An Investigation of Multi-Modality in Aerodynamic Shape Optimization

Oleg Chernukhin* and David W. Zingg†

Institute for Aerospace Studies, University of Toronto, Toronto, Ontario, M3H 5T6, Canada

Two novel optimization algorithms are presented for aerodynamic shape optimization. The first is a gradient-based multi-start method based on Sobol sequence sampling. The second is a hybrid approach that combines Sobol sampling, a genetic algorithm to perform a global search, and a gradient-based algorithm for local refinement. These two algorithms are compared against a pure gradient-based optimizer and a gradient-free genetic algorithm. The performance of the algorithms is tested on an analytical test function, as well as several aerodynamic shape optimization problems in two and three dimensions. The results demonstrate the effectiveness of both the multi-start and the hybrid approaches, with the multi-start method being the preferred choice for most practical problems. In addition, the results show that multi-modality should not always be assumed in aerodynamic shape optimization problems. Multiple local optima, although they do exist in many problems, are not as prevalent as is commonly accepted. We present a classification of optimization problems based on the degree of multi-modality and suggest appropriate algorithms for each problem class.

I. Introduction

Rising fuel prices and climate change present the main challenges for the aviation industry in the 21st century. These considerations will increase the pressure on the aircraft industry to design more fuel-efficient aircraft. Over the past decades, gains in fuel efficiency have mainly come from improvements in engine technology, reduced weight due to the use of composite materials, and reduction of the overall drag through more aerodynamically efficient airframe designs.

The use of computer algorithms for aerodynamic shape optimization (ASO) has the potential to uncover unconventional aircraft configurations that can lead to dramatic reductions in drag. The two major components of ASO are efficient computational fluid dynamics (CFD) solvers and optimization algorithms. While CFD has become a mature technology and has found numerous industrial applications, the use of optimization for CFD-based design is still undergoing rapid development.

Optimization is a very broad field, and the choice of a suitable algorithm is highly problem-dependent. Considerations must be made with regard to the types of design variables, the number of constraints, the properties of the design space, etc. In this work, we assume that the design space is smooth and the design variables are continuous.

Even with these assumptions, many choices of optimization algorithms are available for ASO.[1–3] Traditionally, optimization algorithms have been divided in two broad categories: gradient-free and gradient-based (GB) methods. Both types have been used for ASO, with compelling arguments presented in favour each method. However, various hybrid approaches, incorporating elements from both GB and gradient-free algorithms, have been proposed and successfully applied.[3,4] These hybrid algorithms attempt to address the shortcomings of the traditional optimization methods.

Gradient-based algorithms require sensitivities of the objective and the constraints in order to reach a local optimum. Quasi-Newton methods, such as BFGS,[5] construct a Hessian approximation at each iteration. The main advantage of gradient-based algorithms is their rapid convergence. The difficulties associated with

*MASc Candidate, AIAA Student Member

†Professor and Director, Tier 1 Canada Research Chair in Computational Aerodynamics, J. Armand Bombardier Foundation Chair in Aerospace Flight, Associate Fellow AIAA

GB optimizers arise in the need for efficient gradient calculations and in their inherent tendency to converge to local optima.

Gradient-free approaches usually mimic some real-life phenomenon in an attempt to minimize the objective function. Since gradient information is not required, these algorithms can be easily incorporated into existing frameworks, and have found many applications in academia and industry. Genetic algorithms (GAs) are among the most popular gradient-free methods in use today.[6,7] Aside from their ease of implementation, GAs are particularly suitable for problems with discrete design variables, problems with discontinuous objectives, and problems with multiple local optima (multi-modality). The main disadvantage of genetic algorithms is their slow convergence. Zingg et al.[8] show that for 2-D airfoil optimization problems, a GA can require 200 times more function evaluations than an efficient GB method.

While some advantages of the gradient-free optimizers are obvious, the issue of multi-modality is largely unsettled. Some authors suggest that the existence of multiple local optima is obvious due to the non-linearity of the equations governing the fluid flow.[9] Namgoong et al.[10] attempt to investigate the multi-modality and conclude that since the gradient-based algorithm generates different results, depending on the initial guess, the problem is multi-modal. Various additional publications also suggest that multi-modality makes gradient-based algorithms a poor choice for ASO.[3,11] The existence of multiple local optima may be challenging to disprove. However, it is possible to prove that a local optimum has been reached, provided that gradient information is available. Unfortunately, plots that show gradient convergence or some other measure of optimality are often omitted from papers on ASO.

Nevertheless, multiple local optima certainly exist in aerodynamic shape optimization problems. Buckley et al.[12] present a practical multi-point airfoil optimization case and show that at least two local optima exist for this problem. Leung and Zingg[13] show that a 3-D optimization of an ONERA M6 wing produces at least two local optima, one for forward and one for backward sweep. Zingg and Hicken[14] show that for spanwise vertical shape optimization, at least two local minima exist as well (winglet-up and winglet-down).

The difficulty with aerodynamic shape optimization lies in the fact that the objective function evaluation is computationally expensive, and only a limited number of flow solutions can usually be afforded. In addition to this, practical ASO problems involve many design variables, which makes the design space impossible to explore thoroughly and visualize. These factors make it difficult to draw conclusions about the nature of the design space, although increased computer speeds are beginning to allow for some comparative studies.[8,15]

In this work, we propose two novel optimization algorithms which are designed to be computationally efficient and to avoid converging to local optima. The algorithms employ features from a GA, a GB algorithm, as well as an efficient sampling process in order to thoroughly explore the design space. We first test the algorithms on a highly multi-modal test function, and proceed to apply them to practical optimization problems. In addition, we evaluate two existing optimization methods that can be placed at the opposite ends of the gradient-free/gradient-based spectrum. The objective of this paper is to investigate the multi-modality of the ASO design spaces, to introduce two new algorithms for dealing with multi-modal design spaces, and to provide some guidance with regard to selecting an appropriate optimization algorithm for a specific problem.

The paper is presented in three parts. The first part provides an overview of the main features of the integrated ASO methods used in this work, along with the other components required. In the second part, the detailed descriptions of the optimization algorithms are given. The results of the optimization problems considered in this work are presented in the last section.

II. Overview of the Integrated Aerodynamic Shape Optimization Methods

The key components of the aerodynamic shape optimization algorithms are thoroughly described by Nemec and Zingg[1,16] for 2-D problems and Hicken and Zingg[17,18] for 3-D problems. Here, some of the main features of these algorithms are briefly summarized.

The geometry parameterization is accomplished using B-spline curves in 2-D and B-spline surfaces in 3-D. The design variables in 2-D are vertical coordinates of the B-spline control points. In 3-D, the design variables are the x, y, and z coordinates of the B-spline surface control points. Depending on the problem definition, each control point is assigned anywhere from 0 to 3 design variables.

In the 2-D algorithm, the algebraic mesh movement algorithm of Nemec[1] is used. In the 3-D algorithm, the linear elasticity method of Truong et al.[19] is employed, but applied to a B-spline volume control point mesh, as described by Hicken and Zingg.[18]

The governing equations in 2-D are the compressible Navier-Stokes equations, with the Spalart-Allmaras turbulence model used to compute the eddy viscosity. The discretized equations are solved using a Newton-Krylov approach.[1]

In 3-D the governing equations are Euler equations. Work to incorporate viscous terms and turbulence modelling is underway;[20] however only inviscid flows are considered in 3-D problems here. The spatial discretization is accomplished using Summation By Parts operators, and the interface conditions between blocks are enforced using Simultaneous Approximation Terms. The discretized equations are solved using a Newton-Krylov-Schur approach.[17]

III. Optimization Components

In general terms, an optimization problem can be stated as follows:

$$\begin{aligned} \text{minimize} \quad & \mathcal{J}_o(\mathbf{X}), \\ \text{w. r. t.} \quad & \mathbf{X}, \\ \text{s. t.} \quad & a \le \mathcal{J}_{m_j}(\mathbf{X}) \le b, \quad j = 1, ..., n_m, \\ & c \le \mathcal{J}_{l_k}(\mathbf{X}) \le d, \quad k = 1, ..., n_l. \end{aligned}$$

$\mathcal{J}_o$ is the aerodynamic objective function to be minimized, often C_d or C_d/C_l. $\mathbf{X}$ is a vector of design variables, such as the coordinates of the B-spline control points and/or the angle of attack α. $\mathcal{J}_{m_j}$ and $\mathcal{J}_{l_k}$ are non-linear and linear constraints, respectively. Examples of these are a wing volume constraint, a lift constraint, and upper/lower bounds on the design variables (box constraints).

Constraints can be incorporated into the objective function using a Quadratic Penalty Method (QPM):

$$\mathcal{J}(\mathbf{X}, \rho) = \mathcal{J}_o(\mathbf{X}) + \rho \sum_{i=1}^{n_m} (\max[0, a - \mathcal{J}_{m_i}(\mathbf{X})])^2 + \rho \sum_{j=1}^{n_m} (\max[0, \mathcal{J}_{m_j}(\mathbf{X}) - b])^2. \tag{1}$$

The choice of the appropriate penalty parameter ρ is not a trivial task, especially for practical design problems.[12] The QPM is often used in gradient-free optimizers. However, for gradient-based optimization algorithms, the preferred way to satisfy constraints is through the solution of the Karush-Kuhn-Tucker (KKT) equations using Sequantial Quadratic Programming (SQP) methodology.[5] In this work, both QPM and SQP techniques are employed.

The optimization framework used in this work consists of multiple interacting components. These are described in the following subsections.

A. The Adjoint Method

Efficient gradient calculation is an important part of any gradient-based optimization framework. Although various approaches exist for gradient evaluation, the adjoint method is the most efficient method when the number of design variables exceeds the number of nonlinear constraints. The chief advantage of the adjoint method is that the gradient calculation is nearly independent of the number of design variables.

