

Active Subspaces for Shape Optimization

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

Paul G. Constantine§
Colorado School of Mines, Golden, CO 80401, USA

Aerodynamic shape optimization plays a fundamental role in aircraft design. However, useful parameterizations of shapes for engineering models often result in high-dimensional design spaces which can create challenges for both local and global optimizers. In this paper, we employ an *active subspace method* (ASM) to discover and exploit low-dimensional, monotonic trends in the quantity of interest as a function of the design variables. The trend enables us to efficiently and effectively find an optimal design in appropriate areas of the design space. We apply this approach to the ONERA-M6 transonic wing, parameterized with 50 Free-Form Deformation (FFD) design variables. Given an initial set of 300 designs, the ASM discovered a low-dimensional linear subspace of the input space that explained the majority of the variability in the drag and lift coefficients. This revealed a global trend that we exploited to find an optimal design with reduced computational cost.

Nomenclature

<i>Design problem</i>		\mathbf{U}	Active subspace basis, column matrix
C_D	Drag coefficient	\mathbf{V}	Inactive subspace basis
C_L	Lift coefficient	<i>Surrogate modeling</i>	
B^i	Berstein polynomial of order i	\mathcal{R}	Response surface estimator
P	Bezier solid parametric coefficient	\mathcal{N}	Normal distribution
<i>Design spaces</i>		μ	Mean
\mathbb{R}^n	Set of real numbers of dimension n	$[\sigma]$	Standard deviation matrix
lb, ub	Lower and upper bounds	k	covariance function
\mathbf{x}	Full-space design vector	θ	Hyperparameter
\mathbf{y}	Active subspace design vector	\mathbf{b}	Least squares regression coefficient vector
\mathcal{X}	Space of \mathbf{x}	\mathbf{A}	Least squares regression coefficient matrix
\mathcal{Y}	Space of \mathbf{y}	\approx	Approximation
$f(\cdot)$	Scalar function in full space	\sim	Less accurate Approximation
$g(\cdot)$	Scalar function in active subspace	<i>Subscript</i>	
$\nabla(\cdot)$	Gradient of scalar function	i, j, p, q, t Variable number	
c	Constraint threshold	<i>Superscript</i>	
M, N	Number of samples	m	Design full-space dimension
<i>Active subspace formulation</i>		n	Active subspace dimension
\mathbf{C}	Covariance matrix	$*$	Estimated variable
$\mathbb{E}[\cdot]$	Expected value of a random quantity	N	Model with noise component
\mathbf{W}	Column matrix of eigenvectors		
$\mathbf{\Lambda}$	Diagonal matrix of eigenvalues		

*Graduate Student, Department of Aeronautics and Astronautics, AIAA Student Member.
†Engineering Research Associate, Department of Aeronautics and Astronautics, AIAA Senior Member.
‡Associate Professor, Department of Aeronautics and Astronautics, AIAA Associate Fellow.
§Ben L. Fryrear Assistant Professor, Applied Mathematics and Statistics, AIAA Member

I. Introduction

In the field of optimal shape design, we are interested in iteratively changing an aerodynamic shape in order to improve the performance of an aircraft. In realistic three-dimensional design problems, it is typical for shape optimizations to require hundreds of design variables.^{1,2} The core impetus for these large dimension spaces is that we cannot know *a-priori* which variables are needed to most efficiently optimize the design. We thus construct parameterizations with many variables in order to increase the chance that we reach a useful result. However, this increases the cost required to find an optimum, and increases the possibility that multi-modality appears in complex design problems.

Today's optimization techniques manage this "curse of dimensionality" in different ways. Local optimizers such as gradient-based optimization techniques efficiently find local minima when paired with adjoint-based sensitivity analyses, but are not guaranteed to return a global minimum.³ Global optimizers including genetic algorithms and covariance matrix adaptation can find global minima, but require large numbers of function evaluations, especially in high-dimensional design spaces.⁴ Surrogate-Based Optimization (SBO) approaches seek to strike a balance between global and local optimizers by building an inexpensive response surface approximation. The community has been especially interested in these methods recently because they promise to be an efficient global optimization approach, useful for high-fidelity preliminary design.⁵⁻⁷

Non-parametric regression techniques such as Gaussian Process Regression (also known as Kriging) are useful for SBO, because they make few assumptions about the trends of the objective's response surface. However, training these surrogate models require additional overhead and complexity when working with even modest numbers of design variables, to the point that they struggle to be predictive for complex design problems.

The solution proposed in this paper works around these dimensionality issues by finding a low-dimension subspace that captures the global trends in the objective function using the Active Subspace Method (ASM).⁸ The approach learns the linear subspace that best describes the variability in the objective using an eigenvalue decomposition of the objective's gradients. The input vectors are projected onto this subspace, and the outputs can be mapped in these new coordinates. We refer to these coordinates as the active coordinates.

We are able to apply this to design problems where aircraft shapes are described by high-dimensional geometric parameterizations, but the majority of the variability in the objective functions resides in a low-dimensional subspace of the parameters. The ASM discovers and exploits this subspace for design optimization and surrogate modeling.

We contrast the approach to Principal Component Analysis (PCA), also known as Proper Orthogonal Decomposition (POD).⁹ PCA is typically used to either reduce the dimension of the *output* space, for example the objectives of a multidimensional optimization; or the dimension of an input space that has been conditioned by some process, for example on the optimized samples on a pareto-front.¹⁰ The ASM is different in that it reduces the *input* space with a *non-conditioned evenly-spread* set of training data using only the model outputs and their gradients.

By defining useful inverse-maps, multiple subspaces can be used for different objectives, such as lift and drag coefficients for a constrained optimization problem. The function can be visualized if the reduced space is of dimension one or two. If the quantity of interest varies monotonically along the reduced coordinate, the trend will be apparent in plots. In the presence of such a trend, the optimization becomes much simpler: find the point in this active subspace that minimizes the quantity of interest.

II. ONERA M6 design optimization

In this study we are interested in the optimal shape design (OSD) of the ONERA-M6 fixed wing, a standard transonic test-geometry. This paper is particularly concerned with the problem of maintaining a minimum lift while minimizing the inviscid drag:

$$\begin{aligned}
 & \underset{\mathbf{x}}{\text{minimize}} && C_D(\mathbf{x}) \\
 & \text{subject to} && C_L(\mathbf{x}) \geq 0.2864 \\
 & && lb_i < \mathbf{x}_i < ub_i, \quad i \in \{0, \dots, m\}, \\
 & && \mathbf{x} \in \mathbb{R}^m
 \end{aligned} \tag{1}$$

where m is the number of design variables, and lb_i and ub_i are the lower and upper bounds for those variables respectively.

The flight conditions are a free-stream Mach number of 0.8395, at an angle of attack of 3.06 degrees. In this work we use the Euler solver provided by SU², an open-source simulation suite developed in the Aerospace Design Laboratory at Stanford University.¹¹ In addition to its general purpose partial differential equation solver, SU² is equipped with tools for optimal shape design including flow and adjoint solvers, free-form mesh deformation, goal-oriented adaptive mesh refinement, and a constrained optimization environment. These tools are wrapped in the Python language to efficiently manage the input and output of data and the exchange of information between the different modules in the SU² suite.

