Managing Gradient Inaccuracies while Enhancing Optimal Shape Design Methods

Trent Lukaczyk*, Francisco Palacios†, and Juan J. Alonso‡
Stanford University, Stanford CA 94305, USA

A major focus of aircraft design is the enhancement of CFD-based optimal shape design methods with improved solution accuracies from mesh adaptation, or efficient gradient calculations from adjoint formulations. The key goal of these enhancements is to increase the accuracy of the solution while reducing the computational wall-time. This study is specifically interested in quantifying the impact of mesh adaptation and approximate gradients from continuous adjoint methodologies while performing Gradient Based Optimization (GBO) or Surrogate Based Optimization (SBO).

In the course of this work we have discovered conditions in which these various gradient methods can actually degrade the performance of the optimizer. For example, we have observed that bias errors from continuous adjoint gradients, which are traditionally acceptable for GBO methods, are not acceptable for basic SBO methods, which make a stronger assumption of objective-gradient correlation. We have also observed that applying mesh adaptation to continuous adjoint solutions can exacerbate this error enough to effect GBO convergence rates.

In attempting to improve the convergence of the optimizers, we have built several approaches to better condition gradient accuracies. In one approach we filter the surface sensitivities before projecting them into a parameterized design space. In another approach, we build surrogate models capable of learning the noise of the system. This paper will present the work completed towards developing these methods, and will provide examples in the form of analytical test cases and demonstrative aerodynamic problems.

I. Introduction

A. Motivation: Supersonic Passenger Jets

Under NASA's Supersonics Project, Stanford University is collaborating with the Lockheed Martin Corporation to advance the state of supersonic, low-boom passenger jet technology. Stanford University is contributing to this effort by developing a generic Multi-Disciplinary Analysis and Optimization (MDAO) framework capable of identifying aircraft designs that demonstrate increased fuel efficiency by reducing drag and reduced sonic boom loudness using techniques for optimal shape design.

The motivating work of this paper is focused on the shape design of the N+2 Supersonic Passenger Jet. It is specified to fly at Mach 1.6-1.8 with a range of 4000+ nmi and a capacity of approximately 35-70 passengers. The loudness target is a perceived loudness level of 85 PLdB, which is slightly above the level of conversational noise. The cruise NOx emissions target is lower than 10 g NOx / kg of fuel burned, and the fuel efficiency target is more than 3 pax-mi / lb of fuel burned.[1] Our contributions are meant to help directly with the range and sonic boom requirements, and indirectly with cruise emissions and the fuel efficiency target.

In previous work not presented in this paper, we approach this problem using inverse shape design on an azimuthal series of target equivalent area distributions.[2] To analyze the boom loudness and drag, we are performing high fidelity CFD simulations in the near field and extracting a pressure signature two body-lengths below the aircraft. Using classical linearized supersonic aerodynamics theory,[3] we convert this

*Graduate Student, Department of Aeronautics and Astronautics, AIAA Member
†Engineering Research Associate, Department of Aeronautics and Astronautics, AIAA Member
‡Associate Professor, Department of Aeronautics and Astronautics, AIAA Senior Member

signature into the equivalent area distribution of a body of revolution, which can then be compared to a target distribution known to have low perceived loudness.[1] The shape of the aircraft is changed using arbitrary Free Form Deformation (FFD) control boxes to drive the aircraft's equivalent area distribution to match the target. This inverse design approach is constructed to ensure a smooth behavior of the high-fidelity design problem.[2]

We are using our open-source solver, SU^2, for all direct and adjoint calculations.[4] SU^2 is a general purpose partial differential equation solver equipped with tools for optimal shape design including direct flow and adjoint solvers, free-form mesh deformation, and goal-oriented adaptive mesh refinement. These tools are wrapped in python to efficiently manage the input and output of data.

B. Methods: Optimization Approaches

To address the high computational cost of the CFD simulations and to enable global design space optimization, we are using Response Surface Modeling (RSM) with gradient-enhanced Gaussian Process Regression (GPR). Gradient information available from an adjoint solution is used to increase the accuracy of the RSM at low computational cost. A large body of work has been built around RSM techniques, especially using a stochastic modeling technique known as Kriging[5,6] and its gradient enhanced relative known as Co-Kriging.[7,8] GPR is a superset of Kriging, and is formulated by conditioning a probability distribution over random functions. In general both GPR and Kriging result in the same mathematical fitting models.[6,9] However, GPR is built within the context of machine learning, which allows more flexibility when dealing with complex design spaces.[10]

Response surface models can be used in what is known as Surrogate-Based Optimization (SBO). To generate an initial response surface, a set of designs are sampled using Design of Experiment (DOE) techniques such as Latin Hypercube Sampling. Because the locations of these points are not dependent on each other, they can be run in parallel, dramatically reducing wall-clock time if the resources are available. After this initial sample, additional design points are chosen using an Infill Sampling Criterion (ISC). It is common to take advantage of the uncertainty information available from stochastic response surface models like GPR to efficiently add points that attempt to improve the accuracy of the model near regions of optimal design.[11]

An equally large body of work has been built around surrogate-based optimization. Forrester provides a thorough summary of different infill sampling criterion.[12] Several studies describe it's application to aircraft design problems.[13–16] In previous work, we explored the use of ISCs to further improve the SBO process using a hybrid infill sampling criteria with expected improvement and estimated optimum.[17]

Alternatively, we are able to apply Gradient Based Optimization (GBO) to solve for local minima. Compared to SBO, GBO can require fewer function evaluations in higher dimensional design spaces at the cost of performing local optimization. In this study, we use a sequential quadratic programming (SLSQP) optimizer built into the Scientific Python (scipy) toolbox.[18–20] At every iteration, this code chooses a search direction based on the constrained optimization of a second-order least-squares response surface (the quadratic program). It then performs a line search in this direction to find a point that satisfies first and second order optimality conditions. At each major iteration the response surface is updated with the BFGS update rule. Convergence is declared when improvement of the objective, or norms of the gradients and Hessian fall below a tolerance.[19,20]

Both optimization methods rely on the calculation of sensitivities of the objective and constraints in the design space. In this paper we examine the use of adjoint formulations and finite differencing to perform sensitivity analyses.

While adjoints are efficient and accurate, they build on discretization approximations. Two formulations for adjoints currently exist. Both reformulate the direct solution to solve for the sensitivity of functions of the flow to an input such as geometry variations. The "discrete" adjoint is built completely on top of the direct discretized solution. This results in a numerical model of the sensitivity in the discretized flow solution. The"continuous" adjoint is built from the original governing partial differential equations (PDE). This results in a numerical model of the sensitivity of the physical flow. While it still depends on the direct solution, the result of a continuous adjoint is expected to be more physically exact than the discrete adjoint because it relates to the actual solution and gradients of the continuous PDE.[21]

In the case of an infinitely refined mesh, the continuous and discrete adjoints will yield the same solution and gradient information. However, most design problems will only allow a practically-refined mesh. In this case, the discrete adjoint will be a better estimate for the sensitivity in the *numerical* flow solution, and the continuous adjoint will be a better estimate for the sensitivity in the *physical* flow.[22]

The finite differencing approach is commonly used as a reference for gradient accuracy in the absence of discrete adjoint or complex step methods. It is a more expensive approach but numerically similar to the discrete adjoint. An important parameter to be chosen for finite differencing is the order of the step size. Automatic finite differencing is one method that approaches this by maximizing accuracy while minimizing truncation error.

The sensitivity of N objective to M design variable requires the solution of $N \times M$ additional mesh perturbations and flow solutions for a first order approximation. However because this estimate of the gradient is built on the direct solution only, it will be more closely correlated.

C. Problem: Gradient De-correlation

A key assumption in gradient-enhanced Response Surface Modeling that is important to the current work is that the correlation of all input information can be modeled by a covariance function. Typically when performing gradient enhanced RSM with Kriging or GPR, an exact correlation model is used to relate the function and its gradients.[8] As identified by Dwight,[23] violations of this model have adverse effects on the quality of the fit.

De-correlation of objectives and gradients can effect gradient based optimization as well. However, many modern methods use approximate Hessians with an under-fitting surrogate model such as a quadratic polynomial, which expresses a weak assumption of the behavior of the data in small regions. Along with various relaxation techniques, this makes GBO more robust, but not insensitive, to gradient errors.

The key problem that we will identify, explore and address in this paper is the quality degradation of optimization methods that occurs when using function and gradient data that are de-correlated because they possess varying levels of "physical" exactness.

Here we define physical exactness as the ability of the numerical model to describe the real physical flow. We include governing flow equations, adjoint equations, and finite differencing in this set of numerical models. Each numerical model has limits of physical exactness, based on for example numerical scheme or step size. In this limit, we describe a digit of precision as "physically representative" if an engineer would use it to make a decision between two candidate designs. Experience is largely a determining factor in setting the maximum level of physical representivity. For example, in the case of the Euler simulation of a transonic NACA0012 airfoil (an example we will rely on heavily for this paper), it is uncommon to use estimates of drag coefficient more accurate than 4 digits of precision.

There are four cases that we have identified where variations in physical exactness can potentially affect the convergence of SBO and GBO methods.

First, the continuous adjoint is commonly used because it is simpler to derive than the analytical discrete adjoint, and carries less numerical cost than the discrete adjoint.[22] In the case of a practically refined mesh, the sensitivity from the continuous adjoint will not be well correlated to the direct flow solution because it is based on the physical, not the numerical flow solution. Moreover, surface formulations for the computation of the gradients can greatly reduce the computational cost (by avoiding the need for perturbing the volume mesh) but can also introduce slight errors in finite-size meshes.