In the discrete adjoint method, the gradient $\mathcal{G}$ of the objective function is found using the following equation:

$$\mathcal{G} = \frac{\partial \mathcal{J}}{\partial \mathbf{X}} - \mathbf{\Psi}^T \frac{\partial \mathcal{R}}{\partial \mathbf{X}}. \tag{2}$$

In the above equation, Ψ is the adjoint vector, which is found by solving the following system:

$$\left(\frac{\partial \mathcal{R}}{\partial \mathbf{Q}}\right)^T \mathbf{\Psi} = -\left(\frac{\partial \mathcal{J}}{\partial \mathbf{Q}}\right)^T. \tag{3}$$

$\mathcal{R}$ is the residual of the discretized governing Euler or Navier-Stokes equations. The details of how the adjoint method is implemented can be found in Nemec and Zingg[1] for 2-D problems and Hicken and Zingg[18] for 3-D problems.

B. Sobol Sampling

In order to understand the behaviour of an objective function with respect to its design variables, an evaluation of the objective function must be performed at a certain number of design points. While random sampling can be employed, more efficient strategies have been devised to perform this task. Some of the most popular sampling techniques include Latin Hypercube sampling and Sobol sequences. In this work, we choose the Sobol sequence, due to its deterministic behaviour, ability to perform incremental sampling, and the property of maintaining favourable sampling characteristics when projected to lower-dimensional design spaces.[21]

A Sobol sequence, also known as an LP_τ-sequence, is commonly used for efficient sampling. It was originally introduced by Sobol in 1967, with the goal of approximating an integral of a d-dimensional function on a unit "hypercube" with the fastest possible convergence. We use an extension of Algorithm 659,[22] which uses Gray code implementation for generating Sobol sequences. To generate the d^{th} dimension of the n^{th} sample point $x_{d,n}$ we use a recursive relation:

$$x_{d,1} = 0 \qquad \text{and} \qquad x_{d,n} = x_{d,n-1} \oplus v_{d,c_{n-1}},$$

where $\oplus$ is a bitwise exclusive-or operator, c_n is the index of the first 0 digit from the right of the binary representation of n, and $v_{d,n}$ is the directional number, defined as:

$$v_{d,n} = \frac{m_{d,n}}{2^d}.$$

A proper Sobol sequence requires a set of carefully generated directional numbers. The list of numbers $m_{d,n}$ used in this work is provided by Joe and Kuo[22] and can be used for dimensions up to $d = 21201$.

C. SNOPT Optimization Algorithm (GB)

The gradient-based optimizer used is the optimization package SNOPT. Developed by Gill et al.,[23] SNOPT uses an SQP algorithm to find the solution to nonlinear optimization problems with general constraints. The Hessian of the Lagrangian function is approximated using the quasi-Newton BFGS method.

D. Gradient-based Multi-start Algorithm Based on Sobol Sampling (GB-MS)

Since a gradient-based optimization algorithm, such as SNOPT, uses one initial guess and converges to a local optimum only, it is logical to consider an algorithm that uses multiple starting points to initiate gradient-based optimization. The advantage of incremental sampling is particularly important here. If a Sobol sequence is employed, the user can optimize the first n initial points in this sequence using a GB algorithm. Once these points are optimized, the user can then increment the sampling and optimize the next n initial points. This process can be repeated until some termination criterion has been met (e.g. time constraints, limits on computational resources, the user is convinced that the design space has been thoroughly explored).

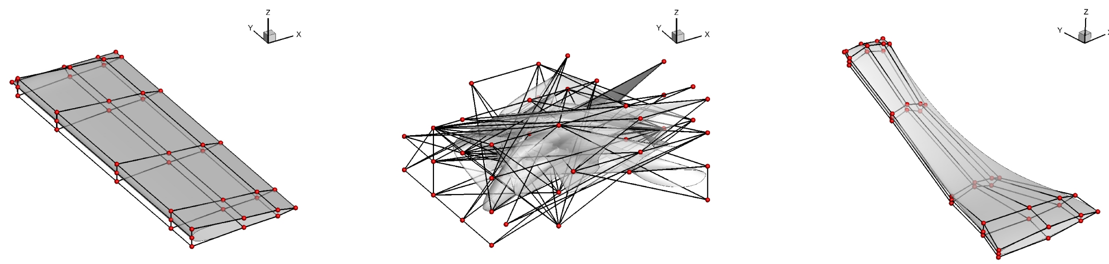
The parallel implementation, is done using a Message Passing Interface library. We use GB-MS to denote this algorithm. In the next subsection, we discuss the subtleties of creating initial samples for practical ASO problems in three dimensions.

Generating Initial Guesses Using SNOPT and Sobol Sampling

A useful feature of SNOPT is the ability to separate linear and non-linear constraints. This becomes an important concept when considering complex optimization problems, especially in 3-D. As described in Section II, the only design variables allowed are the coordinates of the B-spline patch control points. Although this allows for great flexibility in the deformation of the aerodynamic shape, it also presents a challenge in terms of global optimization.

Consider a 3-D wing, consisting of two patches, as shown in Fig. 1(a). Each patch is parameterized with 30 control points: 5 in the streamwise and 6 in the spanwise direction. Each control point has 3 degrees of freedom, and after the frozen (root trailing edge) and duplicate control points are removed, the total number of design variables is 126 (125 geometric DVs and the angle of attack).

How do we create a Sobol sample of the design space with 126 design variables? If we put common box constraints on all control points, the fraction of the initial guesses that would resemble a wing would be



(a) Baseline geometry (b) Example of a random sample (c) Example of a desired geometric deformation

Figure 1. Wing Parameterization

dwarfed by the number of the infeasible shapes (see Fig. 1(b)). A large portion of computational time would be needed just to create a feasible initial population. Even in 2-D, such an approach leads to shapes that would be difficult to analyze using computational tools.[7] This would clearly be a daunting task for 3-D ASO problems. On the other hand, if we manually select an appropriate box constraint for each control point, we are left with an unacceptably restricted design space, essentially disallowing variations in sweep, dihedral, and other "global" shape changes.

Various methods have been proposed to address these issues. Leung and Zingg[13] allow grouping of control points into new design variables (e.g. sweep), which are explicit in the definition of the problem. Morris et al.[24] separate local and global design variables using a novel domain-element methodology. Our approach is through the use of linear constraints. The "global" (sweep, span, dihedral) shape is determined by the control point on the tip trailing edge. The remaining control points are forced to move with the tip trailing edge to maintain the basic wing shape. However, some freedom can be allowed for "local" deformations, such as twist, individual section changes, etc. The optimization problem is carefully set up using linear equality and inequality constraints. SNOPT is then able to satisfy these constraints exactly, without calling the flow solver. Although setting up linear constraints is somewhat time-consuming, this approach is consistent with the idea that the designer's efforts should be spent on defining the optimization problem, while the computer is tasked with producing the optimal shape.[25]

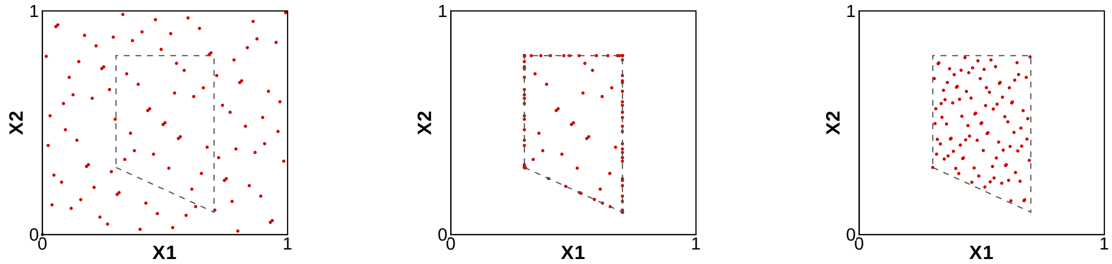
Using this method, we can create a Sobol sample, which does not look like a wing (Fig. 1(b)), and "shape" it into a wing by enforcing the linear constraints (Fig. 1(c)). This process takes place in on the order of seconds or less. The original 126-DV design space is essentially reduced to a smaller subset. Unfortunately, the resulting sample is heavily biased towards the boundaries of the "linear feasible" region $\mathcal{R}_L$. Fig. 2 provides an illustration of this process. In Fig. 2(a) a 2-D sample is shown with the linear feasible region in dashed lines. In Fig. 2(b) is the same sample with linear constraints satisfied by SNOPT. Clearly, many points are on the boundaries of $\mathcal{R}_L$ and the region is not covered well.

In order to address this issue, we have designed a procedure for sampling only within $\mathcal{R}_L$. The main idea is to provide the proper order in which the linear constraints are implemented. Then, it is straightforward to establish the upper and lower bounds on each design variable and sample only within these bounds. Fig. 2(c) shows a sample that results from using this procedure to sample only within $\mathcal{R}_L$. Clearly, the linear feasible region is now covered with samples in a uniform, unbiased manner and is an obvious improvement over the sample in Fig. 2(b).

E. Genetic Algorithm (GA)

Genetic Algorithms have been used extensively in the aerospace field.[2,7,8] GAs attempt to minimize the objective function by mimicking the process of evolution. In the GA context, the design points are called chromosomes, the objective function is the fitness, and genes refer to either the design variables or sub-strings of bit-encoded design variable strings.

Despite numerous variations, the basic operators of any Genetic Algorithm are selection, crossover, and mutation. Once each chromosome is assigned a fitness value, the selection process decides which chromosomes are considered in the creation of a new generation. The crossover operator combines various genes from the



(a) Sobol sample and $\mathcal{R}_L$

(b) Allowing SNOPT to satisfy linear constraints

(c) Sampling only in $\mathcal{R}_L$

Figure 2. Sampling Method

two chromosomes (parents) to create a new one (child). The mutation operator assigns random values to some of the genes in a chromosome. Through the process of selection, crossover, and mutation, the subsequent generations are created. With the aid of the CFD solver, the fitness values are assigned to the chromosomes in the new generation, and the process continues. Although other gradient-free optimizers, such as Particle Swam Optimization, appear promising,[26] a genetic algorithm will be used in this work.

