

Approximation and Model Management in Aerodynamic Optimization with Variable-Fidelity Models

Natalia M. Alexandrov*

NASA Langley Research Center, Hampton, Virginia 23681-2199

Robert Michael Lewis†

College of William and Mary, Williamsburg, Virginia 23187-8795

and

Clyde R. Gumbert,‡ Lawrence L. Green,§ and Perry A. Newman¶

NASA Langley Research Center, Hampton, Virginia 23681-2199

This work discusses an approach, first-order approximation and model management optimization (AMMO), for solving design optimization problems that involve computationally expensive simulations. AMMO maximizes the use of lower-fidelity, cheaper models in iterative procedures with occasional, but systematic, recourse to higher-fidelity, more expensive models for monitoring the progress of design optimization. A distinctive feature of the approach is that it is globally convergent to a solution of the original, high-fidelity problem. Variants of AMMO based on three nonlinear programming algorithms are demonstrated on a three-dimensional aerodynamic wing optimization problem and a two-dimensional airfoil optimization problem. Euler analysis on meshes of varying degrees of refinement provides a suite of variable-fidelity models. Preliminary results indicate threefold savings in terms of high-fidelity analyses for the three-dimensional problem and twofold savings for the two-dimensional problem.

Nomenclature

C_D	=	drag coefficient
C_L	=	lift coefficient
C_I	=	rolling moment coefficient
C_M	=	pitching moment coefficient
c_E	=	equality constraints
c_I	=	inequality constraints
f	=	objective function
M_∞	=	freestream Mach number
S	=	semispan wing planform area
x, x_L, x_U	=	design variables and bounds
α	=	angle of attack
Δ	=	trust-region radius

Introduction

WE describe a general approach to design optimization, the first-order approximation and model management optimization (AMMO) framework, that integrates engineering and

physical modeling concepts with mathematically rigorous nonlinear programming techniques. AMMO uses a range of simulations in a systematic way that guarantees convergence to high-fidelity optimal designs without the expense of relying exclusively on high-fidelity models or simulations.

A few words are in order to place AMMO in relation to other work. Great progress has been made in the ability to simulate the behavior of physical and engineering systems accurately. However, the enormous computational cost of repeated high-fidelity simulations, such as the Navier–Stokes equations or those based on fine computational meshes, makes it impractical to rely exclusively on high-fidelity models for the purpose of design optimization.

To address this difficulty, designers have combined the use of high-fidelity and low-fidelity models for a long time, see, e.g., Schmit et al.^{1–3} Barthelemy and Haftka⁴ survey the use of approximations in structural optimization. Recent overviews of models for aerodynamic analysis and optimization can be found, for example, in Jameson⁵ and Newman et al.⁶

Approaches to engineering design optimization that use variable-fidelity models are sometimes called sequential approximate optimization (SAO).¹ Although practically every optimization method can be called sequential and approximate, the term SAO is usually reserved for methods that replace the objective function and constraints of the design problem by low-fidelity models. A low-fidelity model can be a simplified physics model, a single numerical model evaluated on a relatively coarse mesh, a single numerical model converged to a varying degree of accuracy, one of a variety of response surfaces, or one of a variety of reduced-order models. An SAO method minimizes the low-fidelity model. Some SAO algorithms attempt to create the best possible low-fidelity model and optimize it only once, whereas others update the models during optimization. Haftka and Gürdal⁷ discuss several SAO techniques. SAO procedures have been largely based on heuristics, and convergence to a solution of the high-fidelity optimal design problem has not been guaranteed, in general. The mathematical optimization community, on the other hand, has focused on provably convergent algorithms, but the models used in those algorithms have been assumed to be based on local Taylor-series approximations, as a rule.

Combining the two perspectives, AMMO^{8–10} is a general, mathematically rigorous, globally convergent methodology that can be applied to any derivative-based optimization algorithm to alleviate the

Received 5 April 2000; revision received 12 June 2001; accepted for publication 8 August 2001. Copyright © 2001 by the American Institute of Aeronautics and Astronautics, Inc. No copyright is asserted in the United States under Title 17, U.S. Code. The U.S. Government has a royalty-free license to exercise all rights under the copyright claimed herein for Governmental purposes. All other rights are reserved by the copyright owner. Copies of this paper may be made for personal or internal use, on condition that the copier pay the \$10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0021-8669/01 \$10.00 in correspondence with the CCC.

*Research Scientist, Multidisciplinary Optimization Branch, Aerospace Systems Concepts and Analysis, M/S 159; n.alexandrov@larc.nasa.gov. Member AIAA.

†Assistant Professor, Department of Mathematics, P.O. Box 8795; buckaroo@math.wm.edu. Member AIAA.

‡Research Engineer, Multidisciplinary Optimization Branch, Aerospace Systems Concepts and Analysis, M/S 159; c.r.gumbert@larc.nasa.gov.

§Research Engineer, Multidisciplinary Optimization Branch, Aerospace Systems Concepts and Analysis, M/S 159; l.l.green@larc.nasa.gov. Senior Member AIAA.

¶Senior Research Scientist, Multidisciplinary Optimization Branch, Aerospace Systems Concepts and Analysis, M/S 159; p.a.newman@larc.nasa.gov.

expense of design optimization with simulations. The approach integrates the convergent techniques of nonlinear programming with the use of variable-fidelity models available in engineering disciplines. We work with first-order (i.e., derivative-based) optimization methods because they are generally more efficient and can handle larger numbers of design variables and a broader range of models than methods that do not rely on derivatives.

In this paper we describe the idea that underlies first-order AMMO and give three specific examples of adapting nonlinear programming algorithms in the AMMO framework. Computational demonstrations follow. The paper concludes with lessons learned and open questions under investigation.

First-Order AMMO Methodology

In this work the design optimization problem is represented by a nonlinear program of the form

$$\begin{aligned} & \underset{x}{\text{minimize}} && f(x) \\ & \text{subject to} && c_E(x) = 0 \\ & && c_I(x) \geq 0 \\ & && x_L \leq x \leq x_U \end{aligned} \tag{1}$$

where the evaluation of the objective function and constraints involves a high-fidelity simulation or, for a multidisciplinary problem, a set of coupled simulations, with each analysis a particular aspect of the physical system or the behavior of a subsystem. Some constraints can involve physical states (responses) of the system, whereas others can be algebraic or purely geometrical.

To solve Eq. (1), AMMO relies on the trust-region approach¹¹ in nonlinear programming to ensure robust behavior. Conventional derivative-based nonlinear programming algorithms, including trust-region methods, solve a sequence of subproblems, each of which operates on local first- or second-order Taylor series, with various approximations to the first and second derivatives of the contributing functions. The information exchange between the analysis and the optimizer is depicted at the top of Fig. 1.

If evaluating the functions and derivatives involves a simulation of high accuracy but high computational cost (e.g., the Navier-Stokes equations), the repeated consultations with the analysis required by the optimizer are expensive.

In AMMO we expand the idea of a local model by replacing the Taylor series in the subproblems with general models that have local trends that are similar to those obtained with high-fidelity analyses. AMMO builds models for the sequence of optimization subproblems using high-fidelity and low-fidelity information. The models are constructed so that their trends are similar locally to the trends in the high-fidelity model. This is accomplished by requiring that the models in the optimization subproblems be consistent to first order with the high-fidelity model, as follows.

Let \tilde{f} , \tilde{c}_E , and \tilde{c}_I be low-fidelity models of f , c_E , and c_I , respectively. At each iteration x_k of an AMMO algorithm, the low-fidelity

models are required to satisfy first-order consistency with the high-fidelity counterparts, i.e.,

$$\begin{aligned} \tilde{f}(x_k) &= f(x_k) & \nabla \tilde{f}(x_k) &= \nabla f(x_k) \\ \tilde{c}_E(x_k) &= c_E(x_k) & \nabla \tilde{c}_E(x_k) &= \nabla c_E(x_k) \\ \tilde{c}_I(x_k) &= c_I(x_k) & \nabla \tilde{c}_I(x_k) &= \nabla c_I(x_k) \end{aligned} \tag{2}$$

Higher-order consistency conditions can be imposed for problems with available higher-order derivatives.

