle, the designer must consider the various flight regimes the craft can operate. This problem is inherently a multiobjective problem because of these various flight regimes. In the case of the buckle wing, there are two main objectives; one is long range and/or endurance (when the wing is in the fused form) and the other is maneuverability (when the wing is buckled).

The optimization problem for the buckle wing can be expressed both as a multilevel optimization problem or using a single level approach; both of these methods are described in the following sections. In both cases the lift to drag ratio is used as a measure of range and endurance; the higher the lift to drag ratio the better the range or endurance. Also the lift coefficient is used as a measure of the maneuverability. Both of these measures were found by analyzing the Breguet equations, load factor, and the turn rate.[2]

A. Multilevel Formulation

In Gano *et al.*[2,3] a bi-level approach is presented. The multilevel optimization formulation deals with the fused airfoil shape and the parting shape of the fused airfoil separately. The system level optimizer controls the shape of the fused airfoil, while trying to maximize a combination of the lift to drag ratio (for range) of the fused shape and the lift coefficient for maneuverability of the buckled airfoils. The buckled shape is determined by a lower level optimization which given a fused shape, finds the optimal parting line for the two buckled airfoils. The design variables for both optimizers are variables that control the shapes of the airfoils. This formulation can be expressed mathematically as:

$$\begin{aligned} \underset{\mathbf{x}}{\text{maximize}}: \quad & w_1 \left.\frac{c_l}{c_d}\right|_{fused} + w_2 \, c_l^*|_{split} \\ \text{subject to}: \quad & \mathbf{l} \leq \begin{pmatrix} c_l|_{fused} \\ \mathbf{Aero}(\mathbf{x}) \\ \mathbf{Struct}(\mathbf{x}) \\ \mathbf{x} \end{pmatrix} \leq \mathbf{u}. \end{aligned} \tag{1}$$

Where $c_l^*|_{split}$ is the optimal value of the sub optimization airfoil conforming problem,

$$\begin{aligned} \underset{\mathbf{x}_{sub}}{\text{maximize}}: \quad & c_l|_{split} \\ \text{subject to}: \quad & \mathbf{l} \leq \begin{pmatrix} \left.\frac{c_l}{c_d}\right|_{split} \\ \mathbf{Aero}(\mathbf{x}_{sub}) \\ \mathbf{Struct}(\mathbf{x}_{sub}) \\ \mathbf{x_{sub}} \end{pmatrix} \leq \mathbf{u}. \end{aligned} \tag{2}$$

In this formulation, w_1 and w_2 are the weights between range / endurance and maneuverability, $\mathbf{l}$ and $\mathbf{u}$ are the lower and upper bound vectors, and **Aero** and **Struct** are any other aerodynamic and structural constraints that are placed on the design. This multilevel approach is summarized in Figure 2.

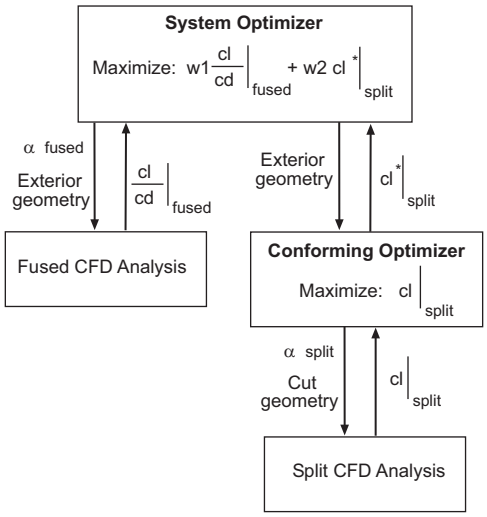

Figure 2. Multi-level optimization formulation flow chart.

The advantages of this formulation are that the system optimizer may face fewer local minima not having to deal directly with both the shape of the fused airfoil and the shape of the separation line. This method, on the other hand, may be more computationally expensive, especially if the system level optimization needs gradients of the lower level optimization.

B. Single-Level Optimization Formulation

Because the design variables in the multilevel formulation are not directly coupled with one another, it is possible to rewrite the problem in a single-level formulation. In this formulation the system optimizer controls both the design variables corresponding to the shape of the fused airfoil and that of the separation line. One difficulty of reformulating the problem is making sure that the separation line doesn't extend outside of the fused airfoil shape. This problem can be prevented if the separation variables are required to be a percentage of the thickness of the fused shape. The problem is then stated concisely as:

$$\underset{\mathbf{x},\mathbf{x}_{sub}}{\text{maximize :}} \qquad w_1 \left.\frac{c_l}{c_d}\right|_{fused} + w_2 \; c_l|_{split} \tag{3}$$

$$\text{subject to :} \qquad \mathbf{l} \leq \begin{pmatrix} c_l|_{fused} \\ \left.\frac{c_l}{c_d}\right|_{split} \\ \mathbf{Aero}(\mathbf{x}) \\ \mathbf{Struct}(\mathbf{x}) \\ \mathbf{Aero}(\mathbf{x}_{sub}) \\ \mathbf{Struct}(\mathbf{x}_{sub}) \\ \mathbf{x} \\ \mathbf{x_{sub}} \end{pmatrix} \leq \mathbf{u}. \tag{4}$$

The advantage to this method is that it may be computationally less expensive since there is no lower level optimization for each iteration of the system optimizer. A disadvantage may be that many local optima occur because the buckled shape design variables are now functions of the thickness of the fused design variables.

III. First Order Optimality Analysis

In this section the formulations presented in Section II are rewritten in a simplified form for the analysis. This simplification helps in a generalization of bi-level formulations. The nomenclature used for both formulations will follow that of the multi-level formulation. The subindex s refers to the system-level while the subindex ss refers to the lower-level part. Following this convention, design variables in the upper level are $\mathbf{x}_s$, the objective function is f_s and the constraints are g_s. The lower level components are $\mathbf{x}_{ss}$, f_{ss} and g_{ss} respectively.