The GA used in this project was developed at the NASA Ames Research Center by Holst and Pulliam.[27] We briefly summarize the main features of the algorithm. Real number encoding is used, which is more practical when all design variables are continuous. In order to create the next generation, the GA uses four basic operators: passthrough, crossover, pure mutation, and perturbation mutation. The input parameter p-vector states what percentage of the chromosomes within the new generation are created using a particular operator. For example, $p = [10, 20, 30, 40]$ states that 10% of the chromosomes are created using passthrough, 20% - crossover, 30% - pure mutation, and 40% - perturbation mutation.

The passthrough operator takes the fittest chromosomes from the previous generation and passes them on to the next generation. This ensures that information from the fittest chromosomes is never lost. The crossover operator combines two chromosomes in the following way:

$$\mathcal{X}_{new} = \frac{1}{2}(\mathcal{X}_1 + \mathcal{X}_2).$$

The pure mutation operator takes one chromosome from the previous generation and replaces some of its genes with random values:

$$x_{new} = \text{RAND}(x_{max} - x_{min}) + x_{min},$$

where RAND is a random number between 0 and 1. The perturbation mutation takes one chromosome from the previous generation and perturbs some of its genes:

$$x_{new} = x_{old} + \beta(x_{max} - x_{min})(\text{RAND} - 0.5),$$

where β is a user specified parameter. Checks are in place to ensure that x_{new} does not exceed its box constraints. For the GA optimizer, the p-vector used in this work is:

$$p = [1/n, (n/2) - 1, n/4, n/4],$$

where n is the population size. Nonlinear constraints are enforced using the QPM. In this work, the penalty term weight ρ is 10. Whenever applicable, the linear constraints are satisfied exactly using SNOPT, as described in the previous section.

F. Hybrid Optimization Algorithm (HM)

Numerous ideas for hybrid optimization algorithms can be found in the literature.[3,4,9,21,28] Although many ideas involve use of surrogate modeling to interpolate the objective function, this approach will not be taken here, because surrogate models can introduce difficulties for high-dimensional optimization problems.

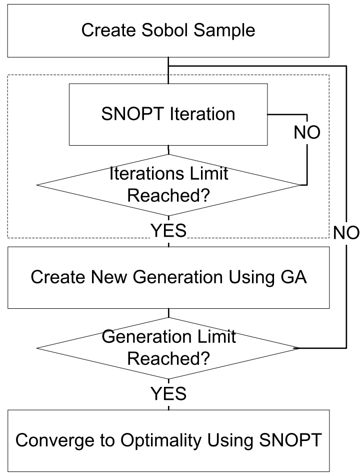


Figure 3. Hybrid Optimizer Block Diagram

We propose a hybrid optimizer that takes advantage of the GA's ability to perform a global search and SNOPT's ability to efficiently find the nearest local optimum. Whenever a new generation is created by the GA, the resulting chromosomes are passed to SNOPT for gradient-based refinement. We limit the number of SNOPT major iterations to equalize the amount of time SNOPT spends on each chromosome, which permits better scaling of the optimizer, i.e. load balancing. The refined chromosomes are then passed back to the GA to create a new generation. The block diagram of the hybrid optimizer is shown in Fig. 3. One clear advantage of this approach is that the optimization process will not stop once a local optimum is found. If SNOPT finds that a chromosome satisfies the optimality conditions, a note of this is made for future reference, but the GA will continue exploring the design space. Unlike a GA, the hybrid optimizer can identify the local optima, since SNOPT can assess whether the optimality conditions are satisfied.

The perturbation mutation operator is designed to improve the GA refinement capability. Since we use a gradient-based optimizer for refinement, the perturbation mutation is avoided in the hybrid optimizer. For the HM optimizer, the p-vector is:

$$p = [1/n, (n/2) - 1, n/2, 0].$$

IV. Optimization Problems

The difficulty with assessing the performance of any optimization algorithm is that one algorithm may be more suited than another, depending on the optimization problem at hand. Our main motivation in the development of the optimization algorithms is the ability to handle a design space with multiple local optima. However, depending on the nature of the design space, this optimizer may not be the best option. We classify optimization problems according to the number of local optima, as shown in Table 1.

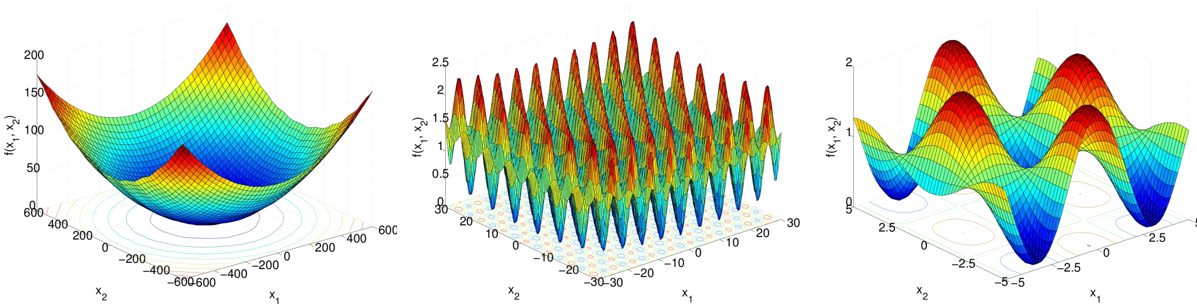
Number of Local Optima l	**Classification**
$l = 1$	Unimodal
$1 < l \leq 10$	Somewhat Multi-Modal
$10 < l \leq 100$	Moderately Multi-Modal
$l > 100$	Highly Multi-Modal

Table 1. Classification of Optimization Problems by Multi-Modality

For each optimization problem considered, four optimization methods will be used. These methods are summarized in Table 2. Every problem will be classified according to the criteria in Table 1, and the most suitable optimization method for this problem will be selected from Table 2.

Description	Acronym
Gradient-based method, single (baseline) initial guess	GB
Gradient-based method, multiple initial guesses determined by Sobol sequence	GB-MS
Hybrid optimization method	HM
Genetic algorithm with linear constraints handled by SNOPT	GA

Table 2. Optimization Methods



(a) $-600 \leq x_i \leq 600$ (b) $-30 \leq x_i \leq 30$ (c) $-5 \leq x_i \leq 5$

Figure 4. Griewank Test Function

A. Finding the Global Minimum of an Analytical Test Function

Before aerodynamic shape optimization problems are considered, the performance of the algorithms on an analytical test function is assessed. The Griewank function is particularly suitable for testing of the global optimization algorithms, since it is smooth, differentiable, and highly multi-modal. Its general definition in n dimensions is given below:

$$f(x) = \sum_{i=1}^{n} \left(\frac{x_i^2}{4000} \right) - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1. \tag{4}$$

Although it can be used in higher dimensions, the restriction to $n = 2$ is made here, since this design space can be easily visualized. Furthermore, the non-ideal behaviour of this function in higher dimensions makes the assessment of optimization algorithms more complicated, as shown by Locatelli.[29]

This function has a global optimum $f(0,0) = 0$. Its main features can be understood by examining Fig. 4, where the Griewank function is plotted over different domains. Although the function looks convex from far away, its highly multi-modal nature becomes clear when a smaller region of the domain is examined. Two cases will be considered: a highly multi-modal problem, and a moderately multi-modal problem. The Griewank function can be used for both cases, by adjusting the box constraints.

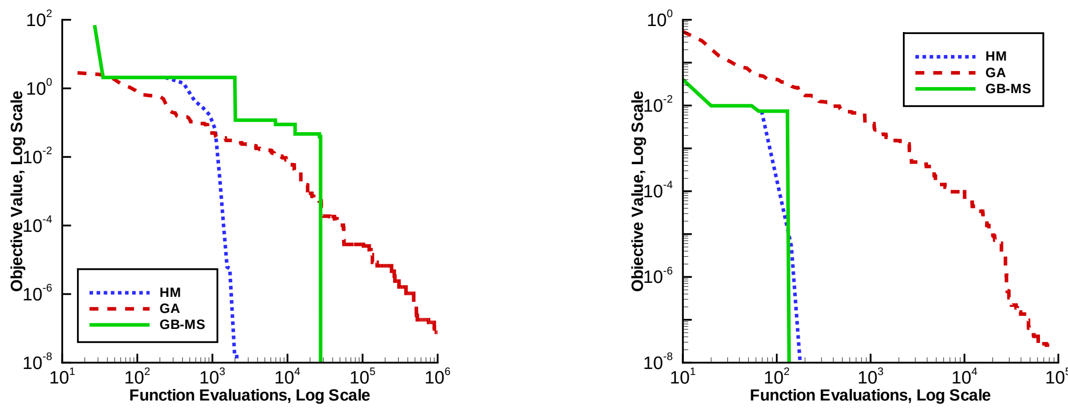
Problem 1: Highly Multi-Modal Test Case

For the highly multi-modal test case, the box constraints are:

$$-1234 \leq x_1 \leq 1321, -934 \leq x_2 \leq 778.$$

By inspecting Fig. 4, it should be clear that the objective function contains thousands of local optima and would present a challenge for the single-initial guess GB optimizer. Therefore, this optimization method will not be used for this problem. Due to their stochastic nature, 10 optimization runs are performed using the GA and HM optimizers, and the geometric mean of these 10 runs is calculated and plotted. Since the multi-start (GB-MS) procedure is fully deterministic (an advantage of the Sobol sequence), only one run is required. The population size is 16 for both GA and HM, and the SNOPT major iterations limit for HM is 10.