Conditions (2) ensure that \tilde{f} , \tilde{c}_E , and \tilde{c}_I mimic the local behavior of first-order Taylor series approximations of f , c_E , and c_I around the current design x_k . First-order consistency is easily obtained in practice. The work reported here uses a technique we call the β -correction, due to Chang et al.¹² Given a high-fidelity function ϕ_{hi} (say, f) and any low-fidelity model ϕ_{lo} of ϕ_{hi} , we correct ϕ_{lo} as follows. Define

$$\beta(x) = \frac{\phi_{hi}(x)}{\phi_{lo}(x)}$$

and construct the linear approximation

$$\beta_k(x) = \beta(x_k) + \nabla \beta(x_k)^T (x - x_k)$$

Then

$$\tilde{\phi}(x) = \beta_k(x) \phi_{lo}(x)$$

satisfies the consistency conditions (2). Other simple correction schemes are available to enforce consistency.

Optimization subproblems in the AMMO framework, depicted at the bottom of Fig. 1, operate on corrected low-fidelity models. Expensive, high-fidelity computations serve to recalibrate the low-fidelity models occasionally, based on a set of systematic criteria, to obtain \tilde{f} , \tilde{c}_E , and \tilde{c}_I . The salient features of AMMO can be summarized as follows:

1) Although a low-fidelity model may not capture a particular feature of the physical phenomenon to the same degree of accuracy (or at all) as its high-fidelity counterpart, a low-fidelity model may still have satisfactory predictive properties for the purposes of finding a good direction of design improvement. Locally, imposing the first-order consistency (2) ensures this property.

2) AMMO replaces the local Taylor series of conventional optimization by general nonlinear models required to satisfy the consistency conditions (2). In principle, AMMO is capable of handling arbitrary models, provided the easily imposed consistency conditions are satisfied.

3) AMMO is based on the trust-region approach, which can be described as an adaptive move limit strategy for improving the global behavior of optimization algorithms based on local models. The trust-region methodology ensures the convergence of the AMMO scheme to a solution of the high-fidelity problem¹³ by providing a measure of the low-fidelity model's predictive behavior, a criterion for updating the model, and a systematic response to situations in which an optimization phase performed using a low-fidelity model gives either an incorrect or a poor prediction of the high-fidelity model's actual behavior.

Practical efficiency of any particular AMMO scheme depends on the predictive qualities of the corrected low-fidelity models for the purposes of optimization, which, in turn, are problem dependent.

AMMO Under Study

The first-order AMMO approach can be used in conjunction with any gradient-based optimization algorithm and any suite of variable-fidelity models. In the remainder of the paper, we describe specific instances of first-order AMMO based on three nonlinear programming algorithms. This discussion will give a prospective user an idea of how to adapt a particular nonlinear programming technique to the AMMO framework.

The three algorithms under study follow the trust-region scheme. Each algorithm solves a sequence of optimization subproblems that operate on models of the objective function and constraints within a trust region where the model trends are thought to approximate

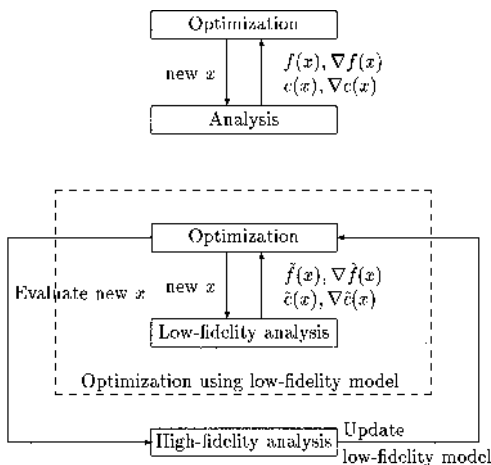


Fig. 1 Conventional optimization vs AMMO.

the function trends adequately for finding a step towards a solution. Once such a trial step is computed according to a specific algorithm, it is evaluated by comparing the actual improvement in the merit function of the problem with the improvement predicted by the model of the merit function. The trial step is then either accepted or rejected, and the trust region is updated, based on the comparative performance of the model.

All subproblems are solved approximately. ‘‘Approximately’’ means that the resulting step should predict sufficient decrease in the merit function or its components. Roughly speaking, global convergence analysis requires a very mild sufficient decrease condition—the step must predict at least a fraction of the decrease a linear Taylor-series model would predict within a given trust region. All algorithms of interest for solving trust-region subproblems satisfy this requirement automatically.

Augmented Lagrangian AMMO

The augmented Lagrangian method for constrained optimization allows for an immediate extension of the unconstrained AMMO⁸ to constrained problems. The underlying algorithm is the augmented Lagrangian approach of Conn et al.¹¹

In this method the explicit nonlinear inequality constraints of problem (1) are converted to equalities by introducing a set of non-negative slack variables z to define the equality constraints

$$c(x, z) = \begin{bmatrix} c_E(x) \\ c_I(x) - z \end{bmatrix} = 0$$

Denoting (x, z) by y , we obtain the following reformulation of Eq. (1):

$$\begin{aligned} & \underset{y}{\text{minimize}} && f(y) \\ & \text{subject to} && c(y) = 0 \\ & && y_L \leq y \leq y_U \end{aligned} \quad (3)$$

where $y_L = (x_L, 0)$ and $y_U = (x_U, \infty)$. The augmented Lagrangian associated with this problem is

$$L(y; \lambda, \mu) = f(y) + \lambda^T c(y) + (1/2\mu)\|c(y)\|_2^2$$

where λ is the vector of Lagrange multipliers and $\mu > 0$ is the penalty parameter. The bound constraints are treated explicitly. For appropriate values of μ and λ , minimization of L solves Eqs. (3). However, because the appropriate values of μ and λ are not known a priori an iterative approach is devised that solves an augmented Lagrangian subproblem while updating μ and λ .

Let P be the projection onto the set $B = \{y \mid y_L \leq y \leq y_U\}$. Given $y \in B$ and v , define $P(y, v) = y - P(y - v)$. The augmented Lagrangian approach is summarized as follows:

Algorithm 1: Augmented Lagrangian method.

Initialization. Set $k = 1$. Select y_1 , the initial penalty parameter $\mu_1 < 1$, and the initial convergence criteria ω_1 and η_1 . Specify the least allowable decrease $0 < \tau < 1$ in the penalty parameter.

Step 1: Subproblem. Approximately solve

$$\begin{aligned} & \underset{y}{\text{minimize}} && L(y; \lambda_k, \mu_k) \\ & \text{subject to} && y_L \leq y \leq y_U \end{aligned}$$

to find y_k satisfying $\|P[y_k, \nabla L(y_k; \lambda_k, \mu_k)]\| \leq \omega_k$. If $\|c(x_k)\| \leq \eta_k$, go to 2. Otherwise go to 3.

Step 2: Lagrange multiplier update. Update Lagrange multipliers with any standard update formula, e.g., the Hestenes–Powell update $\lambda_{k+1} = \lambda_k + c(x_k)/\mu_k$. Choose $\mu_{k+1} \leq \mu_k$, $\omega_{k+1} > 0$, $\eta_{k+1} > 0$, so that $\lim_{k \rightarrow \infty} \omega_k = 0$ and $\lim_{k \rightarrow \infty} \eta_k = 0$. Choose y_{k+1} . Set $k = k + 1$. Go to 1.

Step 3: Reduce the penalty significantly. Set $\lambda_{k+1} = \lambda_k$, $\mu_{k+1} \leq \tau \mu_k$. Update ω_{k+1} , η_{k+1} as in 2. Set $k = k + 1$. Go to 1. For further details, see Conn et al.¹¹

Typically, the subproblem in step 1 is solved by conventional unconstrained trust-region techniques. In the AMMO adaptation of this algorithm, we solve the subproblem using \tilde{L} , an approximation to L , based on low-fidelity models of the objective and constraints in Eq. (3), as follows.