Second, it is a natural motive of automatic finite-differencing approaches to minimize step size. However, these methods are not able to detect the limits of physical models which may produce gradients that are numerically accurate in the limit of small step size, but with large physical inaccuracy (or at least not correlated with the physical model).

Third, by adding smaller and smaller cells to the mesh, gradient-based mesh refinement adds high-frequency components to the flow features. In the case of the direct solution, this allows it to sharpen discontinuities such as shocks. In the case of the continuous adjoint solution however, these high-frequency components violate a strong assumption of smoothness and thus introduce inaccuracies.

Fourth, while gradient-based mesh refinement will tune the direct solution for numerical exactness, a similar refinement on the adjoint solution could result in a very different mesh because the adjoint field has different features. It may then be the case that accuracy of the sensitivity information relative to the direct solution is sacrificed, unless the adjoint solution is refined as well.

These factors will negatively effect the performance of a response surface and can hamper the convergence of SBO and GBO optimizations without methods to manage gradient inaccuracies.

II. Theoretical Background

The following sections will demonstrate the theory behind our approaches to managing gradient inaccuracies. First we will present a short background on a our surface formulation for continuous adjoint problems. We will then describe a filtering approach for correcting surface sensitivity information from adjoint solutions, and a two-step refinement approach for direct and continuous adjoint solutions.

Finally, we will present a response surface model based on GPR in which we identify where the assumption of function-to-gradient correlation is made. Accompanying this will be a description of a noise regression method that could be used to address the current problem where inaccurate gradients violate this assumption.

A. Adjoint Approach for Estimating Gradients

1. Introduction to Adjoint Methods

The adjoint method was originally applied to aerodynamics by Jameson,[24] adapting ideas originally formulated by Lions[25] on optimal control of systems governed by partial differential equations (PDEs). The adjoint equations can be conveniently formulated in a framework to calculate the sensitivity of a given objective function $\mathcal{J}$ (f for the RSM problem) to parameters α in a problem governed by the set of equations which can be represented by $\mathcal{G}(U, \alpha) = 0$, where U is the solution.

The additional computational cost of solving the adjoint problem is generally of the order of one additional flow solution, and the adjoint variables represent the sensitivities of $\mathcal{J}$ to changes in all of the parameters that define the problem at every point in the domain. In contrast, though finite difference methods can also be used to find these sensitivities, they are in general significantly more expensive, requiring at least one additional flow solution to find the gradient of the objective function to each parameter in the domain.

There are two main ways to characterize the adjoint approach, as a discrete method, in which the discretized governing equations are used to derive the adjoint equations, and as a continuous method, in which the adjoint equations are derived from the analytical PDEs. The adjoint equations for both approaches can be derived in two different ways, via the Primal-Dual Equivalence Theorem or via an optimization framework using Lagrange multipliers,[21,26] but in this paper we introduce only the latter method, presenting both the discrete and continuous derivations via the same general approach. This follows the convention used by Taylor.[27]

The discrete and continuous approaches are found to have relative advantages and disadvantages over each other. In theory a discrete method can handle PDEs of arbitrary complexity without significant mathematical development and can treat arbitrary functionals. However, this method requires the evaluation of discrete Jacobians, which we denote as $\frac{\mathfrak{D}}{\mathfrak{D}}$ to distinguish from their continuous alternatives $\frac{\partial}{\partial}$, and there are two main ways to do this. The first is to analytically derive these terms from the discretized forms of the flow residuals and then develop code based on this, and the second it to use algorithmic Automatic Differentiation, either via source code transformation[28] or operator overloading.[29] The former approach requires significant development, more than that generally required in the continuous method,[22] while the latter is computationally expensive, with large memory storage requirements.

In comparison, the continuous adjoint requires significant theoretical development but is better connected to the underlying physics and can be solved in a method independent of the flow solution scheme. However, it is more limited in the types of functionals and governing equations that can be treated, and the gradient calculated will differ more substantially from that found by finite differencing, although as the mesh is refined, all three gradients, discrete, continuous and finite difference, should converge.

2. Continuous Adjoint Approach for Surface Deformations

We provide a brief overview of the development of our continuous adjoint approach for calculating performance variable sensitivities with respect to surface deformations.[4] The sensitivity of the numerical residual R to a parameter α can be written as:

$$\frac{dR}{d\alpha} = \frac{\partial R}{\partial U}\frac{dU}{d\alpha} + \frac{\partial R}{\partial \alpha} = 0, \tag{1}$$

rearranging this last equation gives:

$$\frac{dU}{d\alpha} = -\left(\frac{\partial R}{\partial U}\right)^{-1}\frac{\partial R}{\partial \alpha}. \tag{2}$$

In a similar way the sensitivity of the objective function J to the parameter α can be written as

$$\frac{dJ}{d\alpha} = \frac{\partial J}{\partial U}\frac{dU}{d\alpha} + \frac{\partial J}{\partial \alpha}, \tag{3}$$

where using the result from Eq. 2 gives

$$\frac{dJ}{d\alpha} = -\frac{\partial J}{\partial U}\left(\frac{\partial R}{\partial U}\right)^{-1}\frac{\partial R}{\partial \alpha} + \frac{\partial J}{\partial \alpha} \tag{4}$$

Finally, the adjoint equation is defined as:

$$\left(\frac{\partial R}{\partial U}\right)^{T}\psi = -\left(\frac{\partial J}{\partial U}\right)^{T}, \tag{5}$$

and the variation of the objective function can be written as

$$\frac{dJ}{d\alpha} = \psi^{T}\frac{\partial R}{\partial \alpha} + \frac{\partial J}{\partial \alpha} \tag{6}$$

To find the objective sensitivities to surface based deformations, we consider an arbitrary (but small) perturbation of the boundary S which, without loss of generality, can be parameterized by an infinitesimal deformation of size δS along the normal direction to the surface S. Assuming a regular flow solution U and a smooth boundary S, the variation of the functional J due to the deformation can be evaluated with a solution to the adjoint equations . These are the surface sensitivties to the objective $\frac{\partial J}{\partial S}$.

Given a parameterization of the surface deformation by a set of design variables x, we estimate the sensitivity of the surface to design variables with finite differencing. These gradients are insensitive for a large range of step sizes. We finally calculate the sensitivity of the objective to the design variables by the chain rule[4]

$$\frac{\partial J}{\partial x} = \frac{\partial J}{\partial S}\frac{\partial S}{\partial x}. \tag{7}$$

3. Gradient-Based Error Estimation and Grid Adaptation

We are using a gradient-based mesh adaptation scheme built into SU^2. Given a component of the flow U_i, we can estimate the build a sensor, Z for indicating adaptation in the cell Ω with

$$Z(U) = |\Omega|^{\alpha}\sqrt{(\partial_x U_i)^2 + (\partial_y U_i)^2}, \tag{8}$$

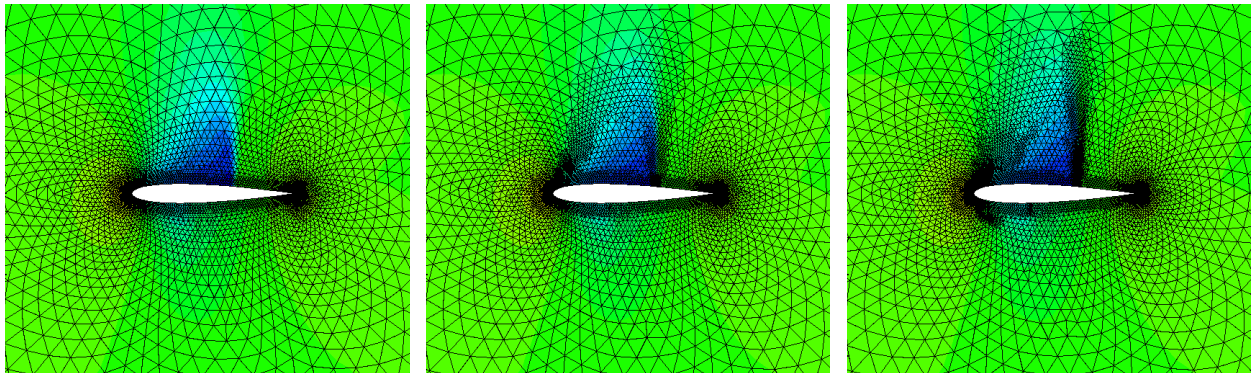
where α is a constant used to de-emphasize small cell volumes $|\Omega|$. In the case of refining a direct solution, we take density to be the flow component used in the adaptation sensor. In the case of refining an adjoint solution, we use the density-adjoint component. Cells are split at locations of high sensor values. Cells along the surface of the airfoil are reconstructed with a linear interpolation between nodes.

Results for two adaptation steps are shown in Figure 1. As expected, refinement is occurring around the shock and other regions of high flow gradients.

However, Figure 2 shows that the adjoint solution has very different behaviors. Several locations, such as around the mid section of the wing, could benefit from mesh refinement for the adjoint. There are several ways to approach the refinement of the adjoint solution, due to its dependence on the direct solution and the need for multiple adjoint solutions in constrained optimization problems.