We consider the total number of function evaluations it takes to reach the global minimum $f(0,0) = 0$ to within 10^{-8}. The convergence plots are shown in Fig. 5(a). For the GA and the HM, the chromosome with



(a) Problem 1 Convergence History

(b) Problem 2 Convergence History

Figure 5. Convergence Plots for Problems 1 and 2

the best fitness value is plotted, at the end of each generation. For the GB-MS procedure, we continuously sample using a Sobol sequence. After each sample point is driven to optimality, we plot the best objective value found.

The genetic algorithm consistently performs the worst of the three algorithms considered, even though a few of its 10 runs are considerably more successful than the others in reaching the global optimum. The GB-MS method required 27643 function evaluations and 1907 Sobol sample points to find the global minimum. The hybrid optimizer consistently outperforms the other algorithms, succeeding in reaching the global minimum every time.

Problem 2: Moderately Multi-Modal Test Case

For the moderately multi-modal problem, the box constraints on the Griewank function are:

$$-12.34 \leq x_1 \leq 3.76, -11.41 \leq x_2 \leq 8.45.$$

There are 18 local optima in this case. Clearly, one would expect the advantage of the hybrid algorithm over the GB-MS method to be diminished, since far fewer sample points should be required to find the global minimum. We perform 10 optimization runs for each GA and the HM algorithms, and 1 run for the GB-MS procedure.

The results support the hypothesis: the GB-MS and HM algorithms require a comparable number of function evaluations to reach the global optimum. This is evident from the convergence plots in Fig. 5(b). Again, as in the previous test problem, the GA performs the worst of the three algorithms tested.

Based on the results from Problems 1 and 2, we can draw a few intermediate conclusions. If the design space for a practical aerodynamic optimization problem is highly multi-modal, the hybrid optimization method presented (HM) may be the preferred choice. However, if the design space contains only a few local optima, a multi-start procedure (GB-MS) may be most efficient in reaching a global optimum. With these thoughts we proceed to more practical optimization cases.

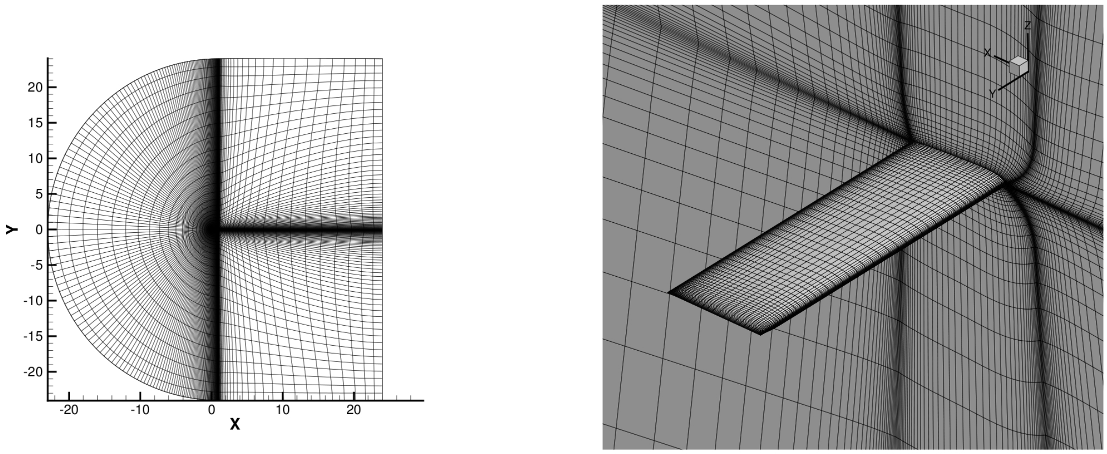
B. 2-D Airfoil Optimization Problem

Problem 3: Airfoil Optimization

The 2-D ASO problem is performed using the Newton-Krylov algorithm described in Section II. The mesh used is a 289 × 65 point C-mesh generated around the RAE 2822 airfoil, shown in Fig. 6(a). The geometry is parametrized using 15 B-spline control points, some of whose y-coordinates are design variables.

We perform the optimization of the RAE2822 airfoil at the following conditions:

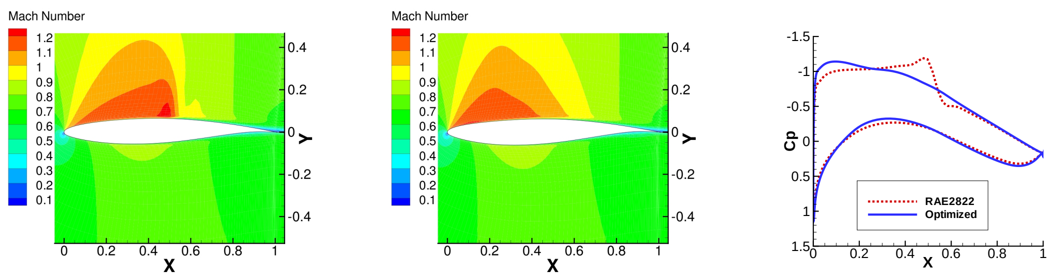
$$Ma = 0.729, \qquad Re = 7.0 \times 10^6 \qquad \alpha = 2.31^\circ.$$



(a) C-grid for 2-D Problem

(b) H-H grid for 3-D Problems

Figure 6. Grids for the Aerodynamic Shape Optimization



(a) RAE2822 Mach Contours

(b) Optimized Mach Contours

(c) C_p Plots

Figure 7. Comparison of the Original and Optimized Airfoils in Problem 3

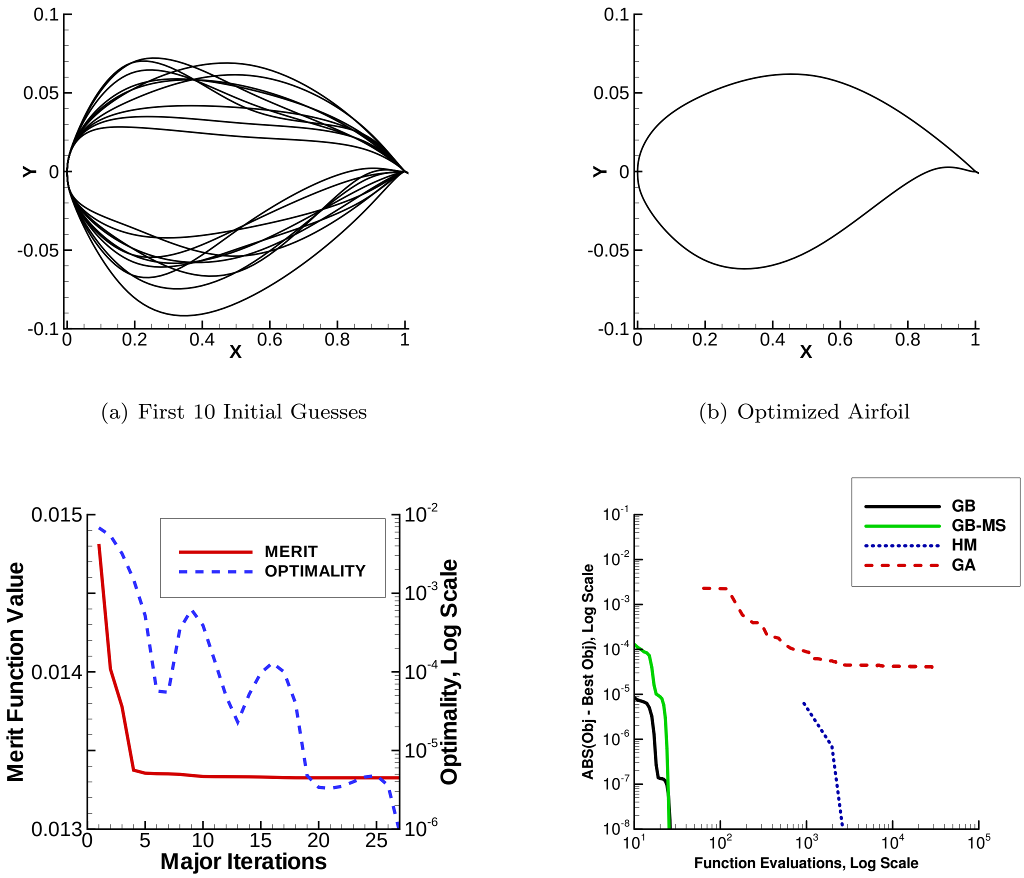
This is a common CFD validation test case.[30] We consider the lift-constrained drag-minimization problem with $C_l = 0.690$, which is the value obtained from the initial flow solve. The required lift coefficient is met by adjusting α, as described by Billing.[31] A minimum area constraint of 0.07772 is imposed, which is the area of the original airfoil.

The solution of the baseline RAE2822 geometry is illustrated in Fig. 7(a), where the Mach number contours are shown. The C_p plot is given in Fig. 7(c). A shock is evident on the upper surface, as seen in the experimental data.[30] The computed drag coefficient is $C_d = 0.014813$.

For the optimization, we use 6 control points as design variables: 3 on upper and 3 lower surfaces of the airfoil. The box constraints, placed on each design variable, are large enough such that no box constraint is active at the optimal point. For the GB-MS procedure, we consider 480 initial guesses, generated by the Sobol sequence. The first 10 initial guesses are shown in Fig. 8(a). For the HM optimizer, we limit SNOPT major iterations to 10 per generation. We use 100 generations for the hybrid method, and 1000 generations for the GA.

The convergence plot for the GB optimizer is shown in Fig. 8(c). The results clearly show that a local optimum has been reached, since the optimality measure is reduced to 10^{-6}. The Mach number contours of the optimized airfoil are shown in Fig. 7(b) and the C_p plot is provided in Fig. 7(c). The drag coefficient is reduced to $C_d = 0.013328$, an 11% improvement over the baseline value.