A. Multilevel Formulation

The upper level optimization problem (1) can be written as:

$$\begin{aligned} \underset{\mathbf{x}_s}{\text{minimize}}: \quad & f_s(\mathbf{x}_s) = f_1(\mathbf{x}_s) + f^*_{ss}(\mathbf{x}_s) \\ \text{subject to}: \quad & \mathbf{g}_s(\mathbf{x}_s) \geq 0. \end{aligned} \tag{5}$$

Where $f^*_{ss}(\mathbf{x}_s)$ is the optimal value of the sub optimization airfoil conforming problem (2) for a given value of $\mathbf{x}_s$,

$$\begin{aligned} \underset{\mathbf{x}_{ss}}{\text{minimize}}: \quad & f_{ss}(\mathbf{x}_s, \mathbf{x}_{ss}) \\ \text{subject to}: \quad & \mathbf{g}_{ss}(\mathbf{x}_s, \mathbf{x}_{ss}) \geq 0. \end{aligned} \tag{6}$$

It is important to underline that in this formulation there are no shared variables between the upper and lower optimization problems, and the objective function at the system level contains the optimum objective function of the subsystem level.

The fist order KKT optimality conditions for the system level problem are given by:

$$\nabla_{\mathbf{x}_s} f_1 + \nabla_{\mathbf{x}_s} f^*_{ss}(\mathbf{x}^*_s) - \sum_i \lambda^* \nabla_{\mathbf{x}_s} g_s(\mathbf{x}^*_s) = 0, \tag{7}$$

$$g_s(\mathbf{x}^*_s) \geq 0, \tag{8}$$

$$\lambda^* \geq 0, \tag{9}$$

$$\lambda^* g_s(\mathbf{x}^*_s) = 0, \tag{10}$$

where $\nabla_{\mathbf{x}_s} f^*_{ss}(\mathbf{x}^*_s)$ is the post optimality sensitivity of the lower level optimization. The KKT equations for the lower level optimization are:

$$\nabla_{\mathbf{x}_{ss}} f_{ss}(\mathbf{x}^*_s, \mathbf{x}^*_{ss}) - \sum \mu^* \nabla_{\mathbf{x}_{ss}} g_{ss}(\mathbf{x}^*_s, \mathbf{x}^*_{ss}) = 0, \tag{11}$$

$$g_{ss}(\mathbf{x}^*_s, \mathbf{x}^*_{ss}) \geq 0, \tag{12}$$

$$\mu^* \geq 0, \tag{13}$$

$$\mu^* g_{ss}(\mathbf{x}^*_s, \mathbf{x}^*_{ss}) = 0. \tag{14}$$

The post optimality sensitivities of the lower level objective function with respect to the upper level design variables (see[3]) are:

$$\nabla_{\mathbf{x}_s} f^*_{ss}(\mathbf{x}^*_s) = \frac{\partial f_{ss}}{\partial \mathbf{x}_s}(\mathbf{x}^*_s, \mathbf{x}^*_{ss}) - \sum \mu^* \frac{\partial g_{ss}}{\partial \mathbf{x}_{ss}}(\mathbf{x}^*_s, \mathbf{x}^*_{ss}). \tag{15}$$

So the system of equations can be rewritten as:

$$\nabla_{\mathbf{x}_s} f1 + \frac{\partial f_{ss}}{\partial \mathbf{x}_s}(\mathbf{x}_s^*, \mathbf{x}_{ss}^*) - \sum \mu^* \frac{\partial g_{ss}}{\partial \mathbf{x}_{ss}}(\mathbf{x}_s^*, \mathbf{x}_{ss}^*) - \sum_i \lambda^* \nabla_{\mathbf{x}_s} g_s(\mathbf{x}_s^*) = 0, \tag{16}$$

$$\nabla_{\mathbf{x}_{ss}} f_{ss}(\mathbf{x}_s^*, \mathbf{x}_{ss}^*) - \sum \mu^* \nabla_{\mathbf{x}_{ss}} g_{ss}(\mathbf{x}_s^*, \mathbf{x}_{ss}^*) = 0, \tag{17}$$

$$g_s(\mathbf{x}_s^*) \geq 0, \tag{18}$$

$$\lambda^* \geq 0, \tag{19}$$

$$\lambda^* g_s(\mathbf{x}_s^*) = 0, \tag{20}$$

$$g_{ss}(\mathbf{x}_s^*, \mathbf{x}_{ss}^*) \geq 0, \tag{21}$$

$$\mu^* \geq 0, \tag{22}$$

$$\mu^* g_{ss}(\mathbf{x}_s^*, \mathbf{x}_{ss}^*) = 0. \tag{23}$$

B. Single-level Formulation

The single-level formulation is given by

$$\begin{aligned} \underset{\mathbf{x}_s, \mathbf{x}_{ss}}{\text{minimize}} : \quad & f_s(\mathbf{x}_s, \mathbf{x}_{ss}) = f_1(\mathbf{x}_s) + f_{ss}(\mathbf{x}_s, \mathbf{x}_{ss}) \\ \text{subject to} : \quad & \mathbf{g}_s(\mathbf{x}_s) \geq 0 \\ & \mathbf{g}_{ss}(\mathbf{x}_{ss}) \geq 0. \end{aligned} \tag{24}$$