Algorithm 2: AMMO solution of step 1.

Choose $\Delta_1 > 0$, constants $\Delta^* > 0$, $0 < r_1 < r_2 < 1$, and $0 < c_1 < 1 < c_2$.

For $j = 1, \dots$, until $\|P[y_k, \nabla L(y_k; \lambda_k, \mu_k)]\| \leq \omega_k$

Compute L and ∇L at y_k .

Select a model \tilde{L}_k of L , with

$$\tilde{L}_k(y_k; \lambda_k, \mu_k) = L(y_k; \lambda_k, \mu_k)$$

$$\nabla \tilde{L}_k(y_k; \lambda_k, \mu_k) = \nabla L(y_k; \lambda_k, \mu_k)$$

Solve approximately to obtain s_j :

$$\begin{aligned} & \underset{s}{\text{minimize}} && \tilde{L}_k(y_k + s; \lambda_k, \mu_k) \\ & \text{subject to} && y_L \leq y_k + s \leq y_U \\ & && \|s\|_\infty \leq \Delta_j \end{aligned}$$

Compute

$$r = \frac{L(y_k; \lambda_k, \mu_k) - L(y_k + s_j, \lambda_k; \mu_k)}{L(y_k; \lambda_k, \mu_k) - \tilde{L}_k(y_k + s_j, \lambda_k; \mu_k)}$$

Evaluate new step:

If $L(y_k + s_j) < L(y_k)$, then $y_k \leftarrow y_k + s_j$;

else $y_k \leftarrow y_k$

Update trust-region:

If $r < r_1$, then $\Delta_{j+1} = c_1 \|s_j\|$;

else if $r > r_2$, then $\Delta_{j+1} = \min\{c_2 \|\Delta_j\|, \Delta^*\}$;

else $\Delta_{j+1} = \Delta_j$.

End for.

Typical values of the constants are $r_1 = 0.1$, $r_2 = 0.75$, $c_1 = 0.5$, and $c_2 = 2$. The trust-region radius constraint uses the ℓ_∞ norm because it conforms naturally to bound constraints on the design variables. Other norms, such as the ℓ_2 norm, are also frequently used in trust-region subproblems.

The augmented Lagrangian AMMO is relatively easy to implement and can be proven to converge reliably under reasonable assumptions.¹³ The expected difficulties are those of the underlying optimization approach: augmented Lagrangian methods can converge slowly, and they are subject to ill-conditioning as μ approaches 0.

MAESTRO-AMMO

The second AMMO under study is based on a class of trust-region multilevel algorithms for large-scale constrained optimization (MAESTRO).¹⁴ The present version of MAESTRO deals with problem (1) by converting the explicit inequalities into equalities via squared slack variables z :

$$c(x, z) = \begin{bmatrix} c_E(x) \\ c_I(x) - z^2 \end{bmatrix} = 0$$

Denoting (x, z) by y , we again obtain Eq. (3), with the lower bound constraints now defined as $y_L = (x_L, -\infty)$ because the nonnegativity of the slack z need not be maintained.

Optimization steps in the basic MAESTRO approach are sums of substeps, each of which is a minimizer of a subproblem designed to improve a part of the total problem, e.g., a block of the constraints, while preserving the predicted improvement already obtained in other parts of the problem. Each subproblem is solved within its own trust region. The total step is evaluated by considering the actual and predicted reductions in the merit function, as in algorithm 2. The augmented Lagrangian and the ℓ_2 penalty function are suitable merit functions. Here we use the ℓ_2 penalty function

$$\mathcal{P}(y; \mu) = f(y) + \mu \|c(y)\|_2^2$$

where $\mu \geq 1$ is the penalty parameter. The corresponding low-fidelity model of \mathcal{P} is

$$\tilde{\mathcal{P}}(y; \mu) = \tilde{f}(y) + \mu \|\tilde{c}(y)\|_2^2$$

Because the current demonstrations are single-discipline design problems with a small number of constraints, the following brief description of MAESTRO-AMMO is given for a single block of constraints. The version for multidisciplinary optimization or multiple blocks of constraints can be found elsewhere.⁹

Algorithm 3: MAESTRO-AMMO.

Initialization. Choose y_1 , Δ_1^f , Δ_1^c ; set $\mu_1 = 1$.

Do $k = 1, \dots$, until convergence:

Subproblem 1: Improve constraints.

Compute c and ∇c at y_k .

Select a model \tilde{c}_k of c , with

$$\tilde{c}_k(y_k) = c(y_k)$$

$$\nabla \tilde{c}_k(y_k) = \nabla c(y_k)$$

Solve approximately to obtain s_k^c :

$$\begin{aligned} & \underset{s}{\text{minimize}} && \tilde{c}_k(y_k + s) \\ & \text{subject to} && y_L \leq y_k + s \leq y_U \\ & && \|s\|_\infty \leq \Delta_k^c \end{aligned}$$

Subproblem 2: Improve objective.

Compute f and ∇f at $y_k + s_k^c$.

Select a model \tilde{f}_k of f , with

$$\tilde{f}_k(y_k + s_k^c) = f(y_k + s_k^c)$$

$$\nabla \tilde{f}_k(y_k + s_k^c) = \nabla f(y_k + s_k^c)$$

Solve approximately to obtain s_k^f :

$$\begin{aligned} & \underset{s}{\text{minimize}} && \tilde{f}_k(y_k + s_k^c + s) \\ & \text{subject to} && \nabla \tilde{c}_k(y_k)^T s = 0 \\ & && y_L \leq y_k + s_k^c + s \leq y_U \\ & && \|s\|_\infty \leq \Delta_k^f \end{aligned}$$

Set $s_k = s_k^c + s_k^f$.

Compute

$$r = \frac{\mathcal{P}(y_k; \mu_k) - \mathcal{P}(y_k + s_k; \mu_k)}{\mathcal{P}(y_k; \mu_k) - \tilde{\mathcal{P}}_k(y_k + s_k; \mu_k)}$$

Update the penalty parameter μ_k :

Increase μ_k , if necessary, so that the predicted

reduction $\mathcal{P}(y_k; \mu_k) - \tilde{\mathcal{P}}_k(y_k + s_k; \mu_k) > 0$.

Evaluate step s_k :

If $\mathcal{P}(y_k + s_k; \mu_k) < \mathcal{P}(y_k; \mu_k)$,

then $y_k \leftarrow y_k + s_k$;

else $y_k \leftarrow y_k$.

Update Δ_k^c and Δ_k^f as in algorithm 2.

End do.

In the basic MAESTRO approach, subproblems 1 and 2 are solved directly with Taylor-series models of c and f . The AMMO version replaces them with low-fidelity counterparts \tilde{c} and \tilde{f} that satisfy first-order consistency (2). Subproblems 1 and 2 are now solved iteratively by conventional methods. MAESTRO-AMMO shares the global convergence properties of the underlying algorithm.

Implementing MAESTRO-AMMO is more laborious than the augmented Lagrangian AMMO. The benefits are the expected greater efficiency and its natural capability for multidisciplinary optimization problems with arbitrary couplings.

SQP-AMMO

The sequential quadratic programming (SQP) approach forms a popular class of nonlinear programming methods¹⁵. The SQP-AMMO shares the global convergence properties of the underlying SQP approach.

Let $\Phi(x; \mu)$ be a merit function for the high-fidelity problem. In the work described here Φ is the l_1 penalty function

$$\Phi(x; \mu) = f(x) + \mu \sum_{i \in E} |c_{E,i}(x)| + \mu \sum_{i \in I} \max[0, -c_{I,i}(x)]$$

where E and I are the index sets of the equality and inequality constraints, respectively. Other choices of the merit function are possible.¹⁶ SQP-AMMO models the merit function by

$$\tilde{\Phi}(x; \mu) = \tilde{f}(x) + \mu \sum_{i \in E} |\tilde{c}_{E,i}(x)| + \mu \sum_{i \in I} \max[0, -\tilde{c}_{I,i}(x)]$$

and the following algorithm results:

Algorithm 4: SQP-AMMO.