None of the other three optimization algorithms was successful in finding additional local optima. The convergence plots for the GB-MS, HM, and GA are shown in Fig. 8(d). The results indicate that the design space is unimodal, and the local optimum found using the RAE2822 as a starting point is, in fact, the global optimum. These results support some of the previous suggestions that this design space is unimodal.[8] We have performed similar studies varying the following parameters:



(a) First 10 Initial Guesses

(b) Optimized Airfoil

(c) GB Convergence Plots

(d) Comparison of Convergence Plots

Figure 8. Optimization Results for Problem 3

- Increased number of design variables (10 out of 15 control points are DVs),
- Different Mach numbers ($Ma = 0.50$ and $Ma = 0.85$), and
- Different objective function (C_d/C_l).

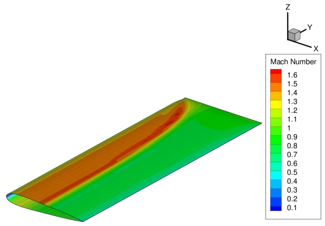
Only one local minimum was found in each of these studies. Therefore, our conclusion is that the most effective algorithm for this type of optimization problems is GB.

C. 3-D Wing Optimization Problems

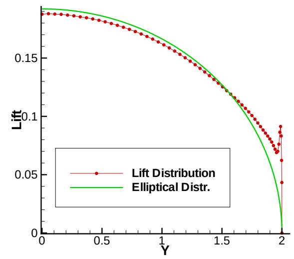
The 3-D ASO problems are solved using the inviscid Newton-Krylov-Schur solver (see Section II). In all cases, the mesh is a 12-block structured H-H topology grid with a total of 1,158,300 nodes, shown in Fig. 6(b). The wing is parameterized using a 2-patch B-spline surface. The baseline intial guess has NACA0012 cross-sections and a rectangular planform with the span of 2 and chord length of 2/3. Projected area is used as a reference and is restricted to 4/3 using a non-linear constraint.

Problem 4: Transonic Wing Section Optimization

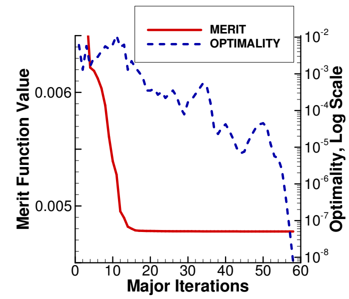
We perform an optimization of the wing sections with NACA 0012 cross-sections as the baseline. The objective is to minimize C_D at $Ma = 0.80$. The lift coefficient is constrained to $C_L = 0.2625$. The volume is constrained to be at least 6.57×10^{-2}, which is the volume of the original wing. We use projected area as the area reference, which stays constant throughout the optimization cycle.



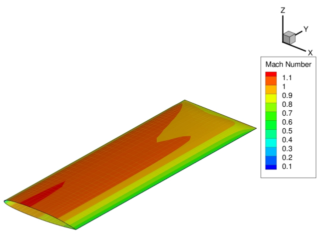
(a) Baseline Mach Contours



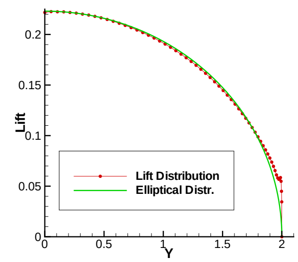
(b) Baseline Lift Distribution



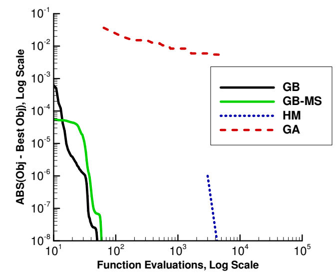
(c) GB Convergence Plot



(d) Optimized Mach Contours



(e) Optimized Lift Distribution



(f) Convergence Plots

Figure 9. Optimization Results for Problem 4

Each patch contains 5 control points in the streamwise direction and 6 in the spanwise direction. The control point at the root trailing edge is fixed. The control points at the root are not allowed to move in the spanwise direction. Accounting for the duplicate control points on the stitches, there are 125 geometric design variables (control point coordinates). The angle of attack is also a design variable. For this problem we are only optimizing cross-sections, thus allowing the interior control points to move only in the vertical direction. This is accomplished using linear constraints, as described in Section III.

The solution of the baseline geometry is plotted in Fig. 9(a), where the Mach number contours are shown. There is a relatively strong shock on the upper surface, and $C_D = 3.15845 \times 10^{-2}$.

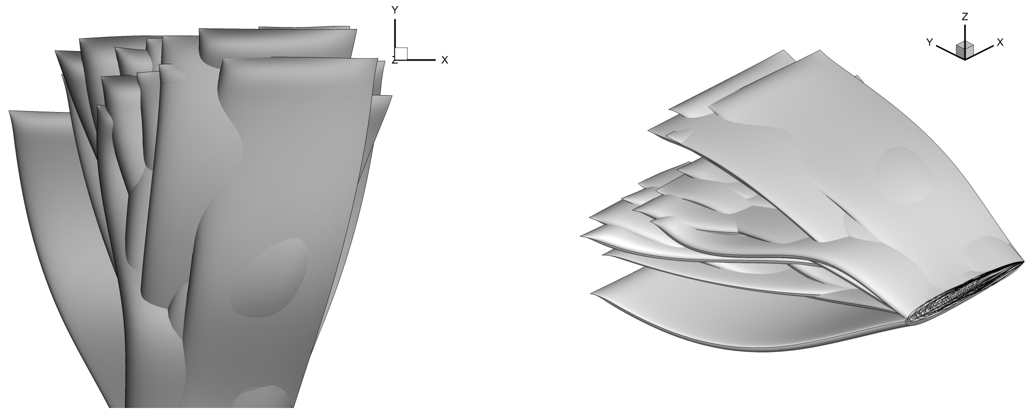
The GB optimization took 57 major iterations (60 function evaluations) to converge to within 10^{-8} optimality tolerance, which demonstrates that a local optimum has been reached. The convergence plot is shown in Fig. 9(c). C_D at the optimal point is 3.5818×10^{-3}.

In Fig. 9(d), we can see that the final geometry is shock-free. Figs. 9(b) and 9(e) compare the spanwise lift distributions for the optimal and baseline geometries, respectively. Note that for the optimized wing, the lift distribution is almost elliptical, except near the tip, where the wing deformation is not allowed by the linear constraints and side edge separation can be present. These results are consistent with lifting-line theory, which predicts that the induced drag is minimized when the lift distribution is elliptical in the spanwise direction. Therefore, the optimizer has successfully eliminated the shock and minimized induced drag within the constraints.

For the GB-MS method, 128 initial geometries, determined by the Sobol sequence, were optimized. No additional local optima were found for this problem and all optimizers converge to the same point. For the HM optimization, the population size is 64. The SNOPT major iterations limit is 50, and the algorithm was run for 10 generations. The local optimum was reached on the second generation. Subsequent generations failed to find any additional local optima.

The convergence plots for all optimizers are provided in Fig. 9(f). Note that for the GA, the first point on the plot is added at the end of the first generation, when 64 flow solves have already been performed. The same is true for the HM, except that at the end of the first generation, 3007 flow solves have been performed. This explains the gaps before the first points on the convergence plots.

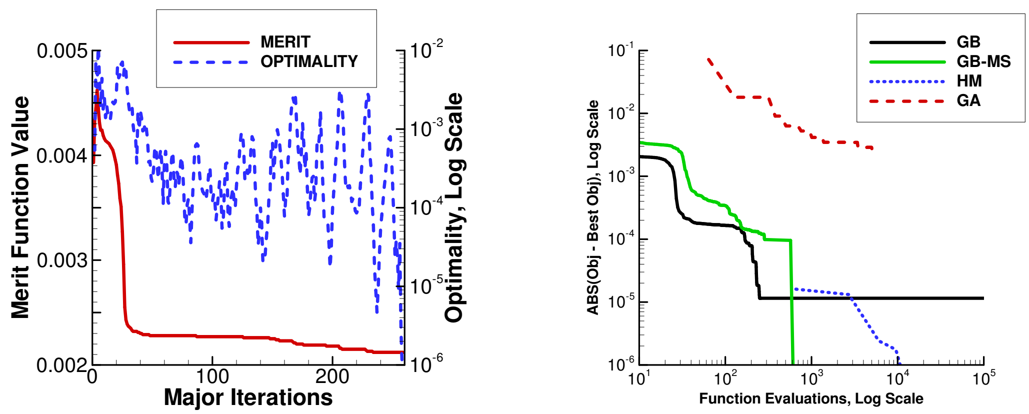
Since only one local optimum was found in this problem, we conclude that the design space is unimodal, and the GB method is the most effective for this type of ASO problem.



(a) Top View

(b) Isometric View

Figure 10. 32 Initial Guesses for Problems 5



(a) GB Convergence

(b) Convergence Plots

Figure 11. Convergence Plots for Problem 5

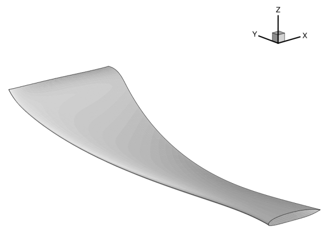
Problem 5: Subsonic Wing Optimization

In this problem we consider subsonic flow and allow greater geometric flexibility than in Problem 4. The objective is to minimize the drag coefficient C_D. The Mach number is 0.50. The lift coefficient C_L is constrained to 0.2625. The projected area is constrained to 4/3. The angle of attack is a design variable and can vary between $-3°$ and $+6°$. The volume is constrained to 6.57×10^{-2}, which is the volume of the baseline geometry. The control point at the trailing edge can have a maximum spanwise extent of 2.4, maximum sweep back to 1.00 (from the original value of 0.33), and vertical bounds are -0.3 and 0.3. Each section is allowed to twist and change its shape. Both leading and trailing edges can be curved.