The first order optimality conditions are

$$\nabla_{\mathbf{x}_s} f1 + \nabla_{\mathbf{x}_s} f_{ss}(\mathbf{x}_s^*, \mathbf{x}_{ss}^*) - \sum_i \lambda^* \nabla_{\mathbf{x}_s} g_s(\mathbf{x}_s^*) - \sum_j \mu^* \nabla_{\mathbf{x}_s} g_{ss}(\mathbf{x}_s^*, \mathbf{x}_{ss}^*) = 0 \tag{25}$$

$$\nabla_{\mathbf{x}_{ss}} f_{ss}(\mathbf{x}_s^*, \mathbf{x}_{ss}^*) - \sum_j \mu^* \nabla_{\mathbf{x}_{ss}} g_{ss}(\mathbf{x}_s^*, \mathbf{x}_{ss}^*) = 0 \tag{26}$$

$$g_s(\mathbf{x}_s^*) \geq 0 \tag{27}$$

$$g_{ss}(\mathbf{x}_s^*, \mathbf{x}_{ss}^*) \geq 0 \tag{28}$$

$$\lambda^* \geq 0 \tag{29}$$

$$\mu^* \geq 0 \tag{30}$$

$$\lambda^* g_s(\mathbf{x}_s^*) = 0 \tag{31}$$

$$\mu^* g_{ss}(\mathbf{x}_s^*, \mathbf{x}_{ss}^*) = 0 \tag{32}$$

It is clear that a design point $(\mathbf{x}_s^*, \mathbf{x}_{ss}^*)$ that is a KKT point for the single-level system is a KKT point for the bi-level system as both systems of equations are equivalent. Note, however, that these are first order optimality conditions and such design point can be only an equilibrium point of either of the formulations without being an optimum. Second order conditions are hard to generalize without knowing the specific nature of the functions and constraints, however it is unlikely to encounter such situation. Furthermore, many optimization algorithms have convergence criteria based on first order KKT conditions, theoretically suffering from the same problem. The expected savings implementing the single-level formulation are far greater an advantage than this low-probability potential problem. Having control of all the design variables at the single level, gives the optimizer more freedom to explore the design space. The problem is suspected to have several local optima and therefore, starting from the same point for the bi-level and the single-level formulations does not guarantee convergence to the same local optimum.

IV. Description of the Buckle Wing Problem

The concept design of the buckle wing starts with the design of the fused airfoil. The buckle configuration is obtained by defining a cut from the leading edge to the trailing edge separating the lower and the upper airfoils of the buckled configuration. The optimization changes the shape of the fused airfoil and the cut. The shape of the fused airfoil is a weighted sum of three basis shapes, this approach is known as a reduced

basis method and was first used in airfoil design by Vanderplaats.[4] The basis shapes employed were standard airfoils: E-387, NACA 64A010, and the S2055. The cut was defined as a weighted sum of the lower (S_L) and the upper (S_U) surfaces of the fused airfoil. The weight is a quadratic function of the chord. The coefficients of this quadratic function are the other three design variable. So the weighting function is defined as:

$$w(z) = x_4 + x_5 * z + x_6 * z^2, \tag{33}$$

where z is the horizontal coordinate from leading edge to trailing edge along the cord. The cut, is then

$$S_C(z) = (1 - w(z)) * S_L(z) + w(z) * S_U(z). \tag{34}$$

To avoid the problem of having a cut that touches or crosses one of the surfaces, or even produced an airfoil too thin, two constraints are added for the minimum and maximum value of this quadratic weight: $0.3 \leq w(z) \leq .7$. for $0 \leq z \leq 1$ which is the cord length. Furthermore, bounds can be imposed to at least one of the cut design variables as $0.3 \leq x_4 \leq 0.7$

For the computational results obtained in this paper the following explicit optimization problem was used:

$$\begin{aligned}
\underset{\mathbf{x}}{\text{Maximize}} \quad & w_1 \frac{c_{l_{fused}}}{c_{d_{fused}}} + w_2 c_{l_{split}} \\
\text{subject to:} \quad & c_{l_{fused}} \geq 0.65. \\
& c_{d_{fused}} \leq 0.0075 \\
& x_1 + x_2 + x_3 \leq 1.6 \\
& x_1 + x_2 + x_3 \geq 0.75 \\
& 0 \leq x_1, x_2, x_3 \leq 1.3 \\
\text{and} \quad & \\
& \frac{c_{l_{split}}}{c_{d_{split}}} \geq 100 \\
& T_{top} \geq 0.06 \\
& T_{bot} \geq 0.06 \\
& 0.3 \leq x_4 \leq 0.7 \\
& -10 \leq x_5 \leq 10 \\
& -1 \leq x_6 \leq 1
\end{aligned}$$

Where x_1, x_2, x_3 are the design variables that represent the weights of the basis functions which described the fused shape and x_4, x_5, x_6 are the design variables that define the cut geometry. The last three design variables are used in the lower level optimization in the multilevel formulation. For the multilevel problem the first six constraints are imposed at the upper level while the remaining constraints are imposed in the lower level. In this formulation T_{top} and T_{bot} are the thickness of the top and bottom airfoils in the buckled state. A description of both the high fidelity and low fidelity analysis packages used to characterize the aerodynamic performance of the airfoils is described in Section VII.A.

V. Optimization results with a low-level fidelity model

As stated in Section II the bi-level approach is expected to be more expensive than the single-level formulation. In this section optimizations with both formulations are compared to show the relative savings in number of analysis performed and the ability of the single-level formulation to produce an optimum of the bi-level formulation.

In Rusnell et. al.[5] a Pareto frontier is identified for the bi-level formulation of the buckle wing concept. Although the analysis code is the same, in this paper some modifications to the formulation of the cut and the constraints have been implemented for the robustness of the problem as described in the previous section. In this paper the low-fidelity results will be concentrated in the comparison of the bi-level and the single-level formulations.

The starting point is $x0 = [1.0\ 0\ 0\ 0.5\ 0\ 0]$. The optimization using the single level formulation resulted in an optimum at $x = [0.1308\ 0.0582\ 0.5681\ 0.6406\ 0.2284\ -0.2081]$. The resulting buckle wing is shown on

the left in Figure 3. The optimum values of the coefficients are presented in Table 1. The weighting function for the cut is presented on the right in Figure 3.