Initialization. Choose x_1 , μ_1 .

Do $k = 1, \dots$, until convergence:

Select models \tilde{c}_I , \tilde{c}_E , and \tilde{f} satisfying consistency (2).

Solve approximately for $s = x - x_k$:

$$\begin{aligned} & \underset{s}{\text{minimize}} && \tilde{f}(x_k + s) \\ & \text{subject to} && \tilde{c}_I(x_k) + \nabla \tilde{c}_I(x_k)^T s \leq 0 \\ & && \tilde{c}_E(x_k) + \nabla \tilde{c}_E(x_k)^T s = 0 \\ & && x_L \leq x \leq x_U \\ & && \|s\|_\infty \leq \Delta_k \end{aligned}$$

Compute

$$r = \frac{\Phi(x_k; \mu_k) - \Phi(x_k + s_k; \mu_k)}{\Phi(x_k; \mu_k) - \tilde{\Phi}_k(x_k + s_k; \mu_k)}$$

Update Δ_k , x_k based on r , as in algorithm 2.

End do.

The penalty parameter μ_k must be greater than the smallest Lagrange multiplier associated with Eqs. (1) It is usually estimated at the beginning of optimization and updated only if necessary. If the low-fidelity models are Taylor series, then SQP-AMMO reduces to conventional SQP.

SQP-AMMO has a number of benefits. It is relatively easy to implement, and it converges very rapidly once it is near a solution. It handles the inequality constraints directly and enjoys the efficiency of SQP methods. By choosing Δ_k sufficiently large, the first iteration yields solution of the low-fidelity optimization problem. This feature must be obtained by preprocessing in the other approaches. SQP-AMMO also allows for an easy incorporation of commercial software.

Computational Demonstrations

The computational demonstrations are intended to validate the concept of AMMO. The ability to maximize the use of low-fidelity, cheaper models, and thereby reduce the overall computational cost, will depend on the predictive qualities of the low-fidelity models. Even though the low-fidelity models may not be good approximators of the high-fidelity models for the purposes of analysis, they may possess suitable predictive properties for the purposes of optimization.

The computational tests include both the case when the relationship between the various levels of models is favorable and the case when it is not. The relationship is favorable when the low-fidelity models can provide a long sequence of steps with satisfactory directions of improvement for the high-fidelity merit function before the low-fidelity model has to be re-calibrated. The relationship is not favorable when the low-fidelity models do not satisfactorily capture the trends in the high-fidelity models on a significant portion of the feasible region.

AMMO approaches could suffer from an overreliance on the low-fidelity model if it does not adequately reflect the behavior of the higher-fidelity model adequately in a large region. In this case AMMO might take only a few steps using the low-fidelity information before having to resort to recalibrating the model. Thus, in the

worst case AMMO reverts to conventional optimization with the high-fidelity models.

The tests described in this paper investigate variable-resolution modeling—that in which a single type of analysis, performed on a variety of related meshes, provides variable-fidelity models. In this case the finer the mesh, the higher the model fidelity (presumably) and the higher the computational expense. AMMO with variable-fidelity physics models is described elsewhere.¹⁰

The initial experiments are conducted only with two design variables in order to visualize the progress of the algorithms easily and completely. For the purposes of understanding the problem, we generated enough data to construct graphical level sets of the objective and constraints; however, this information is not used (nor is it necessary) for any of the solution schemes.

The problems are first solved in single-fidelity mode using conventional optimization methods, such as NPSOL¹⁷ and PORT,¹⁸ to obtain a baseline number of function evaluations or iterations to find an optimum (The use of names of commercial software in this paper is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products by NASA or ICASE.) The problems are then solved with AMMO adaptations of the conventional methods. We terminate the optimization algorithms when the norm of the projected gradient of the objective falls below 10^{-5} .

In our computational study we use the following high-fidelity models: 1) Euler computational fluid dynamics (CFD) analysis on a relatively fine mesh; 2) a synthetic analysis constructed from objective and constraint values from the high-fidelity Euler CFD analysis and two-dimensional, uniform, variation diminishing splines using PORT¹⁸; 3) a synthetic analysis constructed from objective and constraint values from the high-fidelity Euler CFD analysis and kriging¹⁹; and 4) a synthetic analysis constructed from objective and constraint values from the high-fidelity Euler CFD analysis and cubic polynomials, using the RSG software.²⁰ The low-fidelity models are obtained in a similar fashion, using Euler CFD analysis on a coarser mesh.

The synthetic analyses serve two purposes. First, they reduce the computational cost of experimentation. Second, they allow us to study the situation where the uncorrected low-fidelity model does not capture the high-fidelity trends very well. In particular, graphics will show that for the problems under study the objective and constraints obtained from the low-fidelity Euler CFD analysis capture the trends of the high-fidelity problem well. This is a most favorable situation for AMMO. Some of the synthetic analyses, on the other hand, allow us to investigate the adverse situation.

For all three AMMO approaches the consistency conditions were enforced via the β -correction technique, which was found to provide an excellent correction strategy.

Performance is evaluated in terms of the absolute number of calls to the high- and low-fidelity function and sensitivity calculations and the number of “equivalent” high-fidelity computations. The latter are easily obtained because the CFD analysis codes use multigrid techniques, where this metric is commonly computed.

Finally, a conscious effort was made to implement AMMO in a straightforward manner, without any fine tuning, in order to obtain a proof of concept. As will be discussed later, significant improvements in efficiency can likely be made.

Three-Dimensional Wing Problem

Optimization Problem

The first demonstration problem is a three-dimensional aerodynamic wing optimization. The wing consists of a single trapezoidal panel with a rounded tip. It is parameterized by 15 variables: five of which describe the planform, five of which describe the root section shape, and five of which describe the tip section shape. The wing and some of the associated parameters are depicted in Fig. 2. The two design variables are the tip chord and the tip trailing-edge setback. The objective function $f(x)$ is $-C_L/C_D$. Several artificial constraints are imposed in lieu of multidisciplinary constraints: 1) a lower bound on total lift $C_L \times S$, in lieu of a minimum payload requirement; 2) an upper bound on C_M , in lieu of a trim constraint;

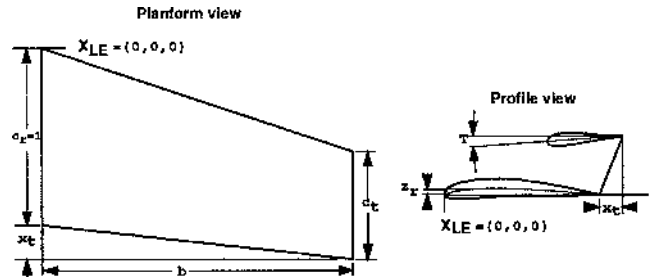


Fig. 2 Three-dimensional wing problem.

and 3) an upper bound on C_I , in lieu of a maximum bending moment. Geometric constraints ensure a minimum leading-edgeradius and a minimum thickness.

For the results we report here, $M_\infty = 0.5$, and $\alpha = 3$ deg. Given the subsonic speed of the flow, the drag is primarily the induced drag caused by lift, $C_D \propto C_L^2$, and so our objective is effectively $-1/C_L$.

The aerodynamic analysis code used for this study is CFL3D.ADII,²¹ a version of CFL3D²² obtained via automatic differentiation. The surface geometry was computed using RAPID.²³ The volume mesh and associated gradients needed for CFL3D are computed using a version of CSCMDO²⁴ generated by automatic differentiation.