The surface contours in Figures 1 and 2 below show typical results for the ONERA-M6 wing from SU². The direct problem (Figure 1) solves for the design objective and constraints, in this case lift and drag. The pressure coefficient contours identify a lambda-shock along the mid-chord of the wing. This a flow feature that strongly contributes to drag, and should be removed by adjusting the wing's shape. The adjoint problem (Figure 2) contributes to this by solving for the flow's shape sensitivity to a particular objective. The surface sensitivity contours identify locations that can be deformed to minimize drag.

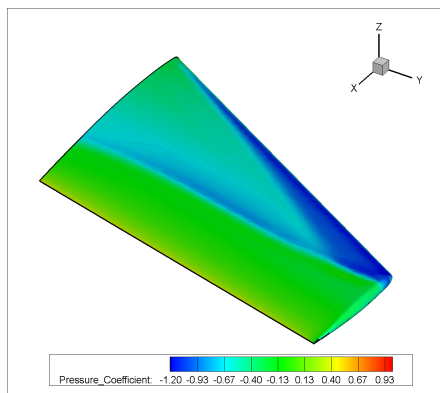


Figure 1: Contours of C_p from a typical direct solution. $C_D = 0.0118$, $C_L = 0.2864$

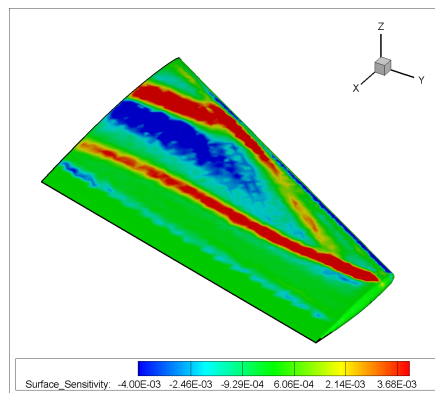


Figure 2: Contours of surface sensitivity from a typical drag adjoint solution.

The actual deformation of the surface is carried out by a separate geometry parametrization. In this paper, a Free-Form Deformation (FFD) strategy¹¹ is used. First, an initial box encapsulating the wing to be redesigned is parameterized as a Bézier solid. Then, a set of control points are defined on the surface of the box, the number of which depends on the order of the chosen Bernstein polynomials. Locations inside the solid box are parameterized by the following expression

$$X(u, v, w) = \sum_{i,j,k=0}^{l,m,n} P_{i,j,k} B_j^l(u) B_j^m(v) B_k^n(w), \quad (2)$$

where $u, v, w \in [0, 1]$, and B^i is the Bernstein polynomial of order i . The Cartesian coordinates of the points on the surface of the object of interest (the wing) are then transformed into parametric coordinates within the Bézier box.

Control points of the box become design variables, as they control the shape of the solid, and thus the shape of the surface grid inside. The box enclosing the geometry is deformed by modifying its control points, with all the points inside the box inheriting a smooth deformation. Arbitrary changes to the thickness, sweep, twist, etc. are possible for the design of an aerospace system. Once the deformation has been applied, the new Cartesian coordinates of the object of interest can be recovered by simply evaluating the mapping inherent in Eq. 2. An example of a deformation of the ONERA-M6 wing is shown in Figure 3 below.

To sample the training data required for the active subspace approach, we use Python scripts that accept a design matrix, perform the CFD analysis, and return a Matlab data structure for post processing. The CFD analysis in this design case requires first domain decomposition, then mesh deformation, and finally the direct flow simulation. The active subspace methods we will show in the next section are implemented in Matlab.

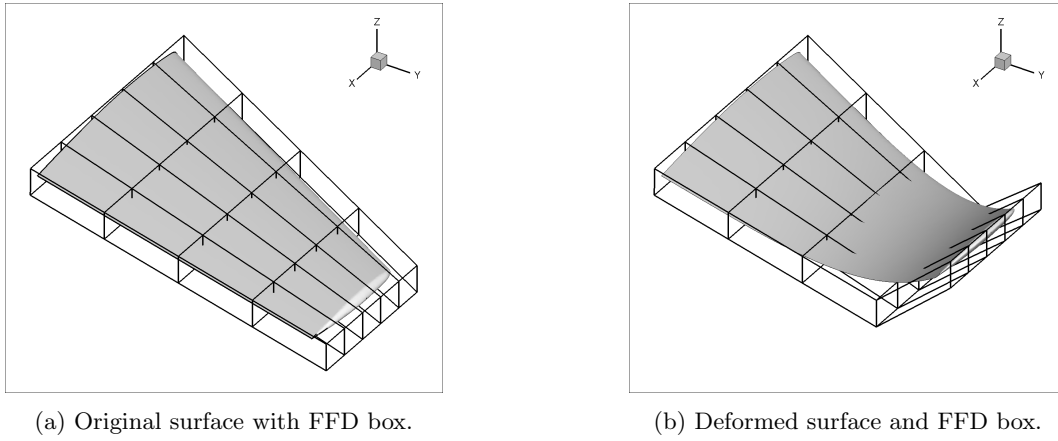


Figure 3: Example surface deformation with FFD boxes.

III. Active subspace method for optimization

A. Construction of the active subspace

In this section we describe the theory behind the active subspace methods we will use to optimize the ONERA-M6 wing design; further details can be found in Constantine, et al.⁸ Consider a scalar function f of a column vector \mathbf{x} , and its gradient $\nabla_{\mathbf{x}}f$ also oriented as a column vector:

$$f = f(\mathbf{x}), \quad \nabla_{\mathbf{x}}f = \nabla_{\mathbf{x}}f(\mathbf{x}), \quad \mathbf{x} \in \mathcal{X} = [-1, 1]^m. \quad (3)$$

where m is the dimension of the design vector \mathbf{x} .

Here we assume that the design space \mathcal{X} is bounded by a hypercube with bounds ± 1 . Thus the design problem should be scaled and translated into this space. The task at hand is to determine the orthogonal directions that most effectively describe the variability of the objective f .

We must first estimate the covariance of the gradient. Let $\rho = \rho(\mathbf{x})$ be a uniform probability density on \mathcal{X} ; we use the expectation operator $\mathbb{E}[\cdot]$ to denote integration against ρ . Thus the $m \times m$ (uncentered) covariance matrix \mathbf{C} is defined as:

$$\mathbf{C} = \mathbb{E}[\nabla_{\mathbf{x}}f \nabla_{\mathbf{x}}f^T]. \quad (4)$$

In practice, we approximate the elements of \mathbf{C} with a Monte Carlo method, by randomly sampling gradient values in the design space. The approximated covariance matrix is

$$\mathbf{C} \approx \frac{1}{M} \sum_{i=1}^M \nabla_{\mathbf{x}}f_i \nabla_{\mathbf{x}}f_i^T, \quad (5)$$

where $\nabla_{\mathbf{x}}f_i = \nabla_{\mathbf{x}}f(\mathbf{x}_i)$, and \mathbf{x}_i is drawn uniformly at random from $[-1, 1]^m$.

The sampling requires M evaluations of f and its gradient. When interrogating aircraft designs with CFD simulations, adjoint methods reduce the computational expense because they allow us to evaluate the entire gradient vector for the equivalent cost of the flow solution. This replaces what would be an expensive finite differencing operation in high dimension.

To identify the important directions of the input space, we find the eigenvectors of the covariance matrix. This matrix is symmetric and positive semidefinite, so it admits a real eigenvalue decomposition,

$$\mathbf{C} = \mathbf{W}\mathbf{\Lambda}\mathbf{W}^T, \quad (6)$$

where \mathbf{W} is a $m \times m$ column matrix of eigenvectors, and $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues.