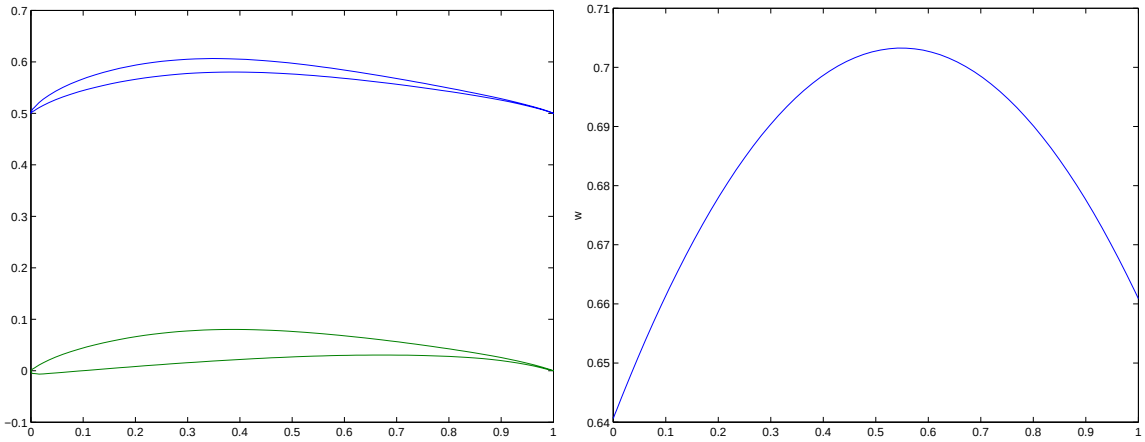

Figure 3. Optimal split airfoil shape produced by the single-level formulation (Left). Optimal weight function for the cut (Right).

Table 1. Coefficient values for the starting and optimum designs.

Coefficient	x_0	x^*
$c_{l_{fused}}$	0.7607	1.255
$c_{d_{fused}}$	0.0058	0.0077
$c_{l_{buc}}$	1.3574	2.16
$c_{d_{buc}}$	0.0163	0.0212

This optimum point was used as starting point in the bi-level formulation to check whether the point was also a numerical optimum of it or not. The optimization performed one cycle and returned as an optimum $x^* = [0.1308\ 0.0582\ 0.5681\ 0.6349\ 0.2128\ -0.1714]$. The optimizer returned the same point for the basis variables and a close neighboring point for the cut variables. This illustrates our claim that the single-level formulation is capable of computing KKT points for the bi-level formulation.

Next we compare the costs of using the single-level formulation versus the bi-level formulation. For that the bi-level formulation was executed using the original starting point. The design space is suspected to be highly non-convex, with several local minima. As a result the optimum obtained using the same starting point is a different minimum: $x^* = [0.0000\ 0.7948\ 0.7119\ 0.6044\ 0.1560\ -0.0604]$. This point has a $c_{l_{fused}} = 2.1742$, $c_{d_{fused}} = 0.0050$, $c_{l_{buc}} = 3.8213$, $c_{d_{buc}} = 0.0396$. Although this point is a better optimum that the one obtained by the single-level formulation, both are local optimum in the bi-level scheme. The comparative cost of using the single-level and the bi-level formulation can be represented by the number of aerodynamic analysis required to find the optimum. In the case of the single-level formulation, each call to evaluate a design point required three calls to the aerodynamic analysis,one for the fused and one for each the lower and upper airfoils of the buckle configuration. It required 68 function calls for a total of 204 analysis. For the case of the bi-level formulation, each function call at the upper level requires one analysis for the fused airfoil and several lower-level function calls. Each lower-level function call requires two analysis, one for the lower and another for the upper airfoils. In total the bi-level required 73 upper-level function calls, that adds to a total of 1623 analysis! Although the comparison is not completely fair, since both reached different local minimum, the number of iterations at the system level was the same: 8.

VI. Variable Fidelity Optimization

As described in Section III multi-level morphing aircraft problems can be converted to a single level problems as long as some precautions are taken. Detailed design on this single level formulation is still computationally expensive, especially if a high fidelity CFD package is used for the analysis and there are a moderate or large number of design variables. To reduce the number of high fidelity CFD simulations that need to be performed a variable fidelity optimization method can be useful. In this research a variable fidelity method that builds scaling function which when applied to the low fidelity analysis approximates the high fidelity response. This scaling function physically is a model of the error between the high and low fidelity models.

A. General Framework

The typical framework for variable fidelity optimization is depicted in Figure 4 and is based in part on work done by Alexandrov.[6,7] In the framework, the scaling function is first approximated using information of the high and low fidelity models at the starting point. This model can approximately update the low fidelity result, f_{low}, to match the high fidelity model, f_{high}. Once the scaling function is constructed, the combination of the low fidelity analysis and scaling function is optimized. After the trust region managed optimization converges, the scaling function is re-constructed using high and low fidelity information at the new design point. Then the lower fidelity model and new scaling function are optimized. This process repeats until it converges to the optimal point of the high fidelity model. The process is typically controlled by the use of the trust region model management strategy[8] which is updated after the convergence test is performed. The trust region methodology is discussed in more detail in Section F.