The CFD analysis is performed on two meshes: 1) $97 \times 25 \times 17$ (low fidelity), and 2) $193 \times 49 \times 33$ (high fidelity). Because the analysis uses a multigrid solution process, the CPU time per analysis is essentially linear in the number of grid points, resulting in an eight-fold difference in execution time between adjacent levels of fidelity. On an Ultra 1 Sun workstation, a single CFD analysis on the $97 \times 25 \times 17$ mesh takes eight minutes, and the $193 \times 49 \times 33$ mesh analysis takes about an hour. The analysis residuals are converged to 10^{-6} . Sensitivity calculations for the objective and constraints take roughly $6\frac{1}{2}$ times as long as the analysis. The sensitivity analysis residuals are converged to 10^{-3} .

Numerical Results

Figure 3 depicts the level sets of the objective functions and active constraints obtained by performing analyses on the $193 \times 49 \times 33$ and $97 \times 25 \times 17$ meshes. The shaded regions are infeasible. The constraint C_I is inactive at the solution and is not depicted. Solutions are marked by black squares. This problem has a favorable structure for AMMO: although the optima are at different locations, the low-fidelity and high-fidelity objectives and constraints have similar trends.

For MAESTRO-AMMO testing was done with function values obtained directly from CFL3D.ADII. The analysis count was as follows. To obtain a solution on the low-fidelity mesh alone, using conventional MAESTRO required 17 function and 17 sensitivity calls. Solution with the high-fidelity mesh alone was attempted but not completed because of the expense of direct function and derivative evaluations. Given the similarity of the level sets of the objective and constraints associated with the two meshes, we assume that conventional optimization on the high-fidelity mesh would take a similar number of iterations as that on the low-fidelity mesh. MAESTRO-AMMO required 18 low-fidelity functions, 18 low-fidelity sensitivities, seven high-fidelity functions, and seven high-fidelity sensitivities, for a total of $7 + \frac{18}{8} = 9\frac{1}{4}$ equivalent high-fidelity functions and as many sensitivities. Thus, the increase in efficiency is approximately two-fold.

Figures 4–6 show the resulting level sets for the objective and active constraints obtained from the synthetic analyses based on the same CFL3D.ADII data used to generate Fig. 3. The low-fidelity synthetic polynomial analysis is not a good approximation to the high-fidelity synthetic polynomial analysis, as Fig. 6 demonstrates. Thus, the synthetic spline and kriging analyses manifest the situation in which the relationship between the high- and low-fidelity approximations is favorable, whereas the synthetic polynomial analysis, the situation when the relationship is not as favorable.

The augmented Lagrangian AMMO was tested with a synthetic kriging analysis. The conventional augmented Lagrangian

Table 1 Wing optimization problem: summary of improvements as a result of AMMO

Algorithm	Improvement
Augmented Lagrangian	3.0/2.6 (kriging)
SQP	2.8/2.8 (spline)
SQP	3.0/3.0 (polynomial)
MAESTRO	1.9/1.9 (CFD)

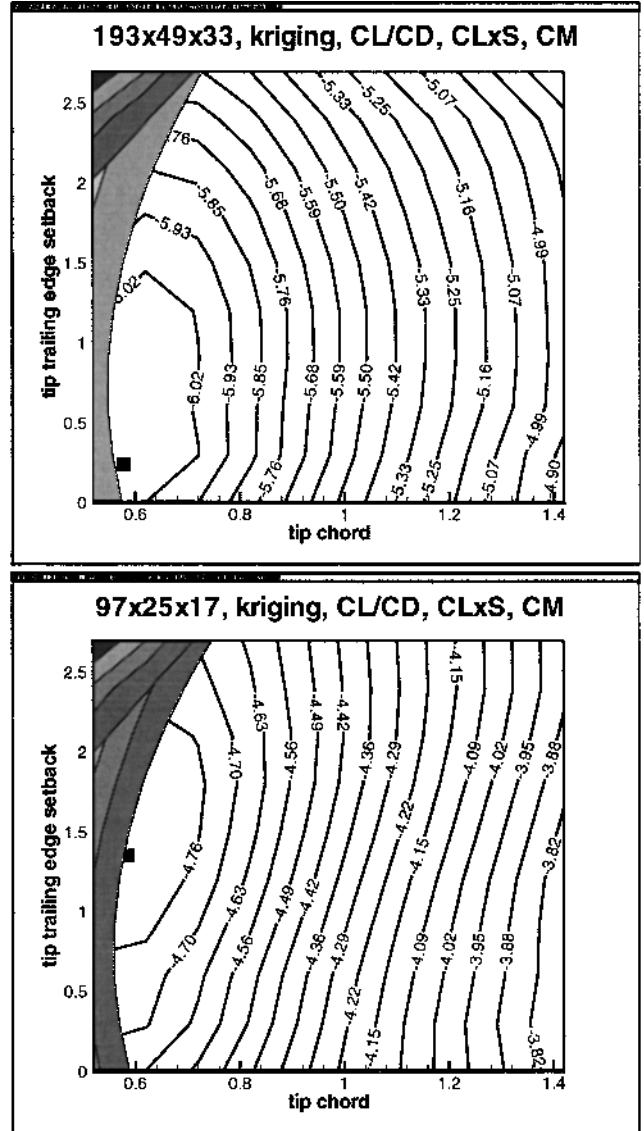
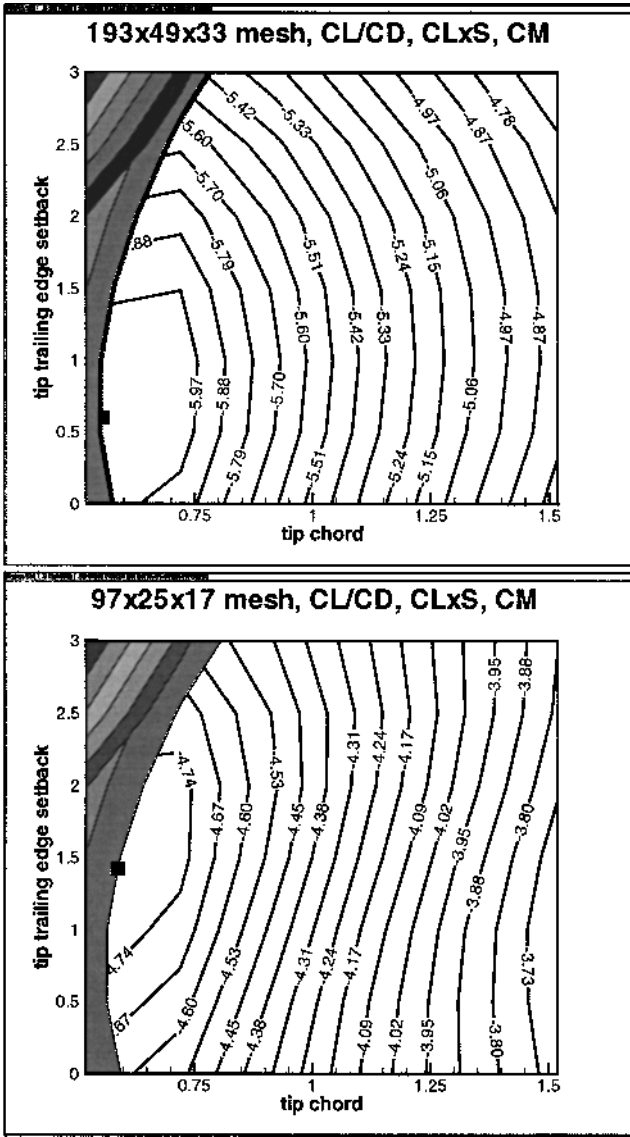


Fig. 3 High-fidelity vs low-fidelity objectives and active constraints: level sets from CFD analysis.