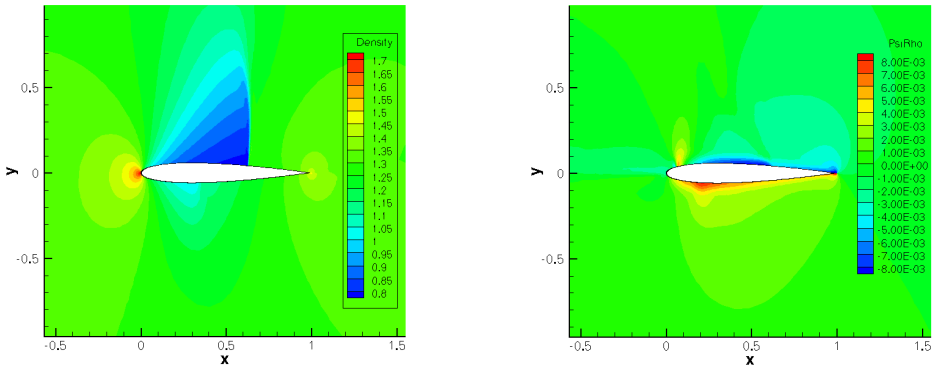
As a first example, one may build a refinement indicator in a multi-objective sense to refine the grid by direct and adjoint solutions simultaneously. This is an intuitive approach, but in order to accommodate multiple adjoint solutions, one must repeat this adaptation loop for each performance parameter, resulting in different refinements of the direct solution.

As a result, we have explored the use of a two-step adaptation of the grid. First, we adapt the direct solution as described before. Then from this mesh, we adapt it separately for each relevant adjoint solution, and only updating the adjoint solution at each adaptation cycle. This results in one output grid for the direct solution, and one output grid for each adjoint solution. While computationally expensive, it is still easier to evaluate than running an isotropically refined solution with finest elements of similar size.



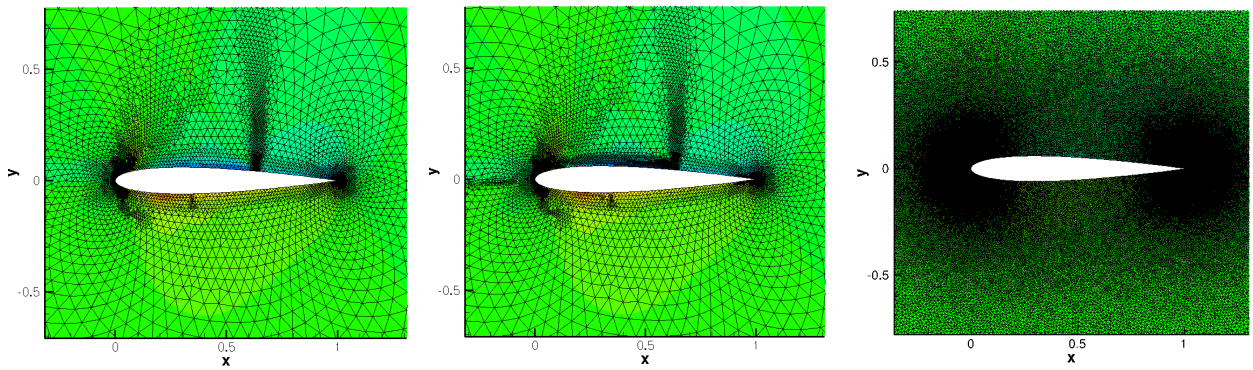
(a) Initial Mesh (b) First Adaptation Step (c) Second Adaptation Step

Figure 1: Example Gradient-Based Mesh Adaptation



(a) Direct Solution (b) Adjoint Solution

Figure 2: Comparison of Direct and Adjoint Flow Solutions



(a) Initial Mesh from Direct Solution Adaptation (b) Adapted Adjoint Mesh (c) Isotropically Refined Mesh

Figure 3: Adjoint Solutions and Grids of Var

In Figure 3, an adapted adjoint solution for drag added cells around the mid-chord and trailing edge of the airfoil, resulting in a final grid size of 22-thousand cells. This is shown against a comparable isotropically refined grid, which needed 200-thousand cells to match the finest cell size of the adapted grid.

One may be able to build a refinement indicator as a function of the direct solution and all relevant adjoint solutions. While this may address the refinement needs of all solutions, it may also sacrifice the refinement of one solution for another given a limited number of cells to add. The merit of such an approach was not explored in this paper and is an area of future work.

B. Filtering Techniques

Our derivation of the adjoint equations for shape design rely on an assumption of smoothness of the adjoint variables. Discontinuous flow features like shocks violate this assumption and can result in high-frequency oscillations or noise in the surface sensitivities.

For example, Figure 4 shows the surface sensitivity for a NACA0012 airfoil adapted for the direct solution. Several large amplitude impulses are present near the sonic points and shock locations. The asymptotic behavior approaching the trailing edge can be explained by the significant dependence of the trailing edge on the behavior of the flow. However, the large amplitude noise that appears at the edge is a numerical artifact created by the sharp training edge.

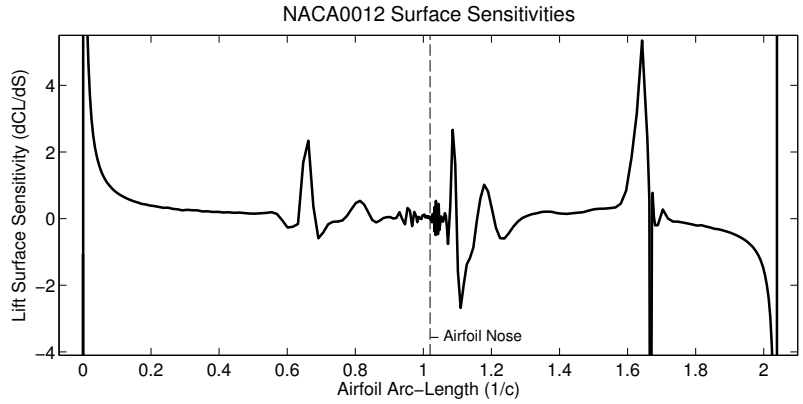


Figure 4: Example NACA0012 Surface Sensitivities for Lift

Smoothing the adjoint surface sensitivities by applying different filters is a common approach for shape design studies which directly deform the shape based on the surface sensitivities. In our study, we project the the surface sensitivity in to a design space of Hicks-Henne bump functions, which can be seen as a filter in itself. Adding a smoothing filter before this is meant to further condition the surface sensitivities towards improving the accuracy of the gradients in the design space.

There are several methods available for filtering the surface sensitivities. A simple first approach is to apply window-based smoothing techniques such as moving average or hamming window filters. Another appropriate approach is a low-pass fourier filter, where the cut-off frequency is chosen based on the edge lengths on the airfoil. A third filter is the Sobolev gradient, which has been applied to the problem of smoothing surface sensitivities by Jameson.[30]

In the course of this study, we examined the windowed smoothing, Fourier filtering, and Sobolev gradient approaches, but found the Sobolev gradient most effective. We will present this method and it's effects on the design space gradient accuracies throughout the rest of this paper.

1. Sobolev Gradient

The Sobolev gradient smooths a signal according to its Laplacian by taking the input $x(t)$ and produces smoothed output $y(t)$ by solving the PDE,

$$y(t) - \varepsilon \frac{\partial^2 y}{\partial t^2} = x(t), \tag{9}$$

where ε is a constant that controls the degree of smoothing. For this problem, the input signal $x(t)$ is the surface sensitivities as a function of airfoil arclength, t, starting from lower trailing edge, and traveling around clockwise to the upper trailing edge.

To solve this on the discretized surface mesh, we setup a system of linear equations of the form

$$y_i - \varepsilon \frac{1}{t_{i+1} - t_{i-1}} \left(\frac{y_{i+1} - y_i}{t_{i+1} - t_i} - \frac{y_i - y_{i-1}}{t_i - t_{i-1}} \right) = x_i \tag{10}$$

which can arranged into the problem $y = A^{-1}x$ and solved with a standard linear algebra package. We apply Dirichlet boundary conditions at the the beginning and end of the airfoil arc length. Before smoothing, we also remove the numerical artifacts at the trailing edge by truncating values within 1% chord from the trailing edge.

The effect of the smoothing on the surface sensitivities is shown in Figure 5 for various values of the constant ε. As the value of ε increase, the amount of smoothing increases and tends to round out the peaks of the signal. We can also see the smoother removing high-frequency noise that appears near adapted regions.

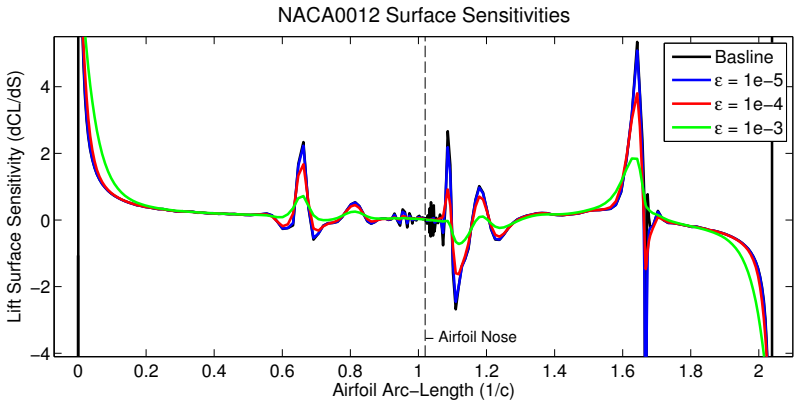


Figure 5: Example Smoothed Surface Sensitivities

C. Response Surface Modeling with Gaussian Process Regression

We will now describe the Gaussian Process Regression approach we use for surrogate based optimization. GPR is as super-set of Kriging. It approaches regression from a Bayesian standpoint by conditioning a probabilistic function to training data.[31] For example, it can be shown that in the case where the probabilistic prior is assumed to be a Gaussian process with a stationary zero mean, the resulting model matches that of Simple Kriging (SK). Because GPR is posed as a conditioning problem and not an expected error minimization problem, it may have more flexibility when handling poorly behaved design spaces, such as discontinuities as found by Chung,[7] or in our case noisy response functions. This is one of the key motivations for our exploration of Gaussian Process Regression.