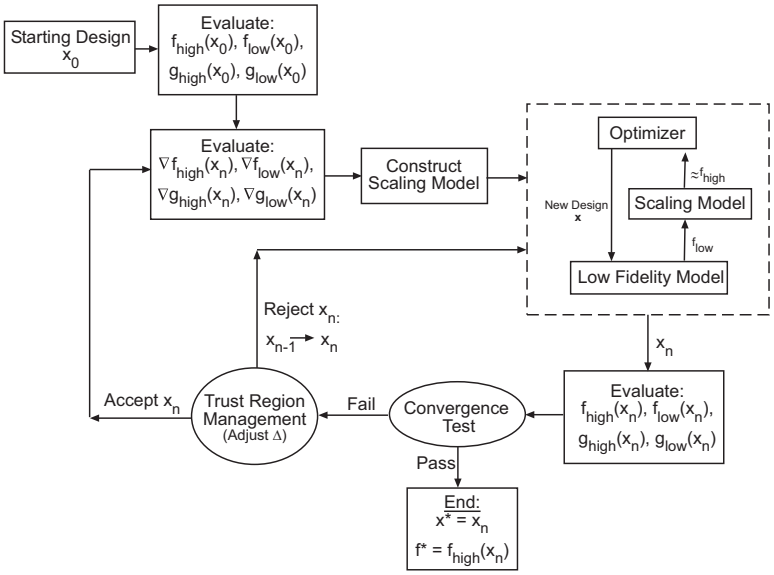

Figure 4. General variable fidelity framework.

The optimizer used in the framework, as seen in Figure 4, can be almost any standard method. In the work done by Alexandrov,[6] three optimizers were compared: augmented Lagrangian method, multilevel algorithms for large-scale constrained optimization (MAESTRO)[9] (used for coupled MDO problems), and sequential quadratic programming (SQP). For typical single discipline problems, Alexandrov found SQP to be the most promising. This is the motivation for its use in this research.

The sequential quadratic programming (SQP) optimizer used was Mathworks MATLAB's *fmincon* from the Optimization Toolbox. In this method, a quadratic programming (QP) subproblem is solved at each iteration. An estimate of the Hessian of the Lagrangian is updated at each iteration using the BFGS formula. A line search is performed using a merit function similar to that proposed by Han[10] and Powell.[11,12] The QP

subproblem is solved using an active set strategy similar to that described in Gill *et al.*[13]

The convergence of the entire framework is governed by two inequalities. If any of the two inequalities at the current point $\mathbf{x}_c$ are true, the algorithm is considered converged:

$$f_{high}(\mathbf{x}_{c_k}) - f_{high}(\mathbf{x}_{c_{k-1}}) < \epsilon_f, \tag{35}$$

$$|\mathbf{x}_{c_k} - \mathbf{x}_{c_{k-1}}| < \epsilon_x, \tag{36}$$

where ϵ_f and ϵ_x are tolerances supplied by the user, k is the current iteration counter, and Δ is the trust region size.

B. First Order Additive Scaling Model

Scaling based methods for variable fidelity optimization were first developed by Chang and Haftka *et al.*[14] They used a multiplicative scaling function to update the value of lower fidelity models to match the higher fidelity models. Alexandrov[6,7] developed an extension of this work by creating an approximation and model management optimization (AMMO) framework that incorporated the first order scaling into a provably convergent methodology. This methodology could be used in various existing optimization routines such as the Augmented Lagrangian Method (ALM), Sequential Quadratic Programming (SQP), or algorithms designed to take into account the coupling of disciplines in multidisciplinary problems. Giunta and Eldred implemented a similar trust region based sequential approximate optimization (SAO) method into the DAKOTA project at the Sandia National Laboratory.[15]

For this morphing aircraft problem an additive method was chosen as opposed to the multiplicative method used by Alexandrov.[6,7] The additive scaling factor was proposed by Lewis and Nash.[16] In the additive method a given set of high and low fidelity models, $f_{high}(\mathbf{x})$ and $f_{low}(\mathbf{x})$, can be matched by adding the low fidelity model to an unknown function $\gamma(\mathbf{x})$. This is expressed mathematically as

$$f_{high}(\mathbf{x}) = f_{low}(\mathbf{x}) + \gamma(\mathbf{x}). \tag{37}$$

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

$$\gamma(\mathbf{x}) = f_{high}(\mathbf{x}) - f_{low}(\mathbf{x}). \tag{38}$$

From Equation 38, 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.

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

$$\gamma(\mathbf{x}_c) = f_{high}(\mathbf{x}_c) - f_{low}(\mathbf{x}_c). \tag{39}$$

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

$$\tilde{\gamma}(\mathbf{x}) = \gamma(\mathbf{x}_c) + \nabla\gamma(\mathbf{x}_c)^T(\mathbf{x} - \mathbf{x}_c). \tag{40}$$

Evaluating this requires gradient information which can be obtained by differentiating Equation 39. This gives

$$\nabla\gamma(\mathbf{x}_c) = \nabla f_{high}|_{\mathbf{x}=\mathbf{x}_c} - \nabla f_{low}|_{\mathbf{x}=\mathbf{x}_c}. \tag{41}$$

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

$$A_c^f(\mathbf{x}) = f_{low}(\mathbf{x}) + \tilde{\gamma}(\mathbf{x}) \approx f_{high}(\mathbf{x}). \tag{42}$$

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.

C. Second Order Additive Scaling Model

A natural extension to the first order additive method presented in the previous section is a second order method. This approach includes the second order terms from the Taylor series expansion of the scaling function. The second order method requires second order information and calculating this using finite difference techniques is computationally prohibitive, especially for the high fidelity model. To combat this problem, two well-known techniques for approximating the second order information from cumulative first order information are used and are briefly presented in Sections D and E. A concurrent and more detailed study of the second order method is to be published by Eldred *et al.*[17]

The additive second order method is found by expanding the Taylor series of γ out to the second order terms around the point $\mathbf{x}_c$:

$$\tilde{\gamma}(\mathbf{x}) = \gamma(\mathbf{x}_c) + (\mathbf{x} - x_c)^T \nabla\gamma(\mathbf{x}_c) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_c)^T \nabla^2\gamma(\mathbf{x}_c)(\mathbf{x} - \mathbf{x}_c). \tag{43}$$

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

$$\nabla^2\gamma(\mathbf{x}_c) = \nabla^2 f_{high}\big|_{\mathbf{x}=\mathbf{x}_c} - \nabla^2 f_{low}\big|_{\mathbf{x}=\mathbf{x}_c}. \tag{44}$$

The second order information is too expensive to evaluate via finite differences. Methods to approximate this information from past first order information are discussed next. Using the second order update schemes provide a means to use the second order scaling method at no more additional function evaluation cost than the first order method. Similar second order information is also required to build scaling functions for the constraints.