Note that for many design problems of interest, m is in the tens to hundreds, so the complete eigenvalue decomposition is easily computed on a personal workstation with standard linear algebra toolboxes. The eigenvalues and eigenvectors must be ordered descending. We will select first n eigenvectors to form a reduced-order basis. *This is where we reduce the dimensionality of the design problem.* We will attempt to

describe the behavior of the objective function by projecting the full-space design variables into this *active subspace*.¹² Low eigenvalues suggest that the corresponding vector is in the nullspace of the covariance matrix, and we can discard these vectors to form an approximation. This is an important step, as a judgment call must be applied to the dimension of the active-subspace basis. It can be informed heuristically by inspecting the decay of the eigenvalues.

After choosing n dimensions to keep, the eigenvectors and eigenvalues can be partitioned accordingly:

$$\mathbf{W} = \begin{bmatrix} \mathbf{U} & \mathbf{V} \end{bmatrix}, \quad \mathbf{\Lambda} = \begin{bmatrix} \mathbf{\Lambda}_1 & \\ & \mathbf{\Lambda}_2 \end{bmatrix}, \quad (7)$$

where \mathbf{U} contains the first k columns of \mathbf{W} , and defines the active-subspace of the input space. With the basis identified, we can now map forward to the active subspace,

$$\mathbf{y} = \mathbf{U}^\top \mathbf{x}, \quad (8)$$

and approximate the function f in this active subspace,

$$f(\mathbf{x}) \approx g(\mathbf{U}^\top \mathbf{x}) = g(\mathbf{y}). \quad (9)$$

We can then apply a surrogate model in the active subspace,

$$g(\mathbf{y}) \approx g^*(\mathbf{y}) \equiv \mathcal{R}(\mathbf{y}; g_1, \dots, g_N), \quad (10)$$

where $g^*(\mathbf{y})$ is the surrogate model defined in the active subspace, and \mathcal{R} is the chosen response surface method trained on the points g_1, \dots, g_N . The domain of g is

$$\mathcal{Y} = \left\{ \mathbf{y} = \mathbf{U}^\top \mathbf{x}, \quad \mathbf{x} \in \mathcal{X} \right\} \subset \mathbb{R}^n. \quad (11)$$

B. Surrogate modeling

The motivation for developing active subspace methods is to enable surrogate modeling for design problems with large dimension. Surrogate models suffer from the curse of dimensionality. In our experience and from experiments not presented here, we have struggled to properly reduce the training and testing error of a surrogate model in dimensions greater than ten, even for reasonably simple responses such as Rosenbrock's function. By reducing the input space dimensionality, we can accept a small penalty in the accuracy of the surrogate model in exchange for the opportunity to tackle high dimension problems.

In this work we have used three types of surrogate models - linear regression, quadratic regression, and Gaussian Process Regression (GPR). All three have been extensively applied in this area of research. However, GPR requires a special consideration for noisy data in the present work. We will briefly describe the construction of this model.

1. GPR Mathematical Description

GPR is a super-set of Kriging. It approaches regression from a Bayesian standpoint by conditioning a probabilistic function to training data.¹³ Following the derivation given by Rasmussen,¹³ Gaussian Process Regression is approached by conditioning a multivariate normal distribution,

$$g \sim \mathcal{N}(\mu, [\sigma]), \quad (12)$$

where g 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:

$$\begin{bmatrix} g_p \\ g_i^* \end{bmatrix} \sim \mathcal{N} \left(0, \begin{bmatrix} k(x_p, x_q) & k(x_p, x_j^*) \\ k(x_i^*, x_q) & k(x_i^*, x_j^*) \end{bmatrix} \right), \quad (13)$$

$$\{ g_i(x_i) \mid i = 1, \dots, M \}, \{ g_p^*(x_p^*) \mid p = 1, \dots, N \}.$$

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 M training point vectors x with function values $g(x)$, and N estimated data point vectors x^* with function values $g^*(x^*)$.

Of the data, we do not know the estimated function values g^* . We do know the training data locations x and function values $g(x)$, as well as the desired estimated data locations x^* . Following Rasmussen's derivation,¹³ we condition the normal distribution with the data we do know

$$g|x^*, x, g \sim \mathcal{N}(g^*, \mathbb{V}[g^*]), \quad (14)$$

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

$$\begin{aligned} g_i^* &= k(x_i^*, x_q) k(x_p, x_q)^{-1} g_p \\ \mathbb{V}[g_i^*] &= (k(x_i^*, x_j^*) - k(x_i^*, x_q) k(x_p, x_q)^{-1} k(x_p, x_j^*))_i \end{aligned} \quad (15)$$

where $\mathbb{V}[g^*]$ is the covariance of the estimated value g^* . These are the relations needed for coding a GPR program. Rasmussen provides an example algorithm that simplifies these relations by using Cholesky decomposition.¹³

2. Noise Models

We are able to model several types of noise within the GPR framework.¹⁴ Allowing noise can relax the assumption of exact modeling of objective and gradient information. The effect on the response surface will have the form:

$$g_N^*(x) = g^*(x) + \epsilon, \quad (16)$$

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

$$[\sigma] = [k] + [k_N], \quad (17)$$

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:

$$[k_N] = \theta_3^2 I_{n', n'} \quad (18)$$

where I is the identity matrix. Note we have introduced a 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 interpolates 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 active subspace due to incompletely spanning the design space.

3. Hyper-parameter Selection

To use the covariance function, the hyper-parameters 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.¹³

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:

$$\log p(g_p|x_p, \theta) = -\frac{1}{2} g_p^\top [\sigma]^{-1} g_p - \frac{1}{2} \log |\sigma| - \frac{n}{2} \log 2\pi, \quad (19)$$

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 guaranteed to be convex. This study used Covariance Matrix Adaptation for global optimization, and then Sequential Least-Squares Quadratic Programming for a local optimization.

C. Subspace regularization

The definition of $\mathbf{y} = \mathbf{U}^T \mathbf{x}$ is a surjective map $\mathbf{y}(\mathbf{x})$, since \mathbf{U} is a tall rectangular matrix by construction. This map defines a set of full-space coordinates \mathbf{x} which map to a smaller set of active-subspace coordinates \mathbf{y} . To make use of this for optimization, we need to define the inverse map, and thus define a process for regularization. Without any considerations, the inverse map actually defines an infinite set of points in the full space. We must express assumptions about the full space in order to choose a single point, which we will send back to the flow solver. In the next section we will describe several formulations for regularization that are appropriate for the optimization context.

1. Bounded full-space

A more appropriate approach for implementing the inverse map $\mathbf{x}(\mathbf{y})$ in the context of optimization is to enforce that the selected point \mathbf{x} is located in bounding box of the full space, while being within the span of the active subspace, and projecting down to a selected location in the active subspace \mathbf{y}_{select} .

$$\begin{aligned} &\text{given} && \mathbf{y} = \mathbf{y}_{select} \\ &\text{find} && \mathbf{x} \\ &\text{subject to} && \mathbf{x} \in \mathcal{X} \\ &&& \mathbf{y} = \mathbf{U}^T \mathbf{x} \end{aligned} \tag{20}$$

This is solvable with a linear program, which can be constructed with:

$$\begin{aligned} &\text{given} && \mathbf{y} = \mathbf{y}_{select} \\ &\text{minimize}_{\mathbf{x}} && \mathbf{0}^T \mathbf{x} \text{ (a dummy function)} \\ &\text{subject to} && lb_i < \mathbf{x}_i < ub_i, \ i \in \{0, \dots, m\} \\ &&& \mathbf{y} = \mathbf{U}^T \mathbf{x} \\ &\text{yield} && \mathbf{x} \end{aligned} \tag{21}$$

To use typical linear programming toolboxes, we have to define a linear objective. In the absence of this, we define a constant zero dummy objective. The optimizer will respond to this by only finding a point that is feasible according to the constraints.