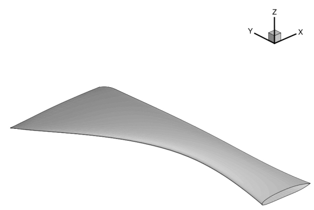
The first 32 initial geometries generated by the Sobol sequence are illustrated in Fig. 10, which shows that all essential areas of the design space have been covered (sweep forward/backward, dihedral up/down, winglet up/down, span long/short).

Since the flow is subsonic and inviscid, only induced drag is present. The baseline initial guess has $C_D = 4.0938 \times 10^{-3}$. Gradient-based optimization took 257 major iterations. After the optimization process C_D was reduced to 1.7444×10^{-3}. The convergence plot is provided in Fig. 11(a), and the optimized wing geometry is shown in Fig. 12(a).

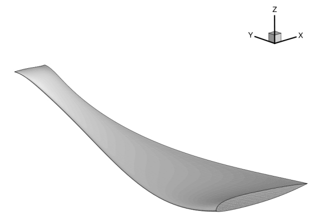
For the GB-MS method, 192 initial guesses, determined by the Sobol sequence, were optimized. The optimization process found 7 distinct local optima, all of which converged to the optimality tolerance of 1×10^{-8}. These optimal geometries are shown in Fig. 12. C_D values for all local optima found are compared



(a) Optimized Geometry 1

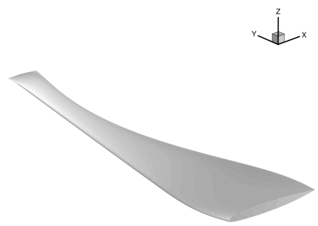


(b) Optimized Geometry 2

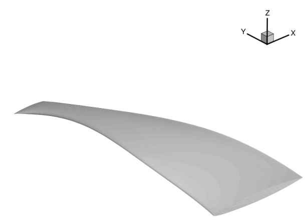


(c) Optimized Geometry 3

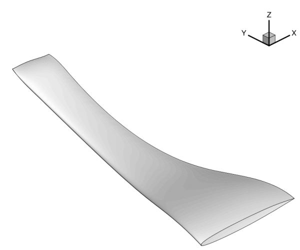
(d) Optimized Geometry 4



(e) Optimized Geometry 5



(f) Optimized Geometry 6



(g) Optimized Geometry 7

Figure 12. Local Optima for Problem 5

in Table 3. The results show that the baseline geometry does not lead to the global optimum, and that a further 0.54% drag improvement can be obtained by considering alternative initial geometries. It is also important to note the significant difference in performance at the local optima. The different local optima can differ by more than 5% in the objective value.

Local Optimum	C_D	**Relative Difference from Local Optimum 1**
1 (baseline optimum)	1.5884×10^{-3}	0.00%
2	1.5798×10^{-3}	−0.54%
3	1.6335×10^{-3}	2.84%
4	1.6208×10^{-3}	2.04%
5	1.6520×10^{-3}	4.00%
6	1.6444×10^{-3}	3.41%
7	1.6688×10^{-3}	5.06%

Table 3. C_D Values for Optimal Geometries in Problem 5

For the hybrid optimizer, the population size is 64. Each population member has a SNOPT major iteration limit of 50. The HM optimizer was run for 10 generations. As one can see in Fig. 11(b), HM is able to find the same best optimal point as the GB-MS, but it takes more function evaluations to reach that point. The GA was run for 100 generations. As shown in Fig. 11(b), the GA converges much more slowly than the other algorithms.

Since seven local optima were found in this problem, we conclude that the design space is somewhat multi-modal, and that the GB-MS optimizer is the most efficient algorithm for this type of problem.

Comments on Grid Refinement

Since induced drag is particularly sensitive to grid size, it is important to understand how the results are affected as the grid is refined. We address two questions:

- Do the optimized geometries change with the refined mesh?
- Do the C_D values change with the refined mesh?

In order to answer the first question, we increased the number of nodes in the original mesh by a factor of 2 in all directions. The resulting grid has 12 blocks and 8,955,180 nodes. The optimization process was started from the local optimum found using the original grid size of 1,158,300. The optimizer is able to converge these geometries to an optimality tolerance of 10^{-6}, so local minima are found again. The optimized geometries on the fine grid are very similar to the geometries on the coarser grids. Therefore, it appears that the optimal shapes are grid-independent.

In order to answer the second question, we increased the number of nodes in the original mesh by a factor of 4 in all directions. Each block was split into 64 blocks. The resulting grid has 70,416,972 nodes with 768 blocks. Flow solutions were computed for three of the optimized geometries on this grid. The values of C_L, C_D, and the span efficiency factor e are compared in Table 4. It appears that the values of C_D change significantly as the grid is refined. Since the values of C_L change slightly, the relative performance of these wing geometries is best assessed in terms of the span efficiency factor e. The relative performance at the local optima remains consistent (i.e. optimized geometries are superior to the baseline geometry, local optimum 2 is better than local optimum 7, etc.).

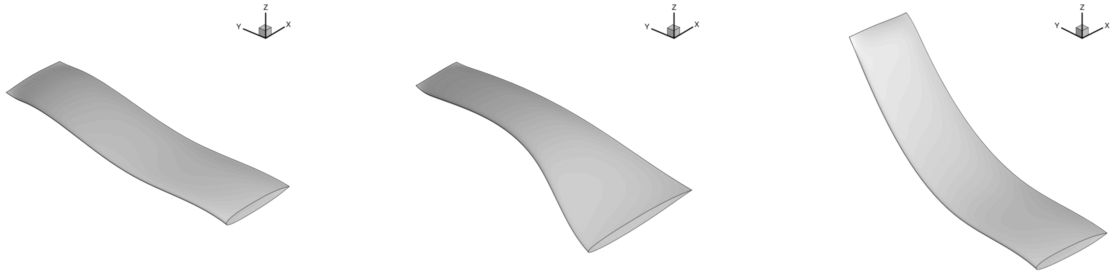
Quantity	C_L		C_D		e	
Mesh Size	1,158,300	70,416,972	1,158,300	70,416,972	1,158,300	70,416,972
Baseline Geometry	0.2625	0.2659	4.0938×10^{-3}	3.821×10^{-3}	0.8929	0.9814
Local Optimum 1	0.2625	0.2639	1.5884×10^{-3}	2.354×10^{-3}	1.5982	1.0894
Local Optimum 2	0.2625	0.2623	1.5798×10^{-3}	2.318×10^{-3}	1.6069	1.0935
Local Optimum 7	0.2625	0.2611	1.6688×10^{-3}	2.331×10^{-3}	1.5213	1.0773

Table 4. Grid Refinement Results

Problem 6: Transonic Wing Optimization

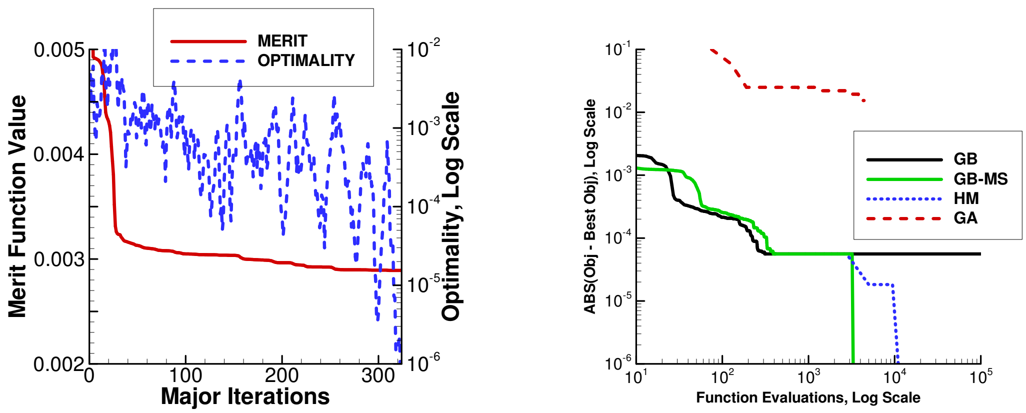
The problem setup (design variables, objective, and constraints) is the same as that of Problem 5; however, the transonic flow regime is considered here, with $Ma = 0.80$. The solution of the baseline geometry shows that a shock is present on the upper surface of the wing (see Fig. 9(a)). The GB optimization process required 323 major iterations (383 function evaluations) to converge. The convergence plot in Fig. 14(a) shows that the optimality conditions are satisfied to within 10^{-6}. After the GB optimization, the spanwise extent of the wing is maximized, the wing is swept back and the vertical extent of the tip maximized as well. Therefore, the box constraints controlling the motion of the tip trailing edge are active. The optimized geometry is shown in Fig. 13(a). The drag coefficient was reduced from 3.15845×10^{-2} to 2.16828×10^{-3}, a 93% reduction.

For the GB-MS optimization, 192 initial geometries were optimized. The optimizer found a total of 3 local optima, with the corresponding geometries plotted in Fig. 13. The objective values at these local optima are compared in Table 5. The local optimum obtained using the GB optimizer from the baseline



(a) Optimized Geometry 1 (b) Optimized Geometry 2 (c) Optimized Geometry 3

Figure 13. Local Optima for Problem 6



(a) GB Convergence (b) Convergence Plots

Figure 14. Convergence Plots for Problem 6

geometry is not the global optimum, and a further 1.93% reduction in drag coefficient value is possible using a global optimization algorithm.