D. Damped BFGS

Broyden,[18] Fletcher,[19] Goldfarb,[20] and Shanno[21] 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.[22] This method is defined by the following:

$$r_k = \theta_k y_k + (1 - \theta_k) H_k s_k, \tag{45}$$

where the scalar θ_k is defined as

$$\theta_k = \begin{cases} 1 & : \quad s_k^T y_k \geq 0.2 s_k^T H_k s_k \\ \frac{0.8 s_k^T H_k s_k}{s_k^T H_k s_k - s_k^T y_k} & : \quad s_k^T y_k < 0.2 s_k^T H_k s_k \end{cases}. \tag{46}$$

The Hessian update H_k is

$$H_{k+1} = H_k - \frac{H_k s_k s_k^T H_k}{s_k^T H_k s_k} + \frac{r_k r_k^T}{s_k^T r_k}. \tag{47}$$

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.

E. 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

$$H_{k+1} = H_k + \frac{(y_k - H_k s_k)(y_k - H_k s_k)^T}{(y_k - H_k s_k)^T s_k}. \tag{48}$$

F. Trust Region Model Management

In order to guarantee convergence of the variable fidelity optimization framework, a trust region model management strategy is employed.[23] 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. The approach is named after its limitation in movement of the optimizer within an area where the approximation is thought to be acceptable.

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 x_k^* is found. A trust region ratio, ρ_k is calculated at this new point:

$$\rho_k = \frac{P(x_{c_k})_{high} - P(x_k^*)_{high}}{P(x_{c_k})_{low} - P(x_k^*)_{low}}, \tag{49}$$

where $P()_{high}$ and $P()_{low}$ are the l_1 penalty functions for the high and scaled low fidelity models and the point x_{c_k} was the initial point of the optimization. Notice that by definition $P(x_{c_k})_{low} = P(x_{c_k})_{high}$. 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. Therefore, the ratio is computed using l_1 penalty function:

$$P = f_k + \frac{1}{\mu_k} \sum \max(0, g_i), \tag{50}$$

where μ is the penalty weight which is decreased by a factor of typically ten each time a new point is accepted. This penalty weighting drives all the active constraints to zero as the algorithm converges. The trust region size is governed by the following standard rules:[15,24]

$$\Delta_{k+1} = \begin{cases} c_1 \Delta_k & : \quad \rho_k \leq R_1 \\ \Delta_k & : \quad R_1 < \rho_k < R_2 \\ \tau \Delta_k & : \quad R_2 \leq \rho_k < R_3. \end{cases} \tag{51}$$

where $\tau = c_2$ if $\|x_k^* - x_{c_k}\|_\infty = \Delta_k$ otherwise $\tau = 1$. 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 good. If ρ is near zero the approximation is not a good, but still captures the minimization trend. If ρ is negative, then the point is a worse design. It is rejected, and the trust region size is reduced by the factor c_1.

VII. Variable Fidelity Optimization of the Buckle Wing Concept

This section demonstrates the application of the variable fidelity optimization methods discussed in the previous section. A description of the variable fidelity models used is provided, followed by a brief overview of the problem. Finally the results are given and discussed.

A. Variable Fidelity Suite

Variable fidelity methods have been widely used for engineering design to decrease computation cost.[7,15,24,25] In this paper, two levels of fidelity codes are used to predict the aerodynamic properties of the various wing

shapes and configurations. The high fidelity code used is FUN2D, a full Navier-Stokes coded for unstructured mesh. The lower fidelity model uses a vortex panel method coupled with a boundary layer method. The low fidelity model was an adaptation of the MATLAB code PABLO by Wauquiez.[26]

The FUN2D (Fully Unstructured Navier-Stokes in 2D) code used for high fidelity CFD analysis was developed by NASA at the Langley Research Center.[27,28] The full Navier-Stokes capability of this code was not utilized in this work, due to the extreme computational expense required. Instead the Euler solver was used to decrease the computational expense while still providing a relatively high fidelity model. The grid was produced by the advancing-front/local-reconnection (AFLR) unstructured grid generation method developed by Marcum.[29] The grid was extended to 30 times the chord length in each direction using inviscid wall spacing. On average, the grid consisted of about 33,000 elements. The Mach numbers and Reynolds numbers used were 0.35 and 1.5×10^6, respectively.

The vortex panel method used in the low fidelity model was adapted from the method presented in Kuethe and Chow[30] so that it could handle both single airfoils and multiple airfoils. Similar methods could also be used for more accuracy or different singularities.[31–33] The method uses a polygon representation of the airfoils with m panels. The circulation density, Γ, along each panel varies linearly from each of its corners. No-flux boundary conditions were imposed along the bodies at the center points of each of the panels; these center points are refereed to as the collocation points. In the presence of a uniform flow of strength V_∞ at an angle of attack α, the velocity potential at the ith collocation point is:

$$\phi(x_i, y_i) = V_\infty(x_i \cos(\alpha) + y_i \sin(\alpha)) - \sum_{j=1}^{m} \int \frac{\Gamma(s_j)}{2\pi} \arctan\left(\frac{y_i - y_j}{x_i - x_j}\right) \mathrm{d}s_j \tag{52}$$

where,

$$\Gamma(s_j) = \Gamma_j + (\Gamma_{j+1} - \Gamma_j)\frac{s_j}{S_j}. \tag{53}$$

Here (x_j, y_j) is an arbitrary point in the jth panel, has a length of S_j, and is a distance of s_j from the front edge of that panel. The boundary condition requires that the velocity in the direction of the outward normal, $\hat{n}_i$, is zero at each collocation point; hence,

$$\frac{\partial}{\partial \hat{n}_i}\phi(x_i, y_i) = 0 \tag{54}$$

for each $i = 1, 2, ..., m$. Applying the Kutta condition, which states that the vorticity at the trailing edge of each airfoil is zero and Equation 54, to each airfoil body gives a set of m equations and m unknowns (which are the circulation distribution: $\Gamma_1, ..., \Gamma_m$). Once the circulation distribution is known, the velocities at each collocation point and lift coefficient can be calculated.