This is an example of a “minimum viable method” for mapping an active subspace point into the full space. Its main drawback is the indeterminacy of the resulting full space point \mathbf{x} . Depending on the initial conditions of the linear program and the trajectory the program takes, we may find very different points in the full space for small changes in the active space. To an approximation however, this indeterminacy is acceptable in optimization if the active subspace contains the major trends in the objective. In this case, changes in \mathbf{x} outside of the parameterization of \mathbf{y} have negligible effect on the objective function of interest.

2. Bounded full-space, minimize higher order surrogate

As we will see in Chapter IV of this paper, there are cases in which we want to restrict the active subspace dimension, e.g., for visualization. Unfortunately in this case, the active subspace discards a non-negligible amount of function behavior. This is observable as high variances of the data around the surrogate model, or as high training error. To account for this, we propose a mapping that introduces an assumption based on our overall goal of minimizing an objective function. Thus we construct the optimization:

$$\begin{aligned} &\text{given} && \mathbf{y}_1 = \mathbf{y}_{select} \\ &&& \mathbf{U}_1 \in \mathbb{R}^{m \times n} \\ &&& \mathbf{U}_2 \in \mathbb{R}^{m \times (n+\ell)} \\ &&& f(\mathbf{x}) \approx g^*(\mathbf{U}_2^T \mathbf{x}) \\ &\text{minimize}_{\mathbf{x}} && g^*(\mathbf{U}_2^T \mathbf{x}) \\ &\text{subject to} && lb_i < \mathbf{x}_i < ub_i, \ i \in \{1, \dots, m\} \\ &&& \mathbf{y}_1 = \mathbf{U}_1^T \mathbf{x} \\ &\text{yield} && \mathbf{x} \end{aligned} \tag{22}$$

Notice that bases \mathbf{U}_1 and \mathbf{U}_2 have a different number of columns, since they are representing active subspaces of different dimension. Furthermore, we are defining a location in the subspace basis \mathbf{U}_1 , but minimizing the objective parameterized in basis \mathbf{U}_2 .

The motivating example is the case in which we want to visualize the behavior of the function in two dimensions (i.e., with a contour plot), but the behavior of the function is complicated enough to require more dimensions, perhaps five as our design example we will show. This mapping allows us to plot in two dimensions, and regularize into the full dimension, *assuming we are trying to minimize an objective in the full space*. There is also a secondary assumption that we apply a surrogate model to the objective projected in an active subspace of small dimension $2 + \ell$.

3. Bounded full-space, given a constraint function

This is needed for constrained optimization, where two functionals, and thus two active subspaces, are needed. It is advantageous to allow the two subspaces to span different regions of the full space, because (i) it enables more accurate decomposition of the subspace for a particular objective, and (ii) the two subspaces combined may span a larger region of the full space for no additional cost of dimensionality in each surrogate model.

It is possible to link the two functionals through the full space \mathcal{X} by constructing the following optimization:

$$\begin{aligned}
 &\text{given} \quad \mathbf{y}_a = \mathbf{y}_{select} \\
 &\quad f_a(\mathbf{x}) \approx g_a^*(\mathbf{y}_a) \\
 &\quad f_b(\mathbf{x}) \approx g_b^*(\mathbf{y}_a) \\
 &\quad \mathbf{y}_a = \mathbf{U}_a^T \mathbf{x}, \mathbf{U}_a \in \mathcal{R}_{m \times n_a} \\
 &\quad \mathbf{y}_b = \mathbf{U}_b^T \mathbf{x}, \mathbf{U}_b \in \mathcal{R}_{m \times n_b} \\
 &\text{minimize}_{\mathbf{x}} \quad 0, \text{ (a dummy function)} \\
 &\text{subject to} \quad lb_i < \mathbf{x}_i < ub_i, i \in \{0, \dots, m\} \\
 &\quad \mathbf{y}_a = \mathbf{U}_a^T \mathbf{x} \\
 &\quad g_b^*(\mathbf{U}_b \mathbf{x}) \leq c \\
 &\text{yield} \quad \mathbf{x}
 \end{aligned} \tag{23}$$

where we again define a dummy function to take advantage of existing optimization toolboxes. In this case we are not able to apply a linear program because of the potentially non-linear constraint g_b^* .

In this formulation, bases \mathbf{U}_a and \mathbf{U}_b represent the subspaces for two separate functionals, $f_a(\mathbf{x})$, and $f_b(\mathbf{x})$, which we are approximating with surrogate models in the active subspaces $g_a^*(\mathbf{y}_a)$ and $g_b^*(\mathbf{y}_b)$. We can see that the common link between f_a and f_b is their embedding in full space \mathcal{X} . However we interrogate the surrogate models in their respective subspaces. The optimization links these two spaces while enforcing that the full space point \mathbf{x} projects into the selected active subspace point \mathbf{y}_a .

D. Summary of methods

The active subspace method allows us to construct surrogate models in a subspace with reduced dimension. Since f varies primarily along the coordinates \mathbf{y} , we expect that optimization over \mathbf{y} using the response surface will yield a good approximation of the optimum of f over its domain \mathcal{X} . We can use the regularizations described above to interrogate the design space.

In summary, to approximate f using the active subspace, we employ the following procedure.

1. Choose M points $\mathbf{x}_i \in [-1, 1]^m$, and compute $f_i = f(\mathbf{x}_i)$ and $\nabla_{\mathbf{x}} f_i = \nabla_{\mathbf{x}} f(\mathbf{x}_i)$.
2. Compute the eigendecomposition as in (6) to determine \mathbf{U} .
3. Choose a set of design points $\mathbf{y}_j \in \mathcal{Y}$ for $j = 1, \dots, N$.
4. For each \mathbf{y}_j , evaluate the inverse map to get $\mathbf{x}_j = \mathbf{x}(\mathbf{y}_j)$. This may require computing a temporary response surface trained with the current samples.
5. Compute the expensive functions $f(\mathbf{x}_j)$. Map into the active subspace with $g_j = g(\mathbf{y}_j) = f(\mathbf{x}_j)$.

6. Construct the final response surface g^* with this training data.
7. Use the response surface to interrogate the design space, i.e. for minimizing the estimated function.

We will now demonstrate this method with an example design optimization of the ONERA-M6.

IV. Application to ONERA design optimization

In this section we apply the general active subspace method to find the ONERA-M6 wing design with the minimum drag whose lift exceeds a particular threshold. The space of the design variables is a scaled hypercube $\mathcal{X} = [-0.05, 0.05]^{50}$, i.e., 50 design variables characterizing the wing shape. The size of this space was determined by trial, shrinking the bounds until a majority of the flow evaluations would successfully converge.