1. GPR Mathematical Description

Following the derivation given by Rasmussen,[31] Gaussian Process Regression is approached by conditioning a multivariate normal distribution.

$$f \sim \mathcal{N}\left(\mu, [\sigma]\right), \tag{11}$$

where f is a normally distributed function with mean vector μ and standard deviation matrix $[\sigma]$.

For this paper, we take a uniformly zero mean vector, and populate the standard deviation matrix with a covariance submatrix k that is a function of training and estimated data:

$$\begin{gathered} \begin{bmatrix} f_p \\ f_k^* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} k(x_p, x_q) & k(x_p, x_j^*) \\ k(x_k^*, x_q) & k(x_k^*, x_j^*) \end{bmatrix}\right), \\ \{\, f_i(x_i) \,|\, i = 1, ..., n \,\} \;,\; \{\, f_t^*(x_t^*) \,|\, t = 1, ..., m \,\}. \end{gathered} \tag{12}$$

The notation $(\cdot)^*$ is used to distinguish the estimated data from the training data. Additionally, index notation is used to describe the sub-blocks of the covariance matrix, where $k(x_p, x_q)$ would be equivalent to the matrix k_{pq}. There are n training point vectors x with function values $f(x)$, and m estimated data point vectors x^* with function values $f^*(x^*)$.

Of the data, we do not know the estimated function values f^*. We do know the training data locations x and function values $f(x)$, as well as the desired estimated data locations x^*. Following Rasmussen's derivation,[31] we condition the normal distribution with the data we do know

$$f|x^*, x, f \sim \mathcal{N}\left(f^*, \mathbb{V}[f^*]\right), \tag{13}$$

which allows us to identify useful relations for estimating a function fit,

$$\begin{aligned} f_k^* &= k(x_k^*, x_q)\, k(x_p, x_q)^{-1}\, f_p \\ \mathbb{V}[f_k^*] &= \left(k(x_k^*, x_j^*) \;-\; k(x_k^*, x_q)\, k(x_p, x_q)^{-1}\, k(x_p, x_j^*)\right)_k \end{aligned} \tag{14}$$

where $\mathbb{V}[f^*]$ is the covariance of the estimated value f^*. These are the relations needed for coding a GPR program. Rasmussen provides an example algorithm that simplifies these relations by using Cholesky decomposition.[31]

2. Covariance Function

The covariance function models the spatial correlation between data points. It is chosen based on the types of functions that are going to be modeled. Highly-smooth or weakly-smooth functions would be examples of different types of functions that would require different choices of covariance functions. A common covariance function is the Gaussian function of the Euclidean distance between points:

$$\begin{gathered} k(x_p, x_q) = k(p, q) = \theta_1^2 \exp\left(-\frac{1}{2\theta_2^2}\sum_{z=1}^{d}(p_z - q_z)^2\right) \\ \{p_i, q_i, \tfrac{\partial}{\partial x_i} \mid i = 1, ..., d\}, \end{gathered} \tag{15}$$

where d is the number of dimensions, and p and q are the position vectors chosen from the design space x. There are two degrees of freedom in the covariance function. These are known as hyper-parameters. In terms of their effect on the function fit, the nominal variance θ_1 is a measure of the amount of variance allowable between points, and the length scale θ_2 is a measure of the range of influence of a point.

3. Adding Gradient Information

Modeling the influence of gradients on the fit involves adding information to the covariance matrix. This requires finding a covariance function to model the correlation between points and derivatives. One approach to do this is shown for Co-Kriging by Chung[32] by deriving the covariance functions from the definitions of variance and derivative. Another approach suggested by Papoulis[33] and used for gradient enhanced GPR by Solak[34] exploits the theorem that the linear operation of an expected value is the expected value of the linear operation. The result from either approach is the same. To include gradient information in the fit, simply take the derivatives of the covariance function:

$$\begin{aligned} k\left(\tfrac{\partial p}{\partial x_v}, q\right) &= \left.\tfrac{\partial k(p,q)}{\partial x_v}\right|_q \\ k\left(p, \tfrac{\partial q}{\partial x_w}\right) &= \left.\tfrac{\partial k(p,q)}{\partial x_w}\right|_p \\ k\left(\tfrac{\partial p}{\partial x_v}, \tfrac{\partial q}{\partial x_w}\right) &= \left.\tfrac{\partial}{\partial x_w}\left(\left.\tfrac{\partial k(p,q)}{\partial x_v}\right|_q\right)\right|_p . \end{aligned} \tag{16}$$

This is where the assumption of a correlation model between function value and its gradient is made. It is a natural and powerful assumption. However, because there will be d-times more gradient information than function values, inaccurate gradients can override the behavior of the fit.

The gradient information must be packed into the covariance matrix. This can be done by updating the definition for the covariance function as follows:

$$k(p,q) \rightarrow \begin{bmatrix} k(p,q) & k\left(p, \frac{\partial q}{\partial x_w}\right) \\ k\left(\frac{\partial p}{\partial x_v}, q\right) & k\left(\frac{\partial p}{\partial x_v}, \frac{\partial q}{\partial x_w}\right) \end{bmatrix} \quad f_p \rightarrow \begin{bmatrix} f_p \\ \frac{\partial f_p}{\partial x_d} \end{bmatrix}, \tag{17}$$

where submatrix $k(p,q)$ has dimension $n \times n$, and submatrix $k\left(\frac{\partial p}{\partial x_v}, \frac{\partial q}{\partial x_w}\right)$ has dimension $n\,d \times n\,d$.

A useful extension of this formulation is estimating the gradients of the response surface given only objective information. This simply involves omitting the blocks associated with the training data gradients but keeping those associated with the estimated data gradients. Given a reasonable amount of objective data, this can be used to build an analytic estimate of the gradients in the design space for those data. While the curse of dimensionality unfortunately constrains this method to low dimensional design spaces, it is still useful for generating an accurate reference when evaluating the errors of the various sensitivity analysis methods.

4. Noise Models

We are able to model several types of noise within the GPR framework. Allowing noise can relax the assumption of exact correlation between objective and gradient information. The effect on the response surface will have the form:

$$f_N^*(x) = f^*(x) + \epsilon, \tag{18}$$

where ϵ is a noise model. Adding noise to our model requires us to update our covariance matrix structure:

$$[\sigma] = [k] + [k_N], \tag{19}$$

where $[k]$ is the full covariance matrix for functions and gradients, and $[k_N]$ is the noise component of the covariance matrix.

A simple but useful model is an independent identically-distributed Gaussian noise with zero mean and given variance. This will only affect the self-correlated covariance terms along the diagonal of $[k_N]$. The noise covariance matrix will then take the form:

$$[k_N] = \begin{bmatrix} \theta_3^2 I_{n',n'} & 0_{n',m'} \\ 0_{m',n'} & \theta_4^2 I_{m',m'} \end{bmatrix}, \tag{20}$$

where I and 0 are identity and zero matrices with $n' = n(1+d)$ and $m' = m(1+d)$. Note we have allowed two separate noise hyper-parameters for the function values and gradients. Adding this diagonal component to the covariance matrix relaxes the requirement that the fit exactly honors the training data. Depending on the magnitude of the noise hyper-parameter, the fit will be allowed to stray a certain distance away from the data. This will allow us to model noise in the gradients due to inaccuracies from mesh refinement or the particular sensitviy method.

An additional approach to modeling noise not explored in this paper could be found in variable fidelity methods. Much work has been done building Kriging formulations that allow the scaling of function samples for different data sources.[8,13] This allows the response surface model to capture function trends while being flexible with respect to function values. Such a model could be applied to gradients possessing bias errors, and investigating them is an area of future work.

5. Data Scaling

Steps are taken to improve numerics and generality of the method by scaling the data based on the initial LHC sample. The sampled objective function range, and design variable bounds are linearly scaled according to:

$$\begin{array}{llllll} f = f' \cdot f_{ref} + f_{off} & s.t. & [min(f), max(f)] & \rightarrow & [0,1] & \\ x_z = x'_z \cdot x_{z,ref} + x_{z,off} & s.t. & [min(x_z), max(x_z)] & \rightarrow & [0,1] & , z = 1, ..., d \\ \frac{\partial f}{\partial x_z} = \frac{\partial f'}{\partial x'_z} \cdot \frac{f_{ref}}{x_{z,ref}}. & & & & & \end{array} \tag{21}$$

Several benefits are realized from scaling the data past improving the condition of the covariance matrix, if we can assume the response surface is smooth with a nominal amount of variation. First, we can approximately say the variation of data in all design variables is brought to be the same order of magnitude. This allows us to assume isotropy of variation in each dimension and reduce the number of length scale parameters to one. This significantly reduces the computational cost of hyperparameter learning. Second, this allows us to claim that the scaled magnitude of the noise parameters for f and $\frac{\partial f}{\partial x}$ are of the same order of magnitude. This is important since the value of the noise in the gradients is difficult to estimate *a-priori*. Finally, it makes the problem robust to more design problems, where different design parameters of different scales can be mixed without having to learn separate length scales.

6. Hyper-parameter Selection

To use the covariance function, the hyper-parameters θ_{1-4} must be chosen. Different values will yield different fits, each being a different view of the data. We present the method of tuning the required hyper-parameters by maximizing Marginal Likelihood.[31]

Marginal Likelihood measures how well a given set of hyperparameters describes the training data. Find its argument maximum is a common way to select hyperparameters for GPR. It can be defined mathematically with:

$$\log p(f_p|x_p,\theta) = -\frac{1}{2} f_p^\top [\sigma]^{-1} f_p - \frac{1}{2}\log|[\sigma]| - \frac{n}{2}\log 2\pi, \tag{22}$$

where θ is a vector of hyper-parameters. Maximizing the marginal likelihood is itself an optimization problem. This problem can be solved with a gradient based optimizer, however the space is not gauranteed to be convex. This study used modified Newton's method with multiple starting locations chosen with Latin Hypercube sampling to account for multiple local minima.