The inviscid flow velocities calculated from the vortex panel method are then used in the boundary layer model to predict a drag coefficient. The model consists of Thwaites' equations for laminar regimes, Head's equations for the turbulent areas, and Michel's criterion to find the point of transition between the two regimes. Finally the drag coefficient is found using Squire and Young's formula.[34]

These two fidelity models vary dramatically in the amount of computer resources needed to run a single analysis. The high fidelity code typically takes around 20 minutes per analysis on a Pentium 4 2.4 GHz processor, while the low fidelity code takes on the order of seconds to complete. The high fidelity code is, of course, more accurate and can predict stall and flow separation. However, the low fidelity code does well in predicting the general trends of the aerodynamic characteristics. For these reasons, a variable fidelity approach that makes use of both of these models to help design the buckle wing airfoils and reduce the computational expense is used.

B. Variable Fidelity Optimization and Results

The airfoils for the Buckle Wing were optimized using both a standard nonlinear optimizer and the variable fidelity methods presented in this paper. The problem was first solved, for comparison purposes, using MATLAB's SQP optimizer *fmincon* and all analysis used the high fidelity model, since this is a single fidelity solver. Then the problem was solved using three additive variable fidelity methods: first order, second order using BFGS, and second order using SR1.

The starting point for the optimization was $x = [1\ 0\ 0\ 0.5\ 0\ 0]^T$; this physically means the starting shape was just the first basis airfoil with a cut along the mean camber line. Weights in the objective function used for this problem were $w_1 = 0.01$ and $w_2 = 2$. The convergence criteria for these tests was $\epsilon_f = \epsilon_x = 1 \times 10^{-3}$. The initial design had fused lift and drag coefficients of $c_{l_{fused}} = 0.7754$ and $c_{d_{fused}} = 0.0016$; in the buckled state the lift and drag were, $c_{l_{buc}} = 1.0922$ and $c_{d_{buc}} = 0.0110$. The reason for the smaller drag coefficients is that the high fidelity simulation solves the Euler equations and not the full Navier-Stokes equations. Here the goal is to demonstrate the savings of using the variable fidelity methods, so the extra computational cost of solving the Navier-Stokes equations wasn't necessary.

The results of all four optimizations converged to the same optima of

$$\mathbf{x}^* = [0.6527\ 0.12\ 0.10\ 0.70\ 0.19\ -0.20]^T. \quad (55)$$

The optimal geometry is shown in Figure 5. This design had significantly better aerodynamic properties than did the initial design. In the optimal fused case the fused airfoil had a $c_{l_{fused}} = 0.6794$ and a $c_{d_{fused}} = 0.0012$. In the buckled configuration, it has a $c_{l_{buc}} = 1.0472$ and $c_{d_{buc}} = 0.0090$. The lift to drag ratio for the fused configuration improved significantly while the lift in the buckled state slipped only slightly. The quantity we are most interested in for this test is comparing the number of high fidelity function calls each optimization method needed. Table 2 compares the computational cost in terms of function calls of the SQP and variable fidelity methods.

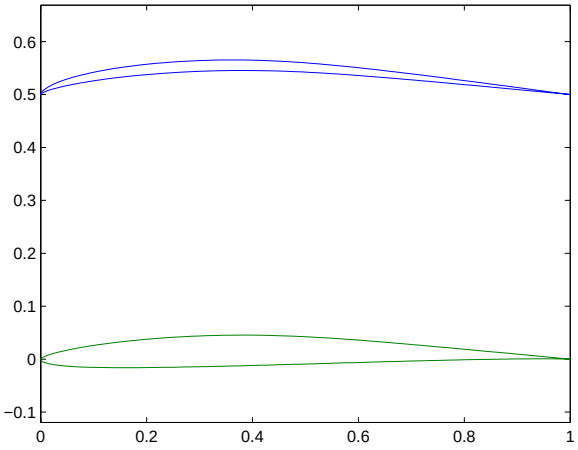

Figure 5. Optimal split airfoil shape produced by variable fidelity optimization.

Table 2. Number of high and low fidelity function calls required for convergence of SQP and variable fidelity methods.

Method	n High Fidelity	n Low Fidelity
SQP	101	-
First Order	43	424
Second Order, BFGS	33	396
Second Order, SR1	34	474

The results in Table 2 show that the variable fidelity optimization framework does reduce the number of high fidelity function evaluations. The SQP optimizer took about one hundred high fidelity simulations, while the first order method reduced this number by over half. Both second order methods were comparable and further reduced the high fidelity evaluations by another 25%. While the amount of computational savings is problem dependent, these results are consistent with past work.[9,25,35] Table 2 also provides the number of low fidelity function calls needed for convergence. This number tends to be fairly high due to the optimization of the scaled low fidelity model during each iteration. The convergence criteria for this inner optimization can effect the number of low fidelity function calls needed, and to a smaller degree, the number of high fidelity

function calls needed. In this problem, the inner optimization tolerances were all $\epsilon = 1 \times 10^{-4}$. This tolerance should be adjusted depending on how expensive the low fidelity model is compared to the high fidelity model.