Local Optimum	C_D	Relative Difference from Local Optimum 1
1 (baseline optimum)	2.1682×10^{-3}	0.00%
2	2.1263×10^{-3}	-1.93%
3	2.1341×10^{-3}	-1.57%

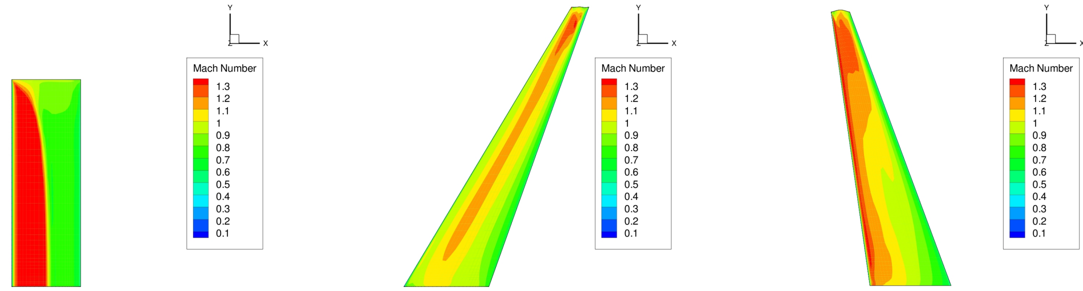
Table 5. C_D Values for Optimal Geometries in Problem 6

For the HM and GA optimizers, the population size was 64. The HM optimizer used 50 gradient-based iterations within each generation, and was run for 10 generations. This algorithm eventually converged to the best local optimum found using the GB-MS procedure, but it required more function evaluations. The GA was unable to find any of these local optima after 100 generations. Fig. 14(b) compares the convergence plots for all optimizers. The GB-MS algorithm requires the fewest function evaluations to converge to the best local optimum.

Since only three local optima were found in this problem, we conclude that the problem is somewhat multi-modal and that the GB-MS optimizer is the most efficient for this type of problem.

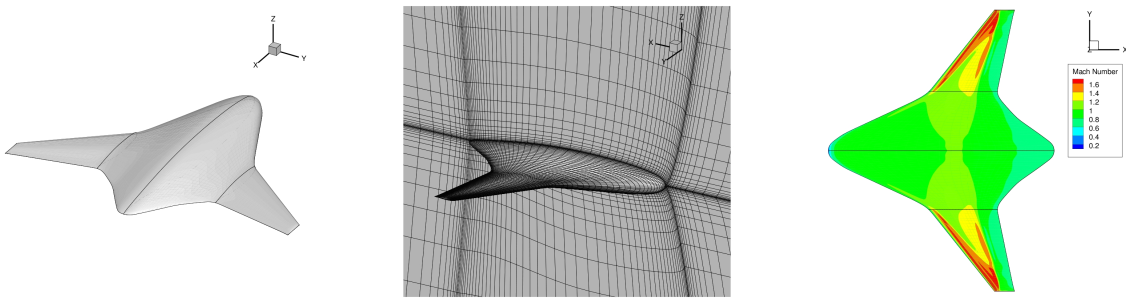
Problem 7: Planar Wing Optimization

The wing shapes obtained in Problems 5 and 6 are intuitively impractical. Structural requirements, performance across multiple cruise conditions, and other considerations are not taken into account. Future work



(a) Baseline Geometry (b) Optimized Geometry 1 (c) Optimized Geometry 2

Figure 15. Mach Number Contours for Geometries in Problem 7



(a) Baseline BWB Geometry (b) H-H grid for BWB Problem (c) Mach Contours, Top Planform View

Figure 16. Baseline Geometry for Problem 8

will incorporate these constraints into the optimization process.

In the meantime, the optimization framework developed allows one to reduce geometric flexibility in order to design more practical wing shapes. In this problem we keep the leading and trailing edges straight and disallow the vertical displacement of the wing (i.e. planar wing optimization only). Changes in wing cross-sections are allowed, as well as the variations in the span along the tip. Linear twist is permitted. The taper ratio is allowed to change as well.

The Mach number is 0.80. The B-spline patches have 7 control points in both spanwise and streamwise directions. The total number of design variables is 224. The objective and the constraints (volume, projected area, lift coefficient) are the same as in the previous three problems.

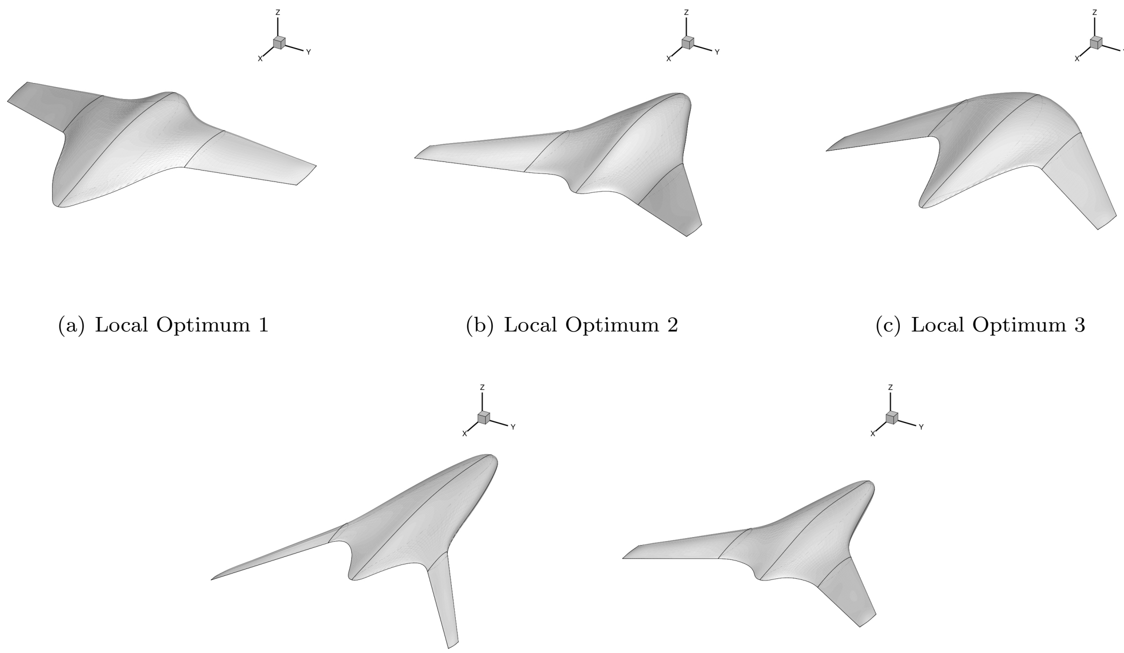
Since the multi-start algorithm has proven to be the most efficient algorithm for assessment of multi-modality, only GB-MS is used in this problem. Ten initial geometries are converged to optimality within the tolerance of 10^{-6}. Two local optima were found. The drag coefficient is reduced from 3.158×10^{-2} for the baseline geometry to 1.460×10^{-3} for the first local optimum and 1.786×10^{-3} for the second local optimum. Note that a better objective value is achieved (compared to Problem 6) because of parameterizing the geometry with a larger number of control points and increasing the span limit from 2.4 to 2.7.

The Mach number contours for the baseline and optimal geometries are plotted in Fig. 15. Not surprisingly, this problem has fewer local optima than the previous two problems. Since only two optima were found, we conclude that this problem is somewhat multi-modal and that GB-MS is the preferred optimization algorithm.

D. Problem 8: Blended-Wing-Body Aircraft Optimization

A Blended-Wing-Body (BWB) is an unconventional configuration that can reduce fuel burn by as much as 30%, compared to a conventional wing-tube configuration.[32] In this problem we apply the GB-MS optimization algorithm to a BWB configuration.

The baseline geometry is shown in Fig. 16(a). The geometry is parameterized with 4 B-spline patches. Each patch has 7 control points in streamwise and 6 in spanwise directions. The total number of design



(a) Local Optimum 1 (b) Local Optimum 2 (c) Local Optimum 3

(d) Local Optimum 4 (e) Local Optimum 5

Figure 17. Local Optima for BWB Problem

variables is 368. A system of linear constraints was created to control the geometric deformation. The mesh, shown in Fig. 16(b), is an H-H topology grid with 627,000 nodes. The drag coefficient value for the baseline geometry is $C_D = 1.849 \times 10^{-2}$. The top planform view with the Mach number contours is displayed in Fig. 16(c), showing a shock on the upper surface.

The objective is to minimize C_D with C_L constrained to 0.3522. The projected area is constrained to 0.24133, and the Mach number is 0.80. We allow variation in the sweep, dihedral, and linear twist of the wing, as well as section changes on both wing and body. Leading and trailing edges are straight. The span is constrained to the initial value.

We used the GB-MS optimization algorithm on 16 initial geometries. Five local optima were found, although difficulties were encountered in converging the cases below a 10^{-6} optimality tolerance. This is likely a result of deteriorating mesh quality, which is a drawback of allowing considerable geometric flexibility. Future work will focus on limiting allowed deformations and improving the quality of initial mesh. All constraints are satisfied to within a 10^{-6} feasibility tolerance. The geometries corresponding to the local optima are plotted in Fig. 17. The C_D values differ by nearly 4% from the best local optimum (Fig. 17(d)) with $C_D = 1.241 \times 10^{-2}$, to the worst one (Fig. 17(a)) with $C_D = 1.292 \times 10^{-2}$.

Local Optimum	C_D	**Relative Difference from Local Optimum 1**
1	1.2920×10^{-2}	0.00%
2	1.2544×10^{-2}	-2.91%
3	1.2492×10^{-2}	-3.31%
4	1.2407×10^{-2}	-3.98%
5	1.2505×10^{-2}	-3.21%

Table 6. C_D Values for Optimal Geometries in Problem 8

This study is only preliminary and further investigation is required to assess the performance of the BWB configuration. The main purpose here is to underscore the importance of considering a global optimization method. It appears that this problem is at least somewhat multi-modal and possibly moderately multi-modal. More initial geometries need to be considered by the GB-MS method. The HM method may be suitable for this problem; this will be investigated in the near future. However, the results of the GB-MS method are quite informative to establish the multi-modality of such problems.