Fig. 4 High-fidelity vs low-fidelity objectives and active constraints: level sets from synthetic kriging analysis.

algorithm required 37 evaluations of the high-fidelity objective and constraints and 27 evaluations of the high-fidelity objective and constraint sensitivities. The augmented Lagrangian AMMO required six evaluations of the high-fidelity objective and constraints, six evaluations of the high-fidelity objective and constraint sensitivities, 51 evaluations of the low-fidelity objective and constraints, and 36 evaluations of the low-fidelity objective and constraint sensitivities. Because the low-fidelity analyses take $\frac{1}{8}$ of the time of the high-fidelity analyses, the augmented Lagrangian required the equivalent work of $6 + \frac{31}{8} = 12\frac{3}{8}$ evaluations of the high-fidelity objective and constraints and $6 + \frac{26}{8} = 10\frac{1}{2}$ evaluations of the high-fidelity objective and constraint sensitivities.

The SQP-AMMO approach yielded similar improvements in performance. Conventional SQP, applied to the synthetic cubic polynomial analysis, required 31 high-fidelity functions and 31 high-fidelity sensitivities. SQP-AMMO required four high-fidelity functions and 51 low-fidelity functions, for a total of $4 + \frac{51}{8} = 10\frac{3}{8}$ equivalent high-fidelity functions, and as many sensitivities. In the case of the synthetic spline analysis, conventional SQP required 21 high-fidelity functions and as many sensitivities. SQP-AMMO required four high-fidelity functions, four high-fidelity sensitivities, 28 low-fidelity analyses, and 28 low-fidelity sensitivities, for a total of $4 + \frac{28}{8} = 7\frac{1}{2}$ equivalent high-fidelity function evaluations and as many sensitivities.

All three AMMO algorithms produced consistent improvements in efficiency compared to conventional versions of the same algorithms. Improvements in efficiency are summarized in Table 1. We compare the costs of conventional optimization using a single model to that of optimization using AMMO. The entries in the table have the form “A/B”, where A is the ratio of the numbers of the objective and constraint evaluations and B is the ratio of the numbers of sensitivity evaluations.

We should emphasize that the amount of improvement as a result of AMMO cannot be predicted a priori. The only theoretical guarantee is the global convergence to a high-fidelity stationary point.

Two-Dimensional Airfoil Problem

Optimization Problem

The objective function is again $-C_L/C_D$, and the single nonlinear constraint is on C_M . Figure 7 depicts the two design variables,

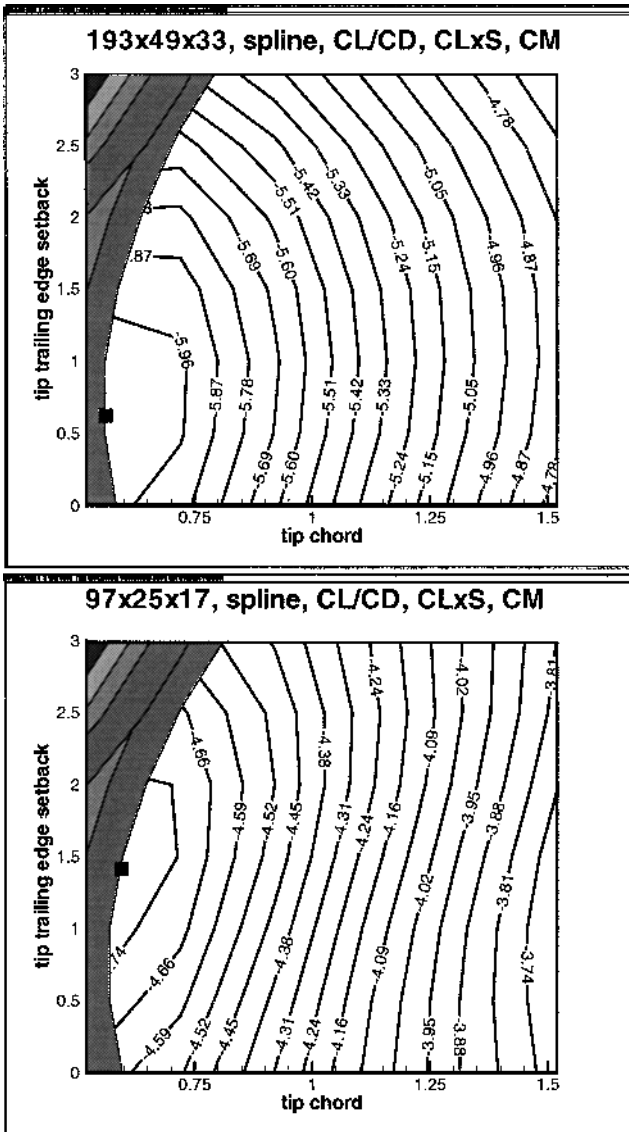


Fig. 5 High-fidelity vs low-fidelity objectives and active constraints: level sets from synthetic spline analysis.

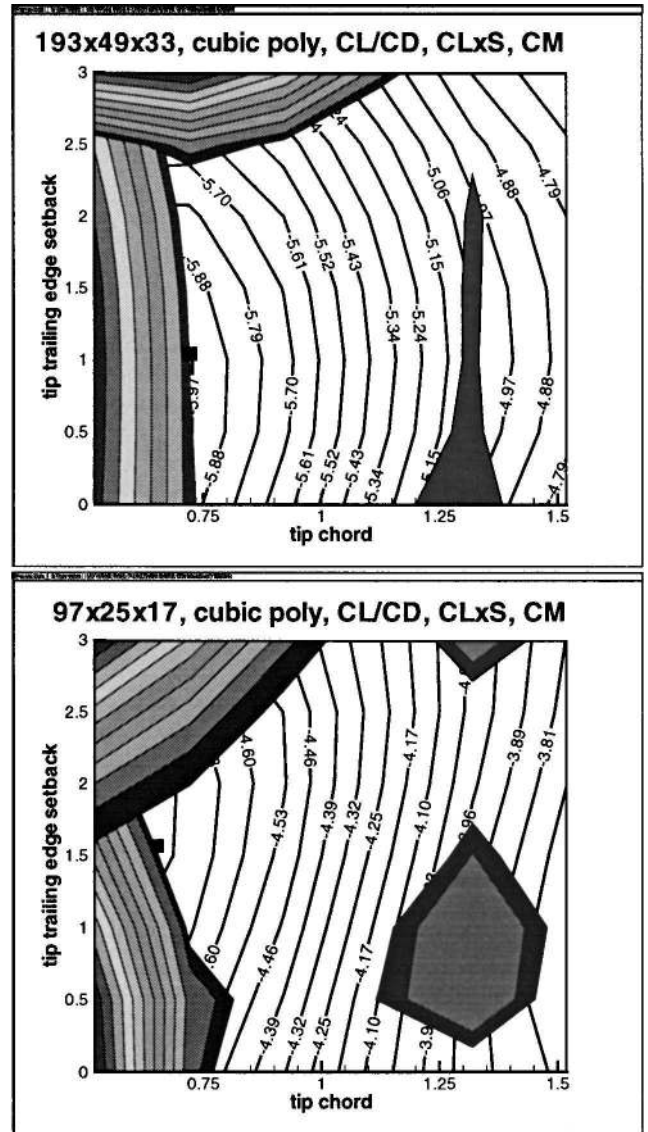


Fig. 6 High-fidelity vs low-fidelity objectives and active constraints: level sets from synthetic cubic polynomial analysis.

maximum camber and maximum thickness. The flow is transonic, with $M_\infty = 0.8$, and $\alpha = 0$ deg. Function and constraint values are obtained with the FLOMG code²⁵ evaluated on a 129×33 mesh and a 257×65 mesh, with the former providing the low fidelity. Figure 8 depicts the level sets obtained directly from FLOMG on the 129×33 and 257×65 meshes, respectively. Figure 9 depicts the level sets of the corresponding synthetic spline analyses. This problem's structure is favorable for AMMO: while the optima are at different locations, the low-fidelity objective and constraint exhibit the same general trends as their high-fidelity counterparts.

The time per analysis on the 257×65 mesh is approximately four times the analysis time on the 129×33 mesh. On an SGI Octane workstation the actual CPU times are approximately 8 and 2 min, respectively, iterating from freestream conditions. One-hundred multigrid iterations were done to converge each analysis; no other stopping convergence criterion was available in FLOMG.