A. Constructing the active subspaces

We begin with a 300-point Latin hypercube design sample in the full space \mathcal{X} . We sampled the lift coefficient, drag coefficient, and their gradients with respect the design variables. The gradients are computed with an adjoint-based strategy. We will use these samples to study the active subspaces of each scalar quantity of interest.

We first determine the dominant active subspace of the lift. The eigenvalues from Equation (6) for lift are shown in Figure 4. The separation between the first and second eigenvalue shows that there is a dominant one-dimensional active subspace.

We can compare the decay of the eigenvalues with the decay of surrogate model training error shown in Figure 4. In this example we project the data into active subspaces of varying dimension, construct a GPR surrogate model, and calculate the root-mean-square error of the training data against the surrogate. This error is scaled such that the range of the objective samples is $[0, 1]$, making it a relative error. Because we have a large amount of training data, we can expect the surrogate model constructed in a low dimension to be accurate if the the data collapses into a manifold. Thus the training error is an indication of how well the active subspace has collapsed the data.

In the example presented in Figure 5, the training error for lift starts at 3% with one active subspace dimension. Considering that this one dimension captures the behavior of lift across 50 design variables, this can be seen as a reasonably accurate surrogate model. Additionally, there is marginal gains in the surrogate's accuracy with increasing subspace dimension.

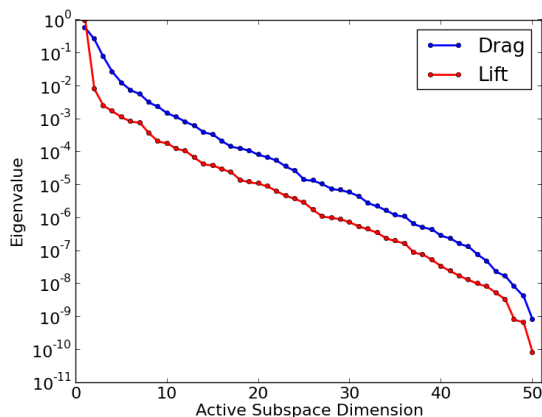


Figure 4: Eigenvalue decay for drag and lift active subspaces.

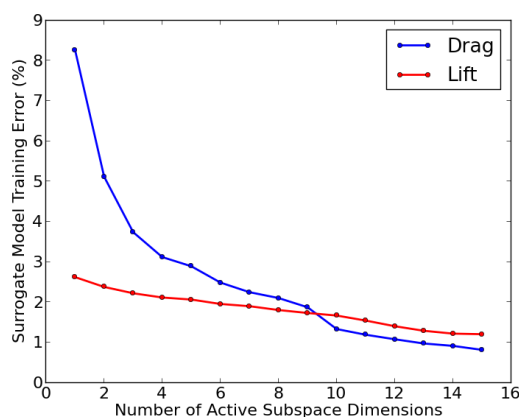


Figure 5: Training error decay for drag and lift surrogate models in active subspaces.

We confirm this behavior when we project the 300 lift samples onto the dominant active subspace in Figure 6. The dominance of the one-dimensional active subspace is observed by the small spread about

the trend. It also suggests that a linear model of lift as a function of the reduced coordinate \mathbf{y} will be a sufficiently good surrogate model.

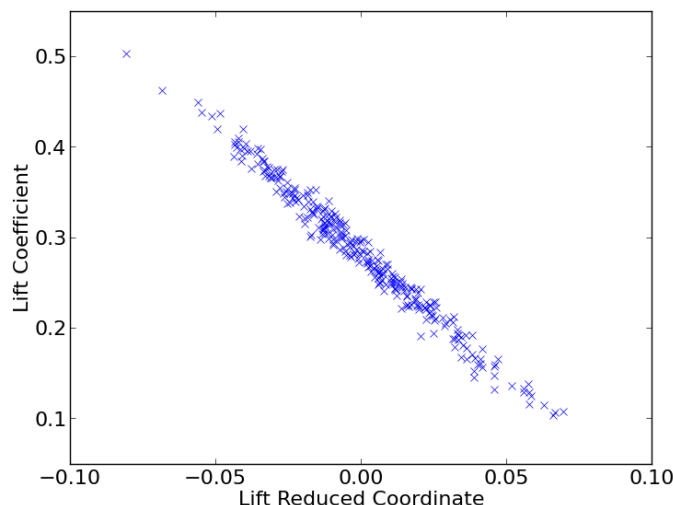


Figure 6: Lift coefficient projected into the first active subspace dimension.

Since we are only concerned with designs whose lift exceeds a threshold ($C_L \geq 0.286$), we can use a linear surrogate model on the active subspace to constrain the design space to the set of points that will likely have a sufficiently large lift. This will yield a simpler optimization problem than the general one presented in Equation 22.

In particular, if \mathbf{u}_ℓ is the eigenvector defining the one-dimensional active subspace for lift, then we model the lift as

$$f_\ell(\mathbf{x}) \approx g(\mathbf{x}) = \mathbf{x}_o + \mathbf{u}_\ell^\top \mathbf{x}, \quad (24)$$

where \mathbf{x}_o is the intercept. If we wish to constrain the design space to points where lift exceeds a threshold \mathbf{c} , then

$$\mathbf{c} - \mathbf{x}_o \leq \mathbf{u}_\ell^\top \mathbf{x} \quad (25)$$

provides an appropriate linear inequality constraint. We will include this constraint in the inverse mapping problem when operating between the full space and the active subspace for the drag.

Turning our attention to the drag, the eigenvalues from the computation (6) associated with drag are also shown in Figure 4. Note that the decay is much slower than the eigenvalues for lift. This implies that a one-dimensional active subspace is not sufficient to construct an accurate response surface for minimization. We apply a heuristic for selecting the number of dimension based on diminishing returns of accuracy. The eigenvalues suggest that after the first five eigenvectors, marginal gains in the decay in eigenvalue are found with increasing dimension. This is also demonstrated in the training error decay shown in Figure 5, where gains in training error reduction drops off after five dimensions. Thus we can make the judgment that drag is sufficiently modeled in a five-dimensional active subspace.

While the surrogate modeling approach will be capable of regressing a five-dimensional manifold, there are reasons for constraining the dimension in the practice of design exploration. In particular, if we constrain our subspace to two dimensions, we ensure two benefits: (i) the ability to re-mesh the subspace using a small number of additional model evaluations and (ii) the ability to plot insight-giving visualizations. One such visualization of the present data is shown in Figure 7. Note that in this figure the RMS error of the training data (the blue crosses) on the surrogate model (the orange surface) is approximately 5%. One additional interesting insight this visualization is able to provide is that we may be able to sufficiently model drag with a quadratic surrogate model.

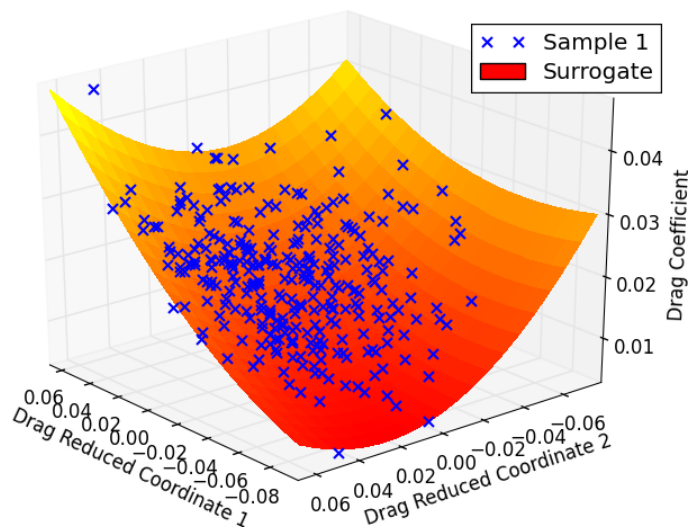


Figure 7: Surrogate model in the 2-D active subspace for drag.
Training error is 5.1%.