In order to avoid honoring the gradient information over the objective function information, we constrain the noise hyperparameters according to

$$\theta_3 < \theta_4. \tag{23}$$

If we had not scaled the gradient data to the same order of magnitude as the objective data, this constraint would not be appropriate. The constraint is especially important in a problem with inaccurate gradient information, as the presence of high-dimensional gradients can overpower the objective data and result in a response surface with high error at the training data locations.

III. Numerical Experiments

In the CFD examples shown in this section, we show the objective and gradients of lift, and drag of a NACA0012 airfoil for varying magnitudes of a Hicks-Henne bump functions. Free-stream flow conditions were set at Mach 0.8 and an angle of attack of 1.25°. All data was simulated with a second order Euler scheme and converged by ten orders of magnitude and output with sixteen digits of precision. While experience tells us that this problem is only physically representative to one count ($1e^{-4}$), we need to minimize the dependence of finite-difference gradients and post-processing error calculations on the rounding error.

A. NACA 0012 Single Design Variable Sweeps

We compared the trends and values of the gradients for continuous adjoint and finite difference based sensitivity analyses, at 41 different magnitudes of a single Hicks-Henne bump function, on the lower airfoil surface at the mid-chord. Our reference for error estimation was the response surface based approach to estimating gradients in a low-dimensional design space. If we accept the direct solutions as physically representative, then an accurate response surface based on only the training data should produce a physically representative gradient for that data because we are able to arrive at them analytically. This makes it a useful basis for error calculation.

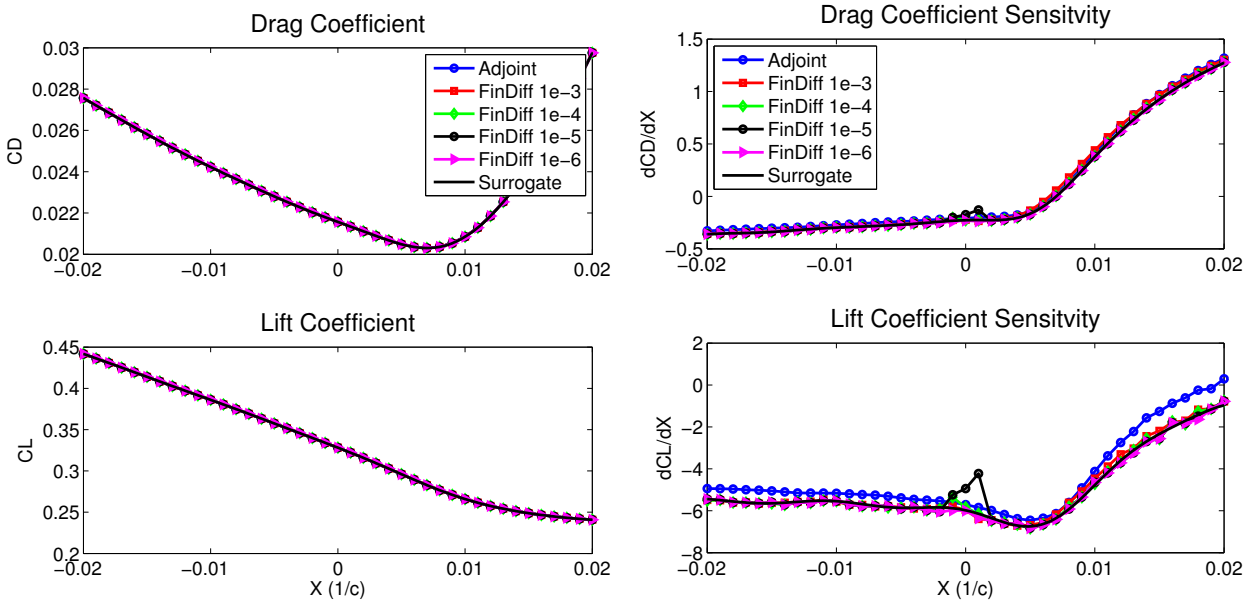
In Figure 6 we are comparing various sensitivity analysis methods for the baseline 20-thousand cell mesh. These methods include the continuous adjoint approach, and finite differencing with various step sizes.

The mean error of the various gradient methods relative to the surrogate model reference is shown in Table 1. These errors are calculated by normalizing the absolute difference of the test gradient and reference

gradient by the min-to-max spread of the reference gradient. Because these measure the accuracy of the gradients in a cross section of the design space, they can be seen as a measure of the design-wide accuracy of the method.

It can be seen that the continuous adjoint gradients are smooth but posses a bias offset on the order of 2% for drag and 10% for lift, which indicates a de-correlation error. This would be a major difficulty if the data were used for response surface methods that assume no correlation error.

In contrast, low bias error but short-scale noise can be seen occurring in the finite difference gradients. There are especially large magnitude errors present in step lengths $1e^{-4}$ and $1e^{-5}$. A closer look at the the noise is shown in Figure 7. The source of the noise is presumably from various shock lines moving between cells during perturbation. The accuracy of the shock is limited by the numerical resolution of the grid, and the need for three nodes to define the location. With small perturbations, this limitation begins to show itself as noise in the design space.



(a) (b)

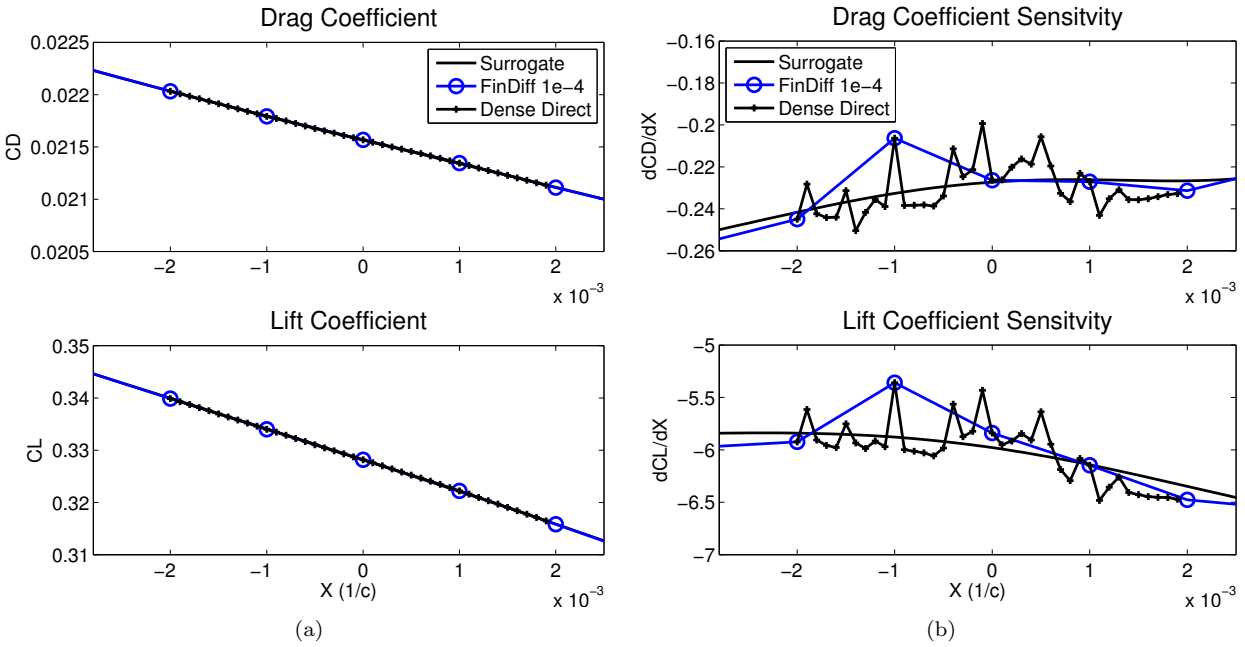
Figure 6: Baseline Mesh Objective and Gradient Sweeps

Gradient Mean Relative Errors					
Variable	Adjoint	FinDiff 1e-3	FinDiff 1e-4	FinDiff 1e-5	FinDiff 1e-6
Lift	9.9%	1.8%	1.4%	2.6%	1.3%
Drag	2.0%	1.3%	0.26%	0.43%	0.20%

Table 1: Baseline Mesh Gradient Errors

The performance of the $1e^{-3}$ step is likely due to the favorable amount of smoothing it performs with its larger step length. This of course may trade accuracy in some design spaces that are not as smooth as the current. We also note that the continuous adjoint estimation of the lift gradient is less accurate than the estimate of the drag gradient. This is because our surface based adjoint approach relies on a central differencing scheme with artificial dissipation constants that were calibrated for the drag gradient. While we could conceivably tune these parameters for the lift gradient, we are trying to avoid internal modifications to the adjoint method in favor of treating it as a black box and applying post-processing fixes.