Numerical Results

Again, AMMO consistently yielded improvements in efficiency compared to conventional versions of the same algorithms. However, the gains in relative efficiency are somewhat smaller (though still very good) than those observed for the three-dimensional wing problem because the relative costs of the low- and high-fidelity calculations are smaller for the two-dimensional calculations.

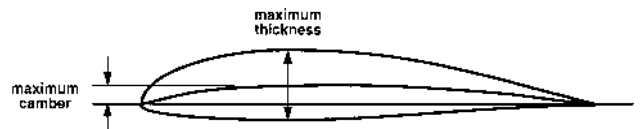


Fig. 7 Two-dimensional airfoil problem.

In tests done directly with FLOMG, MAESTRO required 34 evaluations of the objective and constraints and their sensitivities on the high-fidelity mesh. MAESTRO-AMMO required 20 evaluations of the objective and constraints and their sensitivities on the low-fidelity mesh and nine evaluations of the objective and constraints and their sensitivities on the high-fidelity mesh. A comparison is made by considering that 20 evaluations on the 129×33 mesh are equivalent to five evaluations on the 257×65 mesh. Therefore, MAESTRO-AMMO took 14 equivalent high-fidelity function and sensitivity evaluations. MAESTRO-AMMO took fewer iterations to find an answer than did conventional MAESTRO with the high-fidelity model. This result may appear surprising, but can be attributed to the fact that MAESTRO-AMMO took a different path through the design space.

The augmented Lagrangian AMMO was tested with a synthetic spline analysis. The conventional augmented Lagrangian algorithm (using analytical derivatives) required 58 evaluations of

Table 2 Airfoil optimization problem: summary of improvement as a result of AMMO

Algorithm	Improvement
Augmented Lagrangian	3.1/1.6 (spline)
SQP	2.2/2.2 (spline)
MAESTRO	2.4/2.4 (CFD)

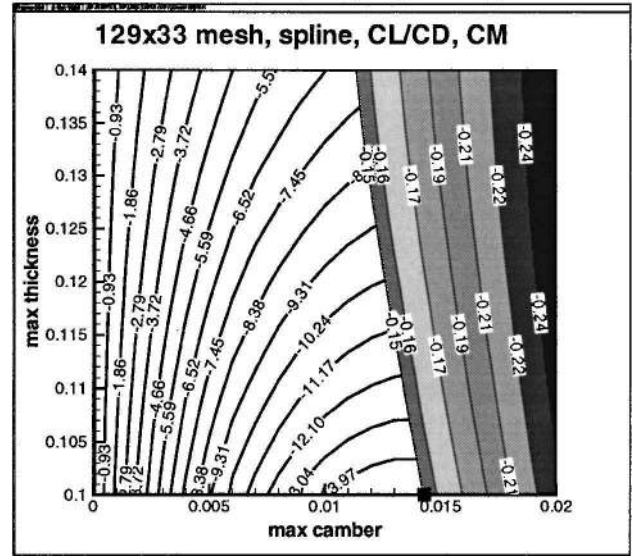
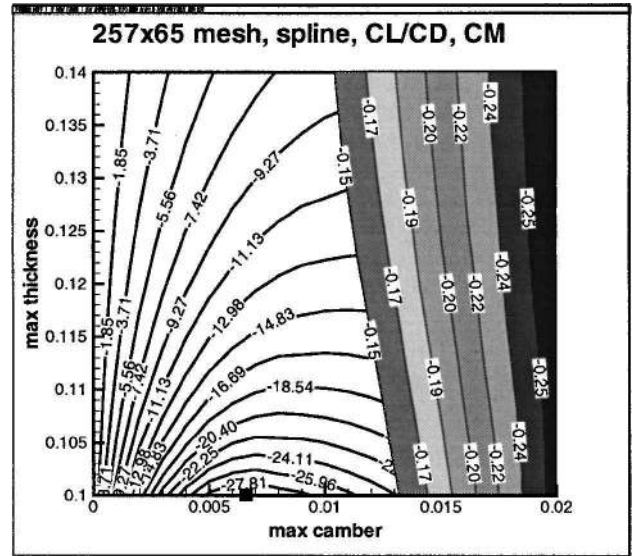


Fig. 9 High-fidelity vs low-fidelity objectives and active constraints: level sets from synthetic spline analysis.

required only four high-fidelity and 19 low-fidelity function and sensitivity calls, each, for a total of $4 + \frac{19}{4} = 8\frac{3}{4}$ equivalent high-fidelity analyses. The two-dimensional airfoil optimization results are summarized in Table 2; the entries have the same meaning as in Table 1.

Conclusions

In the experiments discussed here AMMO yielded about a three-fold improvement in computational cost for the three-dimensional wing design problem and a two-fold improvement for the two-dimensional airfoil problem. We believe that greater improvements can be achieved. No fine tuning of the AMMO approaches was done, and there is room for improvement in the interaction among all the pieces. In particular, currently the inner subproblem of minimizing the low-fidelity model is probably being solved to an unnecessarily high degree of accuracy. Because the analysis of the algorithms requires the subproblem solution to proceed only as far as needed to ensure sufficient predicted improvement in the merit function of the high-fidelity problem, the subproblems are almost certainly being oversolved in some instances. In the examples presented here, the relative cost of the low-fidelity analysis was not inconsequential compared to that of the high-fidelity analysis. In this situation, the efficiency of AMMO can be improved if it is determined how to terminate the inner subproblem as soon as it produces the necessary decrease.

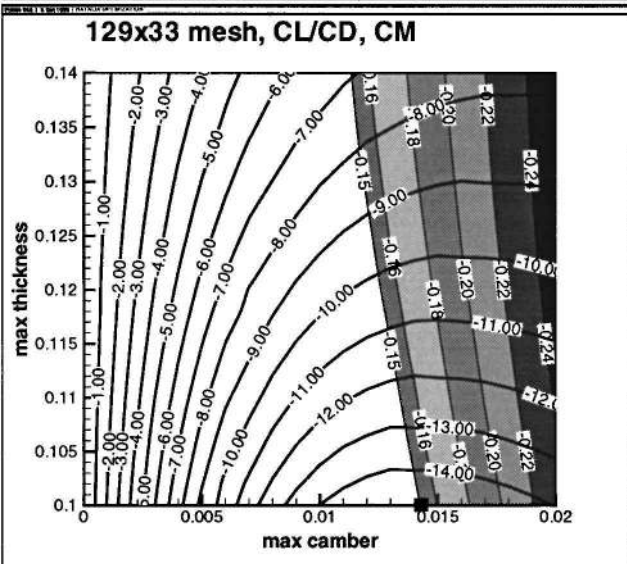
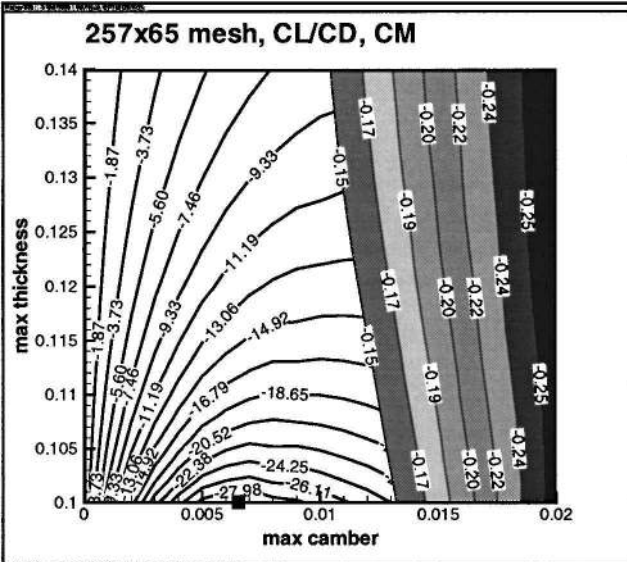


Fig. 8 High-fidelity vs low-fidelity objectives and active constraints: level sets from CFD analysis.

the high-fidelity objective and constraints and 21 evaluations of the high-fidelity objective and constraint sensitivities. The augmented Lagrangian AMMO required six evaluations of the high-fidelity objective and constraints, six evaluations of the high-fidelity objective and constraint sensitivities, 50 evaluations of the low-fidelity objective and constraints, and 30 evaluations of the low-fidelity objective and constraint sensitivities. Because the low-fidelity analyses take $\frac{1}{4}$ of the time of the high-fidelity analyses, the augmented Lagrangian AMMO required the equivalent work of $6 + \frac{50}{4} = 18\frac{1}{2}$ evaluations of the high-fidelity objective and constraints, and $6 + \frac{30}{4} = 13\frac{1}{2}$ evaluations of the high-fidelity objective and constraint sensitivities. These numbers indicate approximately three-fold improvement in the number of equivalent evaluations.