V. Conclusions

The results show the shortcomings of using either pure gradient-based or gradient-free optimization algorithms for high-fidelity aerodynamic shape optimization. A purely gradient-based optimizer may converge to a local optimum with an objective value significantly higher than that of another local optimum in the same design space. On the other hand, the computational expense of using a gradient-free optimization algorithm is unacceptable for many problems, especially if the number of design variables is large.

We have developed two optimization algorithms to avoid these shortcomings. The algorithms are efficient for global optimization of smooth, differentiable objective functions with multiple local optima. The linear constraints allow the size of the problem to be reduced and ensure only feasible geometries are considered. The sampling method uses a Sobol sequence to cover the design space in a uniform and unbiased manner.

The optimization results show that multi-modality is not a predominant feature of the design spaces for many ASO problems. None of the problems considered are highly multi-modal, as defined in Table 1. The perceived multi-modality may be a result of insufficient optimality tolerance or inaccuracies in gradient computations. However, multiple local optima do exist, and a global optimization method is essential to find the most efficient aerodynamic configuration.

Problems 3 and 4 are unimodal, and a GB method is preferred for optimization, although the use of GB-MS procedure is informative to ensure that no other local optima exist. Problems 5, 6, and 7 are somewhat multi-modal, and the GB-MS method is most efficient. Problem 8 also appears to be somewhat multi-modal, although further investigation is required to establish the degree of multi-modality of this case.

Future work will include testing the multi-modality of ASO problems in viscous and turbulent flows, and further studies of unconventional aerodynamic configurations. Adding structural requirements for high-fidelity aerostructural optimization will also be considered in the future in terms of its impact on multi-modality.

Acknowledgments

The authors would like to thank T. Pulliam for help with the genetic algorithm, F. Kuo for providing directional numbers for Sobol sequences.

Computations were performed on the GPC supercomputer at the SciNet HPC Consortium. SciNet is funded by: the Canada Foundation for Innovation under the auspices of Compute Canada; the Government of Ontaio; Ontario Research Fund - Research Excellence; and the University of Toronto.

The funding of the first author by MITACS and Bombardier Aerospace and the second author by the Natural Sciences Research Council of Canada and the Canada Research Chairs program is gratefully acknowledged.

References

[1] Nemec, M. and Zingg, D. W., "Newton-Krylov Algorithm for Aerodynamic Design Using the Navier-Stokes Equations," *AIAA Journal*, Vol. 40, No. 6, June 2002, pp. 1146–1154.

[2] Marco, N., Lanteri, S., Desideri, J.-A., Mantel, B., and Periaux, J., "Parallel Genetic Algorithms Applied to Optimum Shape Design in Aeronautics," *Third International Euro-Par Conference*, Springer, Passau, Germany, August 1997, pp. 856–863.

[3] Vicini, A. and Quagliarella, D., "Airfoil and Wing Design Through Hybrid Optimization Strategies," Tech. Rep. 1998-2729, American Institute of Aeronautics and Astronautics, June 1998.

[4] Ong, Y. S., Lum, K. Y., and Nair, P. B., "Hybrid Evolutionary Algorithm with Hermite Radial Basis Function Interpolants for Computationally Expensive Adjoint Solvers," *Comput. Optim. Appl.*, Vol. 39, No. 1, January 2008, pp. 97–119.

[5] Nocedal, J. and Wright, S. J., *Numerical Optimization*, Springer-Verlag, Berlin, 2nd ed., 2006.

[6] Obayashi, S. and Tsukahara, T., "Comparison of Optimization Algorithms for Aerodynamic Shape Design," *AIAA Journal*, Vol. 35, No. 8, August 1997, pp. 1413–1415.

[7] Jones, B. R., Crossley, W. A., and Lyrintzis, A. S., "Aerodynamic and Aeroacoustic Optimization of Airfoils Via a Parallel Genetic Algorithm," Tech. Rep. 1998-4811, American Institute of Aeronautics and Astronautics, September 1998.

[8]Zingg, D. W., Nemec, M., and Pulliam, T. H., "A Comparative Evaluation of Genetic and Gradient-Based Algorithms Applied to Aerodynamic Optimization," *REMN*, Vol. 17, No. 1, 2008, pp. 103–126.

[9]Laurenceau, J., Meaux, M., Montagnac, M., and Sagaut, P., "Comparison of Gradient-Based and Gradient-Enhanced Response-Surface-Based Optimizers," *AIAA Journal*, Vol. 48, No. 5, May 2010, pp. 981–994.

[10]Namgoong, H., Crossley, W., and Lyrintzis, A. S., "Global Optimization Issues for Transonic Airfoil Design," Tech. Rep. 2002-5641, American Institute of Aeronautics and Astronautics, September 2002.

[11]Keane, A. J. and Nair, P. B., *Computational Approaches for Aerospace Design*, John Wiley & Sons, Ltd, 2005.

[12]Buckley, H. P., Zhou, B. Y., and Zingg, D. W., "Airfoil Optimization Using Practical Aerodynamic Design Requirements," *Journal of Aircraft*, Vol. 47, No. 5, 2010, pp. 1707–1719.

[13]Leung, T. and Zingg, D. W., "Single- and Multipoint Aerodynamic Shape Optimization Using a Parallel Newton-Krylov Approach," Tech. Rep. 2009-3803, American Institute of Aeronautics and Astronautics, June 2009.

[14]Hicken, J. E. and Zingg, D. W., "Induced-Drag Minimization of Nonplanar Geometries Based on the Euler Equations," *AIAA Journal*, Vol. 48, No. 11, November 2010, pp. 2564–2563.

[15]Epstein, B., Jameson, A., Peigin, S., Roman, D., Harrison, H., and Vassberg, J., "Comparative Study of Three-Dimensional Wing Drag Minimization by Different Optimization Techniques," *Journal of Aircraft*, Vol. 46, No. 2, 2009, pp. 526–545.

[16]Nemec, M., Zingg, D. W., and Pulliam, T. H., "Multipoint and Multi-Objective Aerodynamic Shape Optimization," *AIAA Journal*, Vol. 42, No. 6, 2004, pp. 1057–1065.

[17]Hicken, J. E. and Zingg, D. W., "Parallel Newton-Krylov Solver for the Euler Equations Discretized Using Simultaneous-Approximation Terms," *AIAA Journal*, Vol. 46, No. 11, November 2008, pp. 2773–2786.

[18]Hicken, J. E. and Zingg, D. W., "Aerodynamic Optimization Algorithm with Integrated Geometry Parameterization and Mesh Movement," *AIAA Journal*, Vol. 48, No. 2, February 2010, pp. 401–413.

[19]Truong, A. H., Oldfield, C., and Zingg, D. W., "Mesh Movement for a Discrete-Adjoint Newton-Krylov Algorithm for Aerodynamic Optimization," *AIAA Journal*, Vol. 46, No. 7, 2008, pp. 1695–1704.

[20]Osusky, M., Hicken, J. E., and Zingg, D. W., "A Parallel Newton-Krylov-Schur Flow Solver for the Navier-Stokes Equations Using the SBP-SAT Approach," Tech. Rep. 2010-116, American Institute of Aeronautics and Astronautics, January 2010.

[21]Nelson, A., Alonso, J., and Pulliam, T., "Multi-Fidelity Aerodynamic Optimization Using Treed Meta-Models," Tech. Rep. 2007-4057, American Institute of Aeronautics and Astronautics, June 2007.

[22]Joe, S. and Kuo, F., "Remarks on Algorithm 659: Implementing Sobol's quasirandom sequence generator," *ACM Transactions on Math. Software*, Vol. 29, No. 1, March 2003, pp. 49–57.

[23]Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: an SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Journal on Optimization*, Vol. 12, No. 4, 2002, pp. 979–1006.

[24]Morris, A. M., Allen, C. B., and Rendall, T. C. S., "Domain-Element Method for Aerodynamic Shape Optimization Applied to a Modern Transport Wing," *AIAA Journal*, Vol. 47, No. 7, July 2009, pp. 1647–1659.

[25]Nelson, T. E. and Zingg, D. W., "Fifty Years of Aerodynamics: Successes, Challenges, and Opportunities," *Canadian Aeronautics and Space Journal*, Vol. 50, No. 1, March 2004, pp. 61–84.

[26]Jansen, P. W., Perez, R. E., and Martins, J. R. R. A., "Aerostructural Optimization of Non-Planar Lifting Surfaces," *Journal of Aircraft*, Vol. 47, No. 5, 2010, pp. 1490–1503.

[27]Holst, T. L. and Pulliam, T. H., "Evaluation of Genetic Algorithm Concepts Using Model Problems Part I: Single-Objective Optimization," NASA/TM 2003-212812, National Aeronautics and Space Administration, December 2003.

[28]Lehner, S. and Crossley, W. A., "Hybrid Optimization for a Combinatorial Aircraft Design Problem," Tech. Rep. 2009-7116, American Institute of Aeronautics and Astronautics, September 2009.

[29]Locatelli, M., "A Note on the Griewank Test Function," *Journal of Global Optimization*, Vol. 25, No. 2, February 2003, pp. 169–174.

[30]"RAE 2822 Transonic Airfoil," http://www.grc.nasa.gov/WWW/wind/valid/raetaf/raetaf01/raetaf01.html, 2008, [Online; accessed 10-Nov-2010].

[31]Billing, L., *On the Development of an Improved Lift-Constrained Aerodynamic Optimization Algorithm*, Master's thesis, University of Toronto, 2006.

[32]Postdam, M. A., Page, M. A., and Liebeck, R. H., "Blended Wing Body Analysis and Design," Tech. Rep. 1997-2317, American Institute of Aeronautics and Astronautics, June 1997.