B. Surveying feasible regions

In the reduced space, special attention is needed to account for constraints expressed on the full space design problem. In particular, we would like to identify the convex hull(s) which partition the feasible domain of the active subspace, $\mathcal{Y} \subset \mathbb{R}^2$. A major assumption for this part of the design survey is that we have constrained the dimension of the active subspaces to two. This will require us to apply additional considerations when regularizing a subspace point into the full space.

We construct the feasible domain using the following heuristic. First, we find a bounding box on \mathbb{R}^2 for the two active coordinates according to:

$$-0.05 \operatorname{diag} \left(\operatorname{sign}(\mathbf{U}_d)^T \mathbf{U}_d \right) \leq \mathbf{y} \leq 0.05 \operatorname{diag} \left(\operatorname{sign}(\mathbf{U}_d)^T \mathbf{U}_d \right), \quad (26)$$

where \mathbf{U}_d contains the first two eigenvectors from (6) for the drag, and $\operatorname{sign}(\mathbf{U}_d)$ returns a matrix with the sign of the components.

In this domain, we consider a uniform tensor product design of experiments (i.e., a grid survey) on the bounding box with 101 points in each of the two coordinates. Not every point on this survey corresponds to a point from the full hypercube \mathcal{X} . In other words, there are several points where evaluating the function is impossible, since there may not be an \mathbf{x} in \mathcal{X} that maps to some \mathbf{y} in the design.

To address this, we remove every point from the tensor design on the bounding box that does not have a corresponding point in the full hypercube. More precisely, for each \mathbf{y} in the tensor design on the bounding box, we use the linear program outlined in Equation 21 to determine if there is an \mathbf{x} such that $\mathbf{y} = \mathbf{U}_d^T \mathbf{x}$ and $\mathbf{x} \in \mathcal{X}$. If the linear program indicates that the constraints are infeasible, then we remove the offending \mathbf{y} . The remaining points after this procedure are shown in Figure 8.

Incorporating the inequality constraint into this computation is straightforward for this design case because we found that the lift constraint is easily described by a linear surrogate model. This is especially useful since standard linear programs take both linear equality and inequality constraints.

Thus the general optimization problem described in Equation 23 simplifies into this linear program:

$$\mathbf{y} = \mathbf{U}_d^T \mathbf{x}, \quad \bar{f}_\ell - c \leq \mathbf{u}_\ell^T \mathbf{x}, \quad \mathbf{x} \in [-0.05, 0.05]^{50}. \quad (27)$$

For each \mathbf{y} in the tensor design on the bounding box, we seek an \mathbf{x} that satisfies the linear program. This creates a set of points in \mathbb{R}^2 for which we expect there are $\mathbf{x} \in \mathbb{R}^{50}$ that are both in the hypercube \mathcal{X} and will generate a wing design that produces sufficient lift.

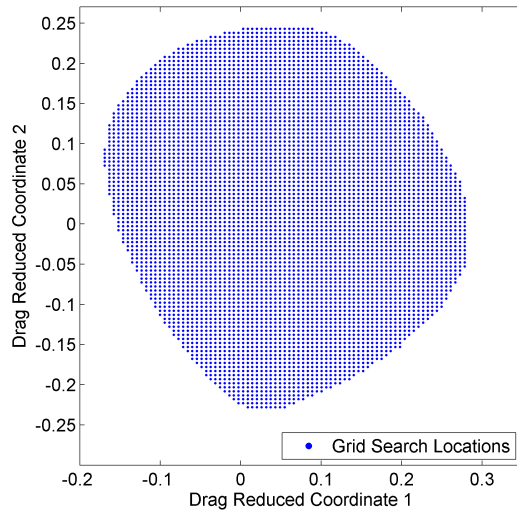


Figure 8: Polytope of drag constrained by lift, with new designs identified for sampling.

By construction, this grid survey is over-resolved. The purpose of the fine resolution was to identify the boundaries of the polytope \mathcal{Y} . To approximate the polytope, we take the convex hull of the feasible \mathbf{y} . This is a computationally inexpensive process in two dimensions. The convex hull now represents the boundary of our polytope. We coarsen the boundary and build an unstructured design of experiments (i.e., a mesh) in this boundary. The final survey on the polytope including the lift constraint is shown in Figure 9.

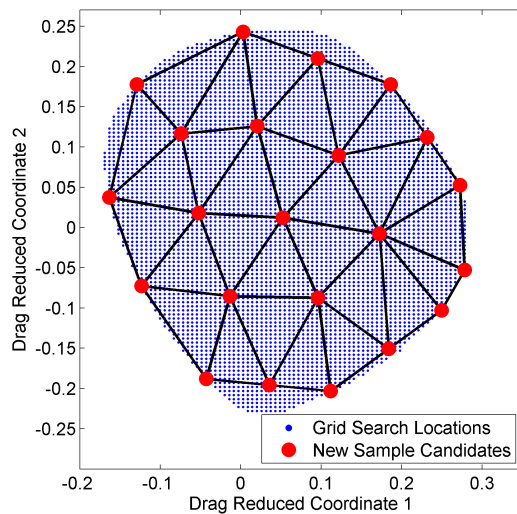


Figure 9: Polytope of drag constrained by lift, with new designs identified for sampling.

We now have identified new designs in the subspace which we will inverse-map to the full space, and evaluate with the CFD solver. The results of this second sampling will be used to construct a broader response surface in two dimensions.

For each point \mathbf{y}_k in the design on the polytope, we need to find an \mathbf{x}_k that maps to $\mathbf{y}_k = \mathbf{U}_d^T \mathbf{x}_k$ such that $\mathbf{x}_k \in \mathcal{X}$ and the inequality constraint (eq. 25) is satisfied. As stated, this map may not be unique. The accuracy of the active subspace is also hampered by the dimensionality restriction we imposed. We apply

the general regularization problem shown in Equation 22, to account for the higher-dimension behavior in drag by minimizing the expected drag while seeking a feasible point for the linear equality, linear inequality, and bound constraints.

As was suggested by the result in Figure 7, we can apply a quadratic surrogate model. We extend this result to hypothesize that we can apply a quadratic model in a five-dimensional active subspace to reasonably capture the behavior of drag. Furthermore, this allows us to formulate the mapping problem with a quadratic program.