SQP-AMMO yielded similar improvements in performance. Conventional SQP applied to the synthetic spline analysis required 19 high-fidelity function and sensitivity calls each. SQP-AMMO

The efficacy of AMMO depends on the ability of the low-fidelity model to predict the trends in the high-fidelity model. We found that even when this prediction was not favorable, as in the case of the synthetic cubic polynomial analysis, the β correction proved effective in adjusting the low-fidelity model to follow the high-fidelity trends.

Although these initial experiments are promising, much work remains on further details of the implementation, as well as conclusions and practical guidance for using AMMO. As already mentioned, one question is that of the proper amount of optimization in the AMMO subproblems and the consequences for overall efficiency. The relative efficiency of AMMOs based on different underlying optimization algorithms is also of interest. At this point SQP-AMMO is the most promising for single discipline problems and for problem formulations that rely on multidisciplinary analysis. A variant of the augmented Lagrangian approach may have merit in the multidisciplinary setting as well. The MAESTRO approach is also promising for multidisciplinary problems. The AMMO idea will also be applied to a broader class of problems and variable-fidelity models. In particular, AMMO with variable fidelity physics models presents an intriguing line of inquiry.¹⁰

Acknowledgments

The authors would like to thank Philip E. Gill for a copy of NPSOL, Anthony A. Giunta for a copy of RSG, and the referees for their many helpful comments. Part of R. M. Lewis' research was supported by NASA under Contract NAS1-97046 while this author was in residence at the Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, Virginia.

References

- ¹Schmit, L. A., Jr., and Farshi, B., "Some Approximation Concepts for Structural Synthesis," *AIAA Journal*, Vol. 12, No. 5, 1974, pp. 692-699.
- ²Schmit, L. A., Jr., and Fleury, C., "Structural Synthesis by Combining Approximation Concepts and Dual Methods," *AIAA Journal*, Vol. 18, No. 10, 1980, pp. 1252-1260.
- ³Schmit, L. A., Jr., and Miura, H., "Approximation Concepts for Efficient Structural Synthesis," NASA TR CR-2552, March 1976.
- ⁴Barthelemy, J.-F. M., and Haftka, R. T., "Approximation Concepts for Optimum Structural Design—a Review," *Structural Optimization*, Vol. 5, No. 3, 1993, pp. 129-144.
- ⁵Jameson, A., "Essential Elements of Computational Algorithms for Aerodynamic Analysis and Design," Inst. for Computer Applications in Science and Engineering, TR 97-68, NASA Langley Research Center, Hampton, VA, Dec. 1997.
- ⁶Newman, J. C., III, Taylor, A. C., III, Barnwell, R. W., Newman, P. A., and Hou, G. J.-W., "Overview of Sensitivity Analysis and Shape Optimization for Complex Aerodynamic Configurations," *Journal of Aircraft*, Vol. 36, No. 1, 1999, pp. 87-96.
- ⁷Haftka, R. T., and Gürdal, Z., *Elements of Structural Optimization*, 3rd ed., Kluwer Academic, Norwell, MA, 1993.
- ⁸Alexandrov, N. M., and Lewis, R. M., "A Trust Region Framework for Managing Approximation Models in Engineering Optimization," AIAA Papers 96-4101 and 96-4102, Sept. 1996.
- ⁹Alexandrov, N. M., "On Managing the Use of Surrogates in General Nonlinear Optimization," AIAA Paper 98-4798, Sept. 1998.
- ¹⁰Alexandrov, N. M., Nielsen, E. J., Lewis, R. M., and Anderson, W. K., "First-Order Model Management with Variable-Fidelity Physics Applied to Multi-Element Airfoil Optimization," AIAA Paper 2000-4886, Sept. 2000.
- ¹¹Conn, A. R., Gould, N., and Toint, P. L., *Trust-Region Methods*, MPS-SIAM Series on Optimization, Society for Industrial and Applied Mathematics-MPS, Philadelphia, 2000.
- ¹²Chang, K. J., Haftka, R. T., Giles, G. L., and Kao, P.-J., "Sensitivity-Based Scaling for Approximating Structural Response," *Journal of Aircraft*, Vol. 30, No. 2, 1993, pp. 283-288.
- ¹³Alexandrov, N. M., and Lewis, R. M., "First-Order Approximation and Model Management in Optimization," *Proceedings of First Sandia Workshop on Large-Scale PDE-Constrained Optimization*, Springer-Verlag, Berlin, 2001.
- ¹⁴Alexandrov, N. M., "Multilevel Methods for MDO," *Multidisciplinary Design Optimization: State of the Art*, edited by N. M. Alexandrov and M. Y. Hussaini, Society for Industrial and Applied Mathematics, Philadelphia, 1997, pp. 79-89.
- ¹⁵Gill, P. E., Murray, W., and Wright, M. H., *Practical Optimization*, Academic Press, London, 1981.
- ¹⁶Fletcher, R., *Practical Methods of Optimization*, 2nd ed., Wiley, New York, 1989.
- ¹⁷Gill, P. E., Murray, W., Saunders, M. A., and Wright, M. H., *User's Guide for NPSOL (Version 5.0): A Fortran Package for Nonlinear Programming*, Stanford Univ., CA, 1995.
- ¹⁸Kaufman, L., and Gay, D., *PORT Library: Optimization and Mathematical Programming*, Bell Labs., Murray Hill, NJ, May 1997.
- ¹⁹Koehler, J. R., and Owen, A. B., "Computer Experiments," *Design and Analysis of Experiments*, edited by S. Ghosh and C. R. Rao, Vol. 13 of Handbook of Statistics, North-Holland, New York, 1996, pp. 261-308.
- ²⁰Kaufman, M., RSG: Response Surface Generation Program, Version 1.32, Virginia Polytechnic Inst. and State Univ., Blacksburg, VA, Jan. 1996.
- ²¹Taylor, A. C., III, Oloslo, A., and Newman, J. C., III, "CFL3D.ADII (Version 2.0): An Efficient, Accurate, General-Purpose Code for Flow Shape-Sensitivity Analysis," AIAA Paper 97-2204, Jan. 1997.
- ²²Rumsey, C., Beidron, R., and Thomas, J., "CFL3D: Its History and Some Recent Applications," NASA TM 112861, May 1997.
- ²³Smith, R. E., Bloor, M. I. G., Wilson, M. J., and Thomas, A. T., "Rapid Airplane Parametric Input Design (RAPID)," *Proceedings of the 12th AIAA Computational Fluid Dynamics Conference*, AIAA, New York, 1985, pp. 452-462.
- ²⁴Bischof, C., Jones, W. T., Samareh-Abolhassani, J., and Mauer, A., "Experiences with the Application of the ADIC Automatic Differentiation Tool to the CSCMDO 3-D Volume Grid Generation Code," AIAA Paper 96-0716, Jan. 1996.
- ²⁵Swanson, R., and Turkel, E., "A Multistage Time-Stepping Scheme for the Navier-Stokes Equations," AIAA Paper 85-0035, Jan. 1985.