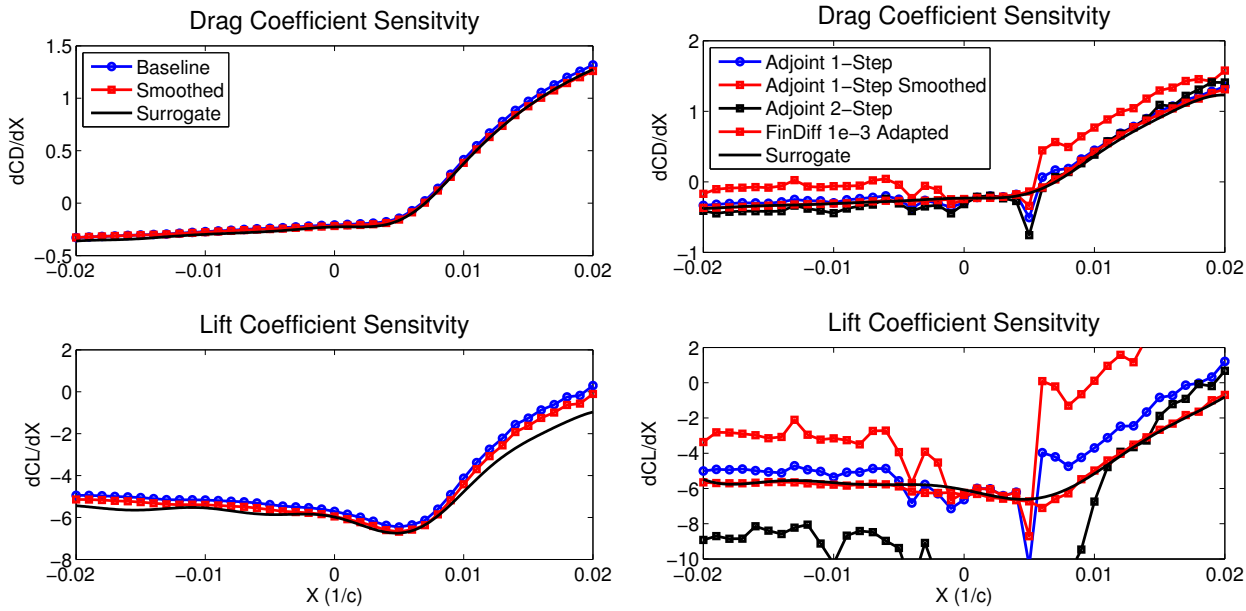
We should be cautious choosing the $1e^{-6}$ despite its apparent accuracy because in a second order scheme, perturbations less than $1/\Delta x^2$ (in a smooth 1D problem) are sensitive to numerical dissipation. Thus for this geometry, design variable and flow conditions, we can observe that the finite difference step of $1e^{-3}$ results in the most physically representative estimation of the gradient.



(a) (b)

Figure 7: Detail of Finite Differencing Noise

Exploring the idea of smoothing sensitivities, we applied the Sobolev gradient to the continuous adjoint surface sensitivities. The result in the design space is shown in Figure 8a. In this case, we have chosen the ε scaling factor to be $1e^{-4}$ by trial and error. We can see that the error of the gradients is being resolved for drag and partially for lift around the center of the design space. This suggests that at least some error in the adjoint based gradient can be explained by numerical noise in the surface sensitivity calculation.



(a) Surface Sensitivity Smoothing (b) Mesh Adaptation

Figure 8: Gradient Sweeps - Surface Sensitivity Improvement Studies

Finally, we applied various adaptation schemes to the gradient analyses in Figure 8b. At every design point a new adaptation study is performed from the deformed baseline mesh. The first approach to adapting the adjoint solution shown is one that adapts only the direct solution (1-Step). The effect on the design space gradients is the introduction of large amplitude noise and differing segments of bias offsets in sections of the design spaces. The second approach, which adapted the direct and then the adjoint solution (2-Step) unfortunately resulted in even poorer gradient behavior. By comparison, the effect of adaptation on finite-differencing was much less severe, although additional errors appear over those of the baseline mesh.

Unfortunately, the Laplacian smoothing approach was unable to correct the adapted gradients in this study and in fact reduced the accuracy of the gradients. In general we have observed inconsistent results when smoothing the sensitivities on adapted meshes, which suggests that a major source of the error may be coming from within the adjoint numerical scheme. Exploring this will be a line of future work.

B. Surrogate Based Optimization of an Analytical Test Problem

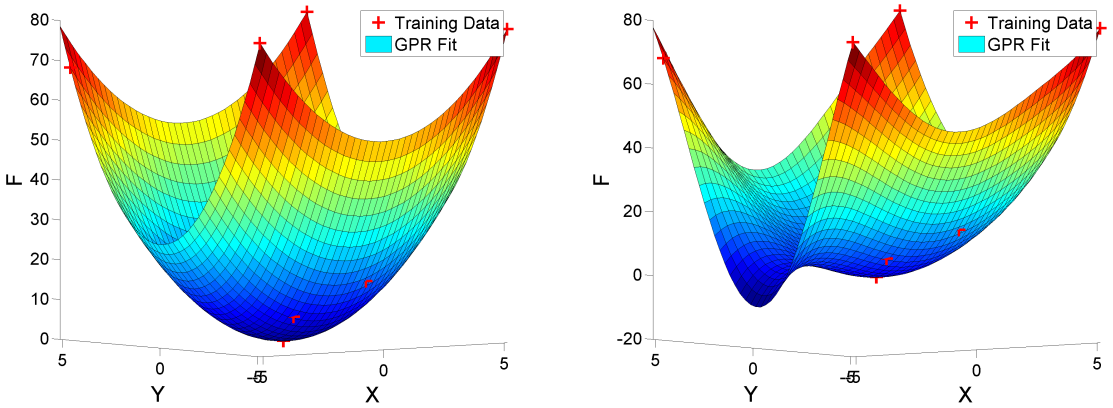
Given the presence of gradient bias error, it may be useful to see its effect in a simple and controlled RSM test case. In this example we examine a response surface fit of a two-dimensional parabola.

The effect of gradient bias error on the response surface was first simulated by scaling the gradient values of a set of sampled training data. The errors were applied according to the equation,

$$\frac{\partial y'}{\partial x} = (1 + \Delta)\frac{\partial y}{\partial x}, \tag{24}$$

where Δ is the amount of relative error being added to the gradients.

Figure 9 shows that a simulated bias error of 4% corrupted the response surface in a way that actually formed a new minimum at the boundary of the design space. The error in the function value and location for this case are shown in Table 2a. We can see that this error grows with increasing gradient bias error.



(a) Simulated Gradient Bias Error: 0%

(b) Simulated Gradient Bias Error: 4%

Figure 9: Response Surfaces for Example Noisy Regression

In reality, a set of perfectly chosen training point locations will not be available. We will only have the initial samples to start with and convergence criteria to know when to stop refining. A second example study was run for the case of design space refinement with gradient bias error. The errors and number of iterations needed are shown in Table 2b. In this simple problem, the RSM was actually able to find the local minimum with desired accuracies, but at the cost of more function evaluations. Note however that the problem with a bias error of 10% failed to converge.

Case 1: Constant Training Data		
Gradient Bias Error	**X_{opt} Error**	**F_{opt} Error**
0.0	1.799E-05	4.336E-05
0.1	8.185E-05	-1.997E-04
0.5	4.865E-04	-1.172E-03
1.0	1.003E-03	-2.389E-03
2.0	2.076E-03	-4.825E-03
3.0	3.207E-03	-7.266E-03
4.0	5.125E+00	-7.063E+00
5.0	5.126E+00	-1.534E+01
10.0	5.129E+00	-5.677E+01

(a)

Case 2: Error During Refinement			
Gradient Bias Error	**X_{opt} Error**	**F_{opt} Error**	**Function Evals**
0.0	1.799E-05	4.336E-05	9
0.1	5.428E-05	-1.061E-04	10
0.5	2.519E-04	-1.525E-04	11
1.0	7.609E-04	-1.915E-04	14
2.0	1.609E-04	-6.894E-05	15
3.0	6.347E-05	-1.535E-04	33
4.0	2.903E-06	-3.274E-04	28
5.0	8.245E-05	-7.390E-04	29
10.0	N/A	N/A	N/A

(b)

Table 2: Convergence and Error Data for Example Noisy Optimization

C. NACA 0012 Response Surfaces

We now apply our gradient enhanced GPR technique to training data from the NACA0012 test case with two Hicks-Henne bump functions on the top and bottom of the airfoil at mid-chord.

In the following examples shown in Figures 10 - 12, we generate a response surface with 20 latin hypercube sampled training data with continuous adjoint based gradients. We then compare the effect of changing the amount of noise the response surface is allowed to model. For reference we compare the fits to a response surface generated from a 10x10 grid of direct solution data only.

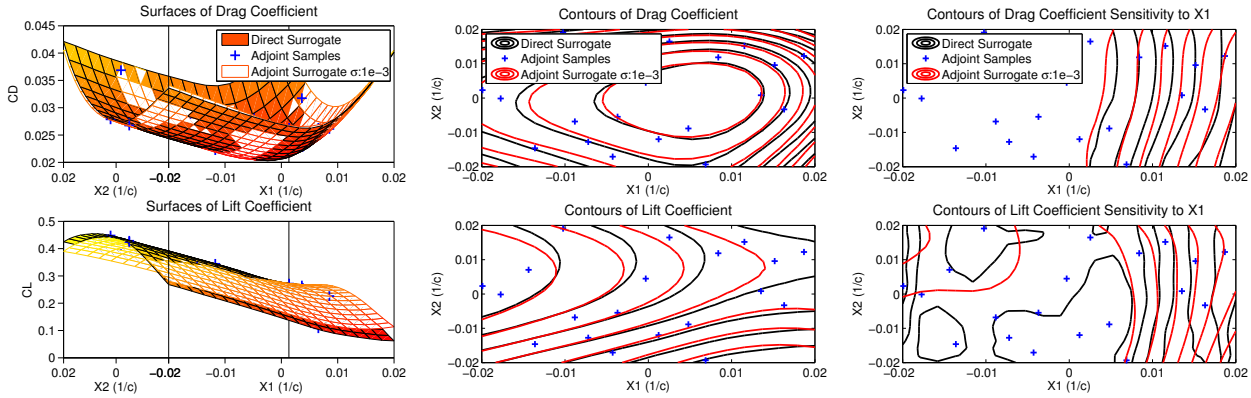
The data shows in Figure 10 that if a noise of $1e^{-3}$ (in dimensions of the design variable) is allowed in the response surface, a reasonable quality fit is achieved with 3.3% mean estimation error in the lift objective, and 1.2% mean estimation error in the drag objective. Finite difference gradients result in a higher quality fit, as shown in Figure 11. If noise is not allowed, or restricted to be very low as in the case of Figure 12, then the accuracy of the response surface degrades significantly. Again as in the analytical case, we see the appearance of additional local minima. This reflects a major observation of this study that response surface with noise-tolerant covariance models are more robust to gradient inaccuracies.