To derive the quadratic surrogate, let $\tilde{\mathbf{U}}_d$ contain the first five eigenvectors of (6) for drag, and define $\tilde{\mathbf{y}} = \tilde{\mathbf{U}}_d^T \mathbf{x}$. We use least squares regression to compute $\mathbf{b} \in \mathbb{R}^5$ and $\mathbf{A} \in \mathbb{R}^{5 \times 5}$ such that

$$g_d(\tilde{\mathbf{y}}) \approx \mathbf{b}^T \tilde{\mathbf{y}} + \tilde{\mathbf{y}}^T \mathbf{A} \tilde{\mathbf{y}}, \quad (28)$$

where g_d is the drag as a function of the reduced coordinates $\tilde{\mathbf{y}}$. Plugging in the definition of $\tilde{\mathbf{y}}$ we have

$$f_d(\mathbf{x}) \approx g_d(\tilde{\mathbf{U}}_d^T \mathbf{x}) \approx \mathbf{b}^T \tilde{\mathbf{U}}_d^T \mathbf{x} + \mathbf{x}^T \tilde{\mathbf{U}}_d \mathbf{A} \tilde{\mathbf{U}}_d^T \mathbf{x}, \quad (29)$$

where f_d is the drag as a function of the design variables. This gives an appropriate quadratic program regularization that creates a unique map $\mathbf{x}(\mathbf{y})$ for training the response surface on \mathcal{Y} . In particular, for each \mathbf{y}_k , we solve

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && \mathbf{b}^T \tilde{\mathbf{U}}_d^T \mathbf{x} + \mathbf{x}^T \tilde{\mathbf{U}}_d \mathbf{A} \tilde{\mathbf{U}}_d^T \mathbf{x}, \\ & \text{subject to} && \tilde{\mathbf{U}}_d^T \mathbf{x} = \mathbf{y}_k, \\ & && \mathbf{u}_\ell^T \mathbf{x} \geq \bar{f}_\ell - c, \\ & && \mathbf{x} \in \mathcal{X}. \end{aligned} \quad (30)$$

This gives us an \mathbf{x}_k for each \mathbf{y}_k in the design on the polytope that (i) satisfies the hypercube bound constraints, (ii) satisfies the linear equality constraints imposed by the active subspace, (iii) satisfies the linear inequality constraint imposed by the lift threshold, (iv) minimizes a global quadratic model of the drag as a function of the design variables. We can use these \mathbf{x}_k to evaluate the drag to create training data for the response surface, and finally use the response surface to minimize the drag on the active subspace subject to the lift threshold.

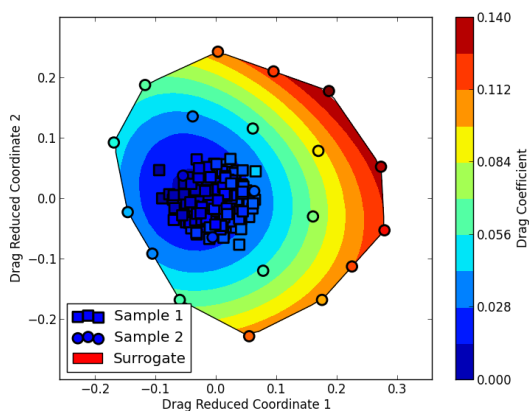


Figure 10: Surrogate model in the 2-D active subspace for drag.

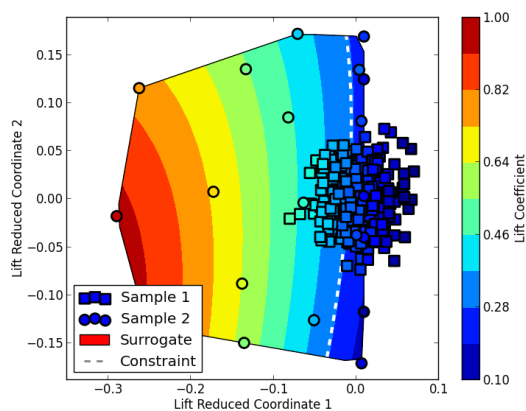


Figure 11: Surrogate model in the 2-D active subspace for lift.

The resulting surrogate models for the design problem at hand, constrained by their appropriate polytopes, are shown in Figures 10 and 11. These plots show contours of drag and lift respectively, overlaid with a scatter plot of the ONERA design samples, also colored by drag and lift. We can see that the resulting surrogate model maintains the trends of the original samples. The hull of the lift contour plot shows that the polytope, built originally in the drag subspace, has translated to the lift subspace while reasonable respecting the lift constraint.

A particularly interesting feature of the drag contour plot is that a clear local minima has appeared, which did not exist in the lift-unconstrained drag polytope (shown later in Figure 12). This minimum appears because the new samples on that side of the polytope (first reduced coordinate ≈ -0.1) are influenced by the lift constraint, and demonstrated an increase in drag over the unconstrained estimates in that region of the design space.

C. Design optimization with active subspaces

We can use surrogate models built in an active subspace to estimate optimal aerodynamic shapes. We will demonstrate this first with an unconstrained drag minimization problem, then a lift-constrained drag minimization problem. We will compare the later with gradient based optimization results.

1. Unconstrained Drag Minimization

In this example, we minimize drag without a lift constraint. Because we do not have to communicate with a separate subspace for lift, we can directly optimize on the surrogate model in the drag subspace. As identified earlier, we need five active subspace dimensions to properly capture the drag behavior. With this we can solve the following optimization problem:

$$\begin{aligned} &\text{given} && \mathbf{y} \in \mathbb{R}^5 \\ &\underset{\mathbf{y}}{\text{minimize}} && g(\mathbf{y}) \\ &\text{subject to} && \mathbf{y} \in \mathcal{Y} \text{ (recall Equation 11)} \end{aligned} \quad (31)$$

Here, g is a surrogate model constructed from the initial 300 training samples described earlier. The constraint $\mathbf{y} \in \mathcal{Y}$ is a placeholder for a secondary optimization required to check that there is a full space design \mathbf{x} in \mathcal{X} . This requires a secondary optimization problem at each evaluation of \mathbf{y} . This problem solves the linear program described in Equation 21.

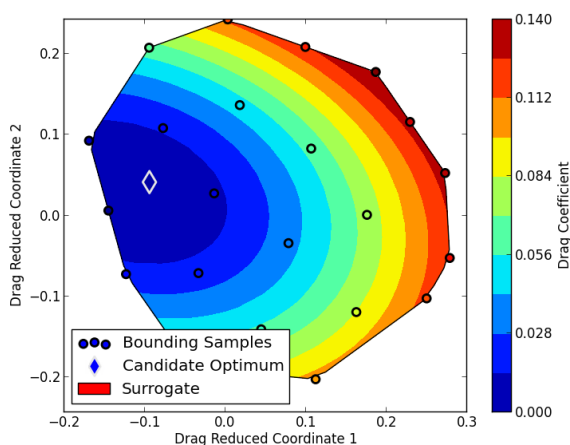


Figure 12: Surrogate model in the 2-D active subspace for drag without lift constraint.

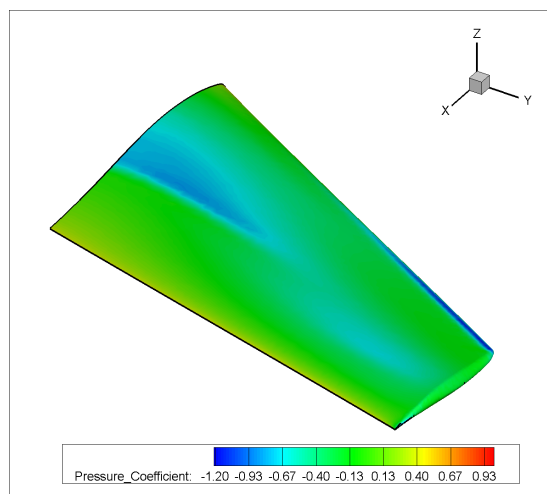


Figure 13: Surface contours of pressure coefficient for the sampled drag optimum. $C_D = 0.0045$, (61.8% reduction)

Because of the minimal computational expense, we are able to use a genetic algorithm. This demonstrates the ability to conduct *high-dimensional global optimization* using surrogate models with active subspaces. Specifically, we applied a python implementation of Covariance Matrix Adaptation (CMA).⁴

The optimization problem identified a candidate design, which we evaluated with CFD to verify the drag. The resulting plots of pressure coefficient in Figure 13 show that the shock line on the upper surface of the wing was reduced. This can be compared with the baseline contour plot provided in Figure 1. The optimum is plotted in the 2-D active subspace for visualization, showing that even in the dimension-constrained subspace this method can identify global trends and locate regions to interrogate for the global minimum.