VIII. Summary and Conclusions

In this paper, the computational challenges associated with a multilevel high fidelity optimization of a morphing aircraft were addressed. The buckle wing morphing UAV concept which has both long range and high maneuverability configurations was used to demonstrate the computational savings tools discussed herein. First an approach to converting these types of problems to single level formulations was discussed. The first order KKT conditions were shown to be equivalent between both formulations. Both the multilevel approach and single level approach were optimized for the Buckle Wing and found to converge to the same solution. To further decrease the computation expense of the single level formulation, a family of variable fidelity optimization methods were used. These variable fidelity methods used both first order and second order approximations to model the error between the high and low fidelity models. The computational time required by the first order method was half that of a standard SQP optimization. The second order methods, which used approximated second order information using BFGS or SR1, both converged using 25% fewer high fidelity function calls than the first order method. This result shows that a great savings can be obtained from reducing the morphing aircraft problem to a single level formulation and using variable fidelity optimization. These methods decrease optimization expense without reducing the design space or converging to different solution.

Acknowledgments

This research effort was supported in part by the following grants and contracts: AFRL / DARPA / Anteon Corporation Contract F33615-98-D-3210, a contract from Sandia National Laboratory, ONR Grant N00014-02-1-0786, and NSF Grant DMI-0114975.

References

[1]Bowman, J., Sanders, B., and Weisshaar, T., "Evaluating the Impact of Morphing Technologies on Aircraft Performance," No. AIAA-2002-1631, 2002.

[2]Gano, S. E., Renaud, J. E., Batill, S. M., and Tovar, A., "Shape Optimization for Conforming Airfoils," *Proceedings of the 44th AIAA/ASME/ASCE/AHS Structures, Structural Dynamics, and Materials Conference*, No. AIAA 2003-1579, Norfolk, VA, 7-10 April 2003.

[3]Gano, S. E., "Sensitivity of Optimal Conforming Airfoils To Exterior Shape," *Proceedings of the AIAA Student Conference 2003*, University of Kentucky at Paducah, 4-5 April 2003.

[4]Vanderplaats, G. N., *Numerical Optimization Techniques for Engineering Design*, Vanderplaats Research & Development, Inc., 3rd ed., 2001.

[5]Rusnell, M. T., Gano, S. E., Pérez, V. M., Renaud, J. E., and Batill, S. M., "Morphing UAV Pareto Curve Shift for Enhanced Performance," *Proceedings of the 45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*, Palm Springs, California, 19-22 April 2004.

[6]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.

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

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

[9]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.

[10]Han, S. P., "A Globally Convergent Method for Nonlinear Programming," *Journal of Optimization Theory and Applications*, Vol. 22, 1977, pp. 297.

[11]Powell, M. J. D., *The Convergence of Variable Metric Methods For Nonlinearly Constrained Optimization Calculations*, Academic Press, 1978.

[12]Powell, M. J. D., "A Fast Algorithm for Nonlineary Constrained Optimization Calculations," *Numerical Analysis, Lecture Notes in Mathematics, Springer Verlag*, Vol. 630, 1978.

[13]Gill, P. E., Murray, W., and Wright, M. H., *Practical Optimization*, Academic Press, London, 1981.
[14]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.
[15]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.
[16]Lewis, R. M. and Nash, S. G., "A Mulitgrid Approach to the Optimization of Systems Governed by Differential Equations," No. AIAA-2000-4890, 2000.
[17]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," *to appear in Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, NY, 30 Aug. - 1 Sept. 2004.
[18]Broyden, C. G., "The Convergence of a Class of Double-rank Minimization Algorithms," *J. Inst. Maths. Applics.*, Vol. 6, 1970, pp. 76–90.
[19]Fletcher, R., "A New Approach to Variable Metric Algorithms," *Computer Journal*, Vol. 13, 1970, pp. 317–322.
[20]Goldfarb, D., "A Family of Variable Metric Updates Derived by Variational Means," *Mathematics of Computing*, Vol. 24, 1970, pp. 23–26.
[21]Shanno, D. F., "Conditioning of Quasi-Newton Methods for Function Minimization," *Mathematics of Computing*, Vol. 24, 1970, pp. 647–656.
[22]Nocedal, J. and Wright, S. J., *Numerical Optimization*, Springer-Verlag, 1999, Springer Series in Operations Reseach.
[23]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.
[24]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.
[25]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.
[26]Wauquiez, C., *Shape Optimization of Low Speed Airfoils using MATLAB and Automatic Differentiation*, Ph.D. thesis, Royal Institute of Techology, Stockholm, Sweeden, 2000, Department of Numerical Analysis and Computing Science.
[27]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.
[28]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.
[29]Marcum, D. L., "Advancing-Front/Local-Reconnection (AFLR) Unstructured Grid Generation," *Computational Fluid Dynamics Review*, 1998.
[30]Kuethe, A. M. and Chow, C. Y., *Foundations of Aerodynamics Bases pf Aerodynamic Design*, John Wiley & Sons Inc., 5th ed., 1998.
[31]Katz, J. and Plotkin, A., *Low-Speed Aerodynamics, 2nd Edition*, Cambridge University Press, 2001.
[32]Moran, J., *Theoretical and Computational Aerodynamics*, John Wiley & Sons Inc., 1984.
[33]Henne, P. A., *Applied Computational Aerodynamics*, American Institute of Aeronautics and Astronautics, 1990, Progess in Astronautics and Aeronautics Series Vol. 125.
[34]Squire, H. B. and Young, A. D., *The calculation of the profile drag of aerofoils*, ARC RM, 1938.
[35]Torczon, V. and Trosset, M. W., "Using approximations to accelerate engineering design optimization," *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, No. AIAA 98-4800, St. Louis, Missouri, 2-4 September 1998.