D. NACA 0012 Gradient Based Optimization

We examined the effect of the gradient smoothing techniques on gradient based optimization. In this example we minimize the drag of the NACA0012 case while holding lift fixed and varying ten Hicks-Henne design variables. In Figures 13 - 14, we are plotting the objective and constraint values in the optimization history at every function evaluation, which thus includes sub-iterations. In these figures, a sub iteration can be identified if it does not improve the objective. Sub iterations that show large reductions in performance can be loosely indicative of an inaccurate update of the Hessian estimate. We also are interested in minimizing the number of sub-iterations in order to improve the overall convergence of the optimization.

In the Figure 13a, we see that finite differencing with a step of 1e-6 results in several large amplitude sub-iterations and several iterations which violate the constraint, while the other two methods have smoother convergence. This suggests that the 1e-6 finite differencing (which would have been chosen by automatic finite differencing techniques) is struggling to produce accurate gradients.

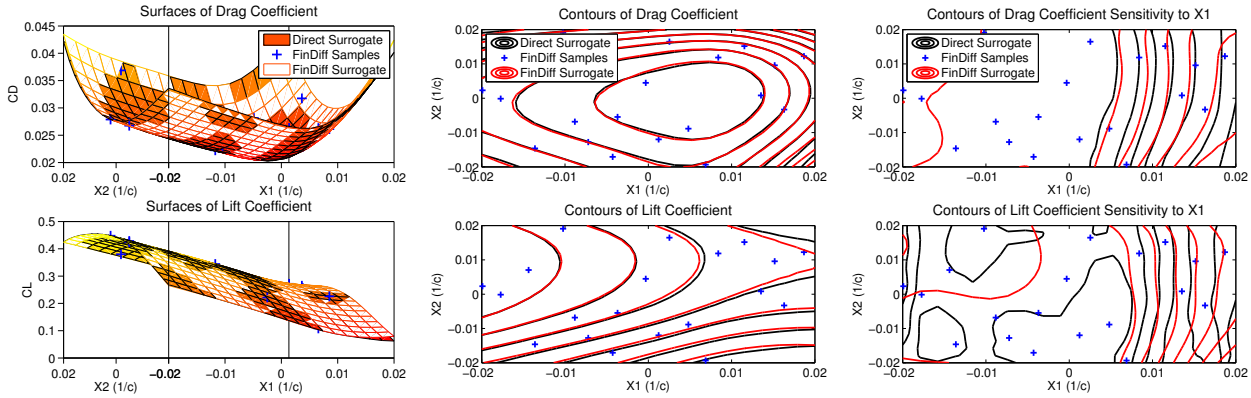
In Figure 13b, we see the effect of adaptation on the convergence. The large amplitude errors of the 1e-6 finite differencing remain, and adjoint gradients develop two poor sub-iteration steps. The 1e-3 finite difference run remains largely unaffected. This reflects a recurring observation we have made during this



(a) (b) (c)

Mean Errors: Lift Objective: 3.3%; Lift Gradient: 10.9%; Drag Objective: 1.2%; Drag Gradient: 3.4%

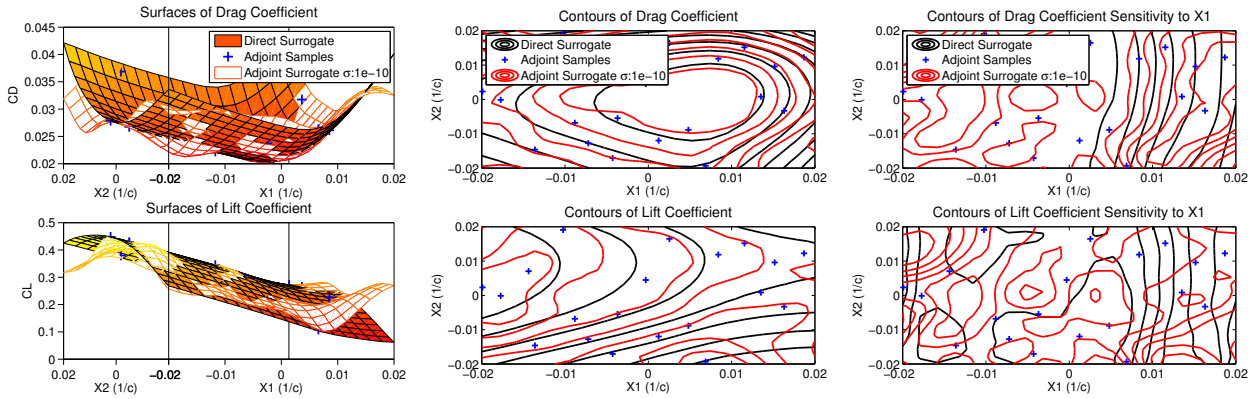
Figure 10: Example RSM with Adjoint Gradients, Noise Tolerence = 1e-3



(a) (b) (c)

Mean Errors: Lift Objective: 0.31%; Lift Gradient: 4.0%; Drag Objective: 0.75%; Drag Gradient: 2.2%

Figure 11: Example RSM with Finite Difference Gradients, Noise Tolerence = 1e-3

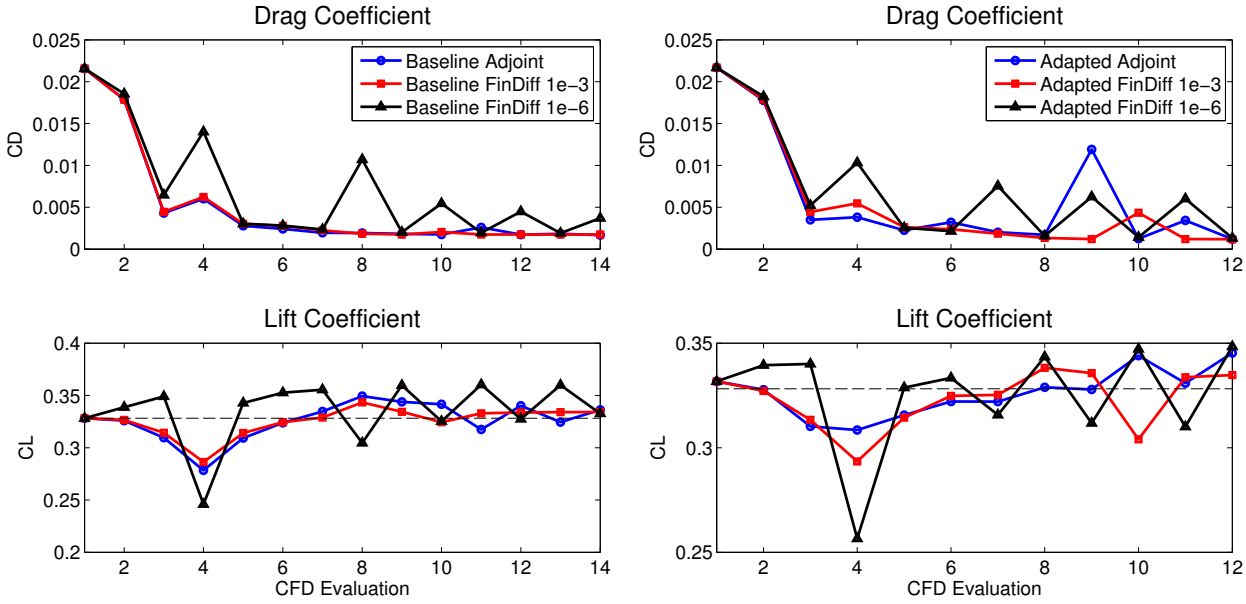


(a) (b) (c)

Mean Errors: Lift Objective: 5.5%; Lift Gradient: 50.8%; Drag Objective: 4.8%; Drag Gradient: 12.8%

Figure 12: Example RSM with Adjoint Gradients, Noise Tolerence = 1e-10

study that physically representative choices of finite differencing steps tend to be more robust to changes in problem discretization.



(a) Baseline Mesh (b) Adapted Mesh

Figure 13: 10-DV Gradient Based Optimizations of NACA0012

Figure 14 shows the effect of the Sobolev filter on the NACA0012 optimization problem. Compared to adapted adjoint run, it has smoothed out the large amplitude sub-iteration errors, presumably by improving the estimation of the Hessian. While this result is unexpected given the result previously in Figure 8b, it could be the case that across several design variables the smoothing technique is helping the optimizer choose a useful search direction for this problem.