2. Lift-Constrained Drag Minimization

In this example, we minimize drag while enforcing a lift constraint. We must now communicate between subspaces for drag and lift. As identified during the eigenvalue analysis, we need five active subspace dimensions to properly capture the drag behavior. However we have constructed our predictive model in two. We will thus construct an inverse map to connect three different active subspaces.

The first step is to select an estimated optimum of the surrogate shown earlier in Figure 10, in the two-dimensional active subspace for drag under lift constraint. This solves the first optimization problem,

$$\begin{aligned} &\text{given} && \mathbf{y}_{a1} \in \mathbb{R}^2 \\ &\text{minimize}_{\mathbf{y}_{a1}} && C_D \sim g_{a1}(\mathbf{y}_{a1}) \\ &\text{subject to} && \mathbf{y}_{a1} \in \mathcal{Y}_{a1} \\ &\text{yield} && \mathbf{y}_{select}, \end{aligned} \tag{32}$$

where again (\sim) identifies that the surrogate model g_{a1}^* is of lower dimension than required. We optimize on the surrogate model g_{a1}^* , here a GPR model, using a global optimizer, CMA, followed by a local optimizer, SLSQP. The location of this optimum is plotted on in the 2-D drag map in Figure 14.

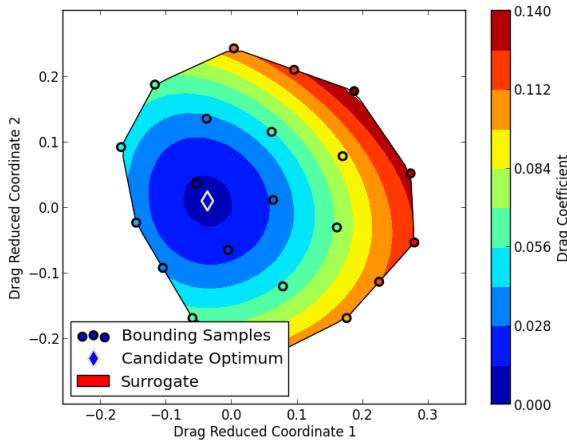


Figure 14: Surrogate model in the 2-D active subspace for drag including lift constraint.

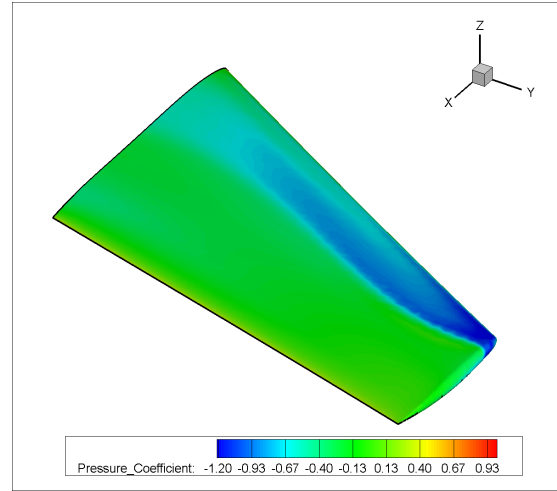


Figure 15: Surface contours of pressure coefficient for the sampled drag optimum.

$C_D = 0.0101$, (14.4% reduction)

$C_L = 0.2786$, (2.7% reduction)

With the location of this optimum identified in the 2-D drag active subspace, we use an inverse map to find a point in the full space to evaluate with CFD. This map now requires combining the knowledge from three active subspaces and surrogate models – (a_1) the 2-D drag space shown in Figure 14 with a GPR surrogate model, (a_2) the 5-D drag space with a quadratic surrogate model from Equation 29, and (b) the 1-D lift space shown in Figure 6 with a linear surrogate model. The optimization problem which solves this map is shown in the equation below.

$$\begin{aligned}
&\text{given} && \mathbf{y}_{a1} = \mathbf{y}_{select} \\
&&& C_D(\mathbf{x}) \approx g_{a2}^*(\mathbf{y}_{a2}) \\
&&& C_L(\mathbf{x}) \approx g_b^*(\mathbf{y}_a) \\
&&& \mathbf{y}_{a1} = \mathbf{U}_{a1}^\top \mathbf{x}, \quad \mathbf{U}_{a1} \in \mathbb{R}^{m \times n_a} \\
&&& \mathbf{y}_{a2} = \mathbf{U}_{a2}^\top \mathbf{x}, \quad \mathbf{U}_{a2} \in \mathbb{R}^{m \times n_a + \ell} \\
&&& \mathbf{y}_b = \mathbf{U}_b^\top \mathbf{x}, \quad \mathbf{U}_b \in \mathbb{R}^{m \times n_b} \\
&\text{minimize}_{\mathbf{x}} && g_{a2}^*(\mathbf{U}_{a2}^\top \mathbf{x}) \\
&\text{subject to} && lb_i < \mathbf{x}_i < ub_i, \quad i \in \{0, \dots, m\} \\
&&& \mathbf{y}_{a1} = \mathbf{U}_{a1}^\top \mathbf{x} \\
&&& g_b^*(\mathbf{U}_b^\top \mathbf{x}) \leq c \\
&\text{yield} && \mathbf{x}
\end{aligned} \tag{33}$$

This is a combination of the two inverse maps given in Equations 22 and 23. We have two active subspace bases \mathbf{U}_{a1} and \mathbf{U}_{a2} for drag, and one basis \mathbf{U}_b for lift. It finds a location in the full space which maps to the chosen optimum point \mathbf{y}_{a1} while minimizing the estimated drag g_{a2}^* , and constraining the estimated lift g_b^* .

With a potential design identified in the full space, we ran an additional CFD solution. The resulting surface-plot of pressure coefficient is found in Figure 13. It shows that the shock line on the upper surface of the wing was reduced. A drag reduction of 14.4% was realized, at the penalty of a 2.7% violation of the lift constraint.

The lift constraint violation is the result of the approximations made while discarding dimensions of variability during the active subspace analysis. As we saw in Figure 11, the subspace-bounding hull identified by the resampling step extended past the lift constraint. In practice, this algorithmic error would be an acceptable cost for the opportunity to perform surrogate based optimization in low dimension. The estimated minimum identified by this approach could be followed up by local optimization with gradient-based methods to enforce the lift constraint and potentially find more reduction in drag.

3. Comparison to Gradient Based Optimization

To compare the quality of the estimated minimum found by the active subspace method, we performed four gradient based analyses in the full design space, each with random start locations. The trajectories of those optimizations were projected into the 2-D drag space. These are plotted in Figure 16