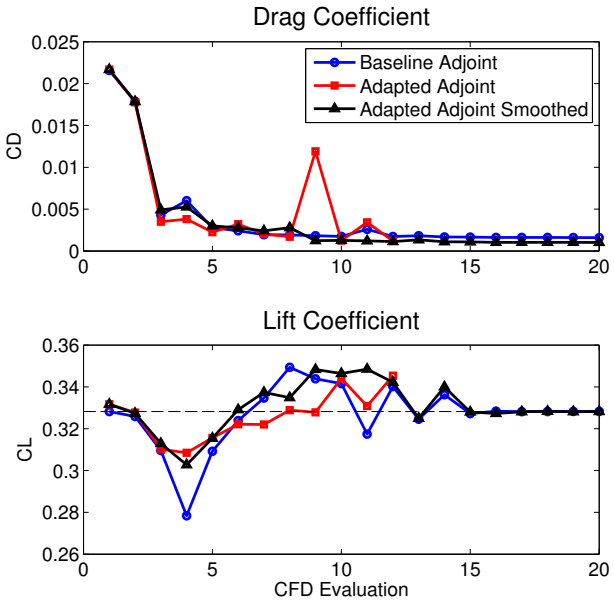


Figure 14: Optimizations of NACA0012 with Adaptation and Smoothing

IV. Conclusions

The main goal of this paper was to develop post-simulation approaches that make optimal shape design methods more robust to inaccurate gradients. A core component of the study relied on the NACA0012 airfoil optimization problem. The relevance of these data stands on the fact that the problem is a standard test case. With it we demonstrated that the basic adjoint-based gradients have bias errors that can corrupt response surface models by as much as 10%. Based on insight from our parabolic test case, this much error could prevent the convergence of a surrogate based optimization. We also demonstrated that minor modifications to the working gradient based optimization case in the form of adaptation can result in adjoint and finite difference gradient errors that add several iterations or also prevent convergence. If the method is struggling with these problems for this simple case, we can expect to meet similar problems for more complicated cases including three-dimensional aircraft.

Several contributions were developed in the course of addressing these problems. First, we used an RSM-based gradient estimation technique which analytically solves for the gradient of a response surface fitted to a dense sampling of performance objectives. This was useful for providing a physically relevant reference gradient for error estimation, and allowed us to identify a finite difference step size that was robust to mesh adaptation. Second, we built a response surface model capable of learning training data noise in order to increase the robustness of SBO methods. Third, we applied the Sobolev gradient to condition continuous adjoint surface sensitivities before projecting them into the parametric design space. And finally, we suggested a two-step mesh adaptation schedule to account for the difference in flow features between the direct and adjoint solutions.

In observing the different magnitudes of error between the continuous adjoint estimates for lift and drag gradients, we can identify an area of future work. Our continuous adjoint solution relies on a dimensional artificial dissipation term, which makes it sensitive to the performance objective of interest. Lift and drag have sensitivities of different orders of magnitude, so the dimensional dissipation we used across both problems could be responsible for their different magnitudes of error. Thus future work could be found developing adjoint solution methods that are independent of the chosen objective function.

Finally, we observed that continuous adjoint gradients errors have similar trends in a design space as their response-surface reference gradients, but had a biased offset indicative of correlation error. While we found that an RSM which models independent noise is robust to these errors, other approaches that model correlated noise or apply variable fidelity methods could perform better. This is another area of future work.

V. Acknowledgments

The authors would like to thank the inputs and useful discussions on adjoint methods from Thomas Taylor of the Aerospace Design Lab, and the financial support of the NASA Supersonics Project of the NASA Fundamental Aeronautics Program.

References

[1]Alonso, J. J. and Colonno, M., "Multidisciplinary Optimization with Applications to Sonic-Boom Minimization," *Annual Review of Fluid Mechanics*, Vol. 44, 2012, pp. 505–526.

[2]Palacios, F., Alonso, J. J., Colonno, M., Hicken, J., and Lukaczyk, T., "Adjoint-based Method for Supersonic Aircraft Design Using Equivalent Area Distributions," *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, AIAA Paper 2012-0269, Nashville, TN, January 2012.

[3]Seebass, R., "Sonic Boom Theory," *Journal of Aircraft*, Vol. 6, 1969, pp. 177–184.

[4]Palacios, F., Colonno, M. R., Aranake, A. C., Campos, A., Copeland, S. R., Economon, T. D., Lonkar, A. K., Lukaczyk, T. W., Taylor, T. W. R., and Alonso, J. J., "Stanford University Unstructured (SU^2): An open-source integrated computational environment for multi-physics simulation and design." *AIAA Paper, 51st Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition (Submitted for Publication), Grapevine, TX*, January 2013.

[5]Myers, R. H., Montgomery, D. C., and Anderson-Cook, C. M., *Response surface methodology: process and product optimization using designed experiments*, John Wiley & Sons, Hoboke, NJ, 2009.

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

[7]Chung, H. and Alonso, J., "Design of a Low-Boom Supersonic Business Jet Using Cokriging Approximation Models," *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA Paper 2002-5598, Atlanta, GA, September 2002.

[8]Yamazaki, W. and Mavriplis, D., "Derivative-Enhanced Variable Fidelity Surrogate Modeling for Aerodynamic Functions," *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 2011.

[9]Parzen, E., "A new approach to the synthesis of optimal smoothing and prediction systems," *Mathematical Optimization Techniques*, edited by R. Bellman, University of California Press, Berkley, CA, 1963, pp. 74–108.

[10]Paciorek, C. and Schervish, M., "Nonstationary Covariance Functions for Gaussian Process Regression," *Advances in Neural Information Processing Systems 16*, edited by S. Thrun, L. K. Saul, and B. Schoelkopf, MIT Press, Cambridge, MA, 2004, pp. 273–280.

[11]Jones, D., "A Taxonomy of Global Optimization Methods Based on Response Surfaces," *Journal of global optimization*, Vol. 21, No. 4, 2001, pp. 345–383.

[12]Forrester, A. and Keane, A., "Recent Advances in Surrogate-Based Optimization," *Progress in Aerospace Sciences*, Vol. 45, No. 1, 2009, pp. 50–79.

[13]Han, Z.-H., Zimmerman, R., and Goertz, S., "A New Cokriging Method for Variable-Fidelity Surrogate Modeling of Aerodynamic Data," *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, 2010.

[14]Choi, S., Alonso, J., Kroo, I., and Wintzer, M., "Multifidelity Design Optimization of Low-Boom Supersonic Jets," *Journal of Aircraft*, Vol. 45, 2008, pp. 106–118.

[15]Wintzer, M. and Sturdza, P., "Conceptual Design of Conventional and Oblique Wing Configurations for Small Supersonic Aircraft," *44th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA Paper 2006-0930, Reno, NV, January 2006.

[16]Ordaz, I. and Rallabhandi, S. K., "Boom Minimization Framework for Supersonic Aircraft Using CFD Analysis," *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, AIAA Paper 2010-1506, Orlando, FL, January 2010.

[17]Lukaczyk, T. W., Palacios, F., and Alonso, J. J., "Adjoint-based method for supersonic aircraft design using equivalent area distributions," *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO*.

[18]Jones, E., Oliphant, T., Peterson, P., et al., "SciPy: Open source scientific tools for Python," 2001–.

[19]Kraft, D., "Algorithm 733: TOMP–Fortran modules for optimal control calculations," *ACM Transactions on Mathematical Software (TOMS)*, Vol. 20, No. 3, 1994, pp. 262–281.

[20]Kraft, D., "A software package for sequential quadratic programming," *Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt fur Luft- und Raumfahrt*, 1988.

[21]Giles, M. B. and Pierce, N. A., "An Introduction to the Adjoint Approach to Design," *Flow, Turbulence and Combustion*, Vol. 65, 2000, pp. 393–415.

[22]*A Comparison of the Continuous and Discrete Adjoint Approach to Automatic Aerodynamic Optimization*, No. AIAA-2000-0667, Reno, NV., January 2000.

[23]Dwight, R. P. and Han, Z., "Efficient Uncertainty Quantification using Gradient Enhanced Kriging," *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2009.

[24]Jameson, A., "Aerodynamic Design via Control Theory," *Journal of Scientific Computing*, Vol. 3, No. 3, 1998, pp. 233–260.

[25]Lions, J. L., *Optimal Control of Systems Governed by Partial Differential Equations*, Springer Verlag, New York, 1971.

[26]Belegundu, A. D. and Arora, J. S., "A Sensitivity Interpretation of Adjoint Variables in Optimal Design," *Computer Methods in Applied Mechanics and Engineering*, Vol. 48, 1985, pp. 81–89.

[27]Taylor, T., Palacios, F., Duraisamy, K., and Alonso, J., "Towards a hybrid adjoint approach for arbitrarily complex partial differential equations," *42nd AIAA Fluids Dynamics Conference. New Orleans, LA*, 2012.

[28]Hascoët, L. and Pascual, V., "TAPENADE 2.1 user's guide," Technical Report 0300, INRIA, http://www.inria.fr/rrrt/rt-0300.html, 2004.

[29]Griewank, A., Juedes, D., and Utke, J., "Algorithm 755: ADOL-C: A Package for the Automatic Differentiation of Algorithms Written in C/C++," *ACM Transactions on Mathematical Software,*, Vol. 22, No. 2, 1996, pp. 131–167.

[30]Jameson, A. and Kim, S., "Reduction of the adjoint gradient formula in the continuous limit," *AIAA paper*, Vol. 40, 2003, pp. 2003.

[31]Rasmussen, C. and Williams, C., *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, MA, 2006, pp. 13–30.

[32]Chung, H., *Multidisciplinary Design Optimization of Supersonic Business Jets Using Approximation Model-Based Genetic Algorithms*, Ph.D. thesis, Stanford University, March 2004.

[33]Papoulis, A., *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, New York, 1991, p. 310.

[34]Solak, E., Murray-Smith, R., Leithead, W. E., Leith, D. J., and Rasmussen, C. E., "Derivative Observations in Gaussian Process Models of Dynamic Systems," *Advances in Neural Information Processing Systems 15*, edited by S. Becker, S. Thrun, and K. Obermayer, MIT Press, Cambridge, MA, 2003, pp. 1033–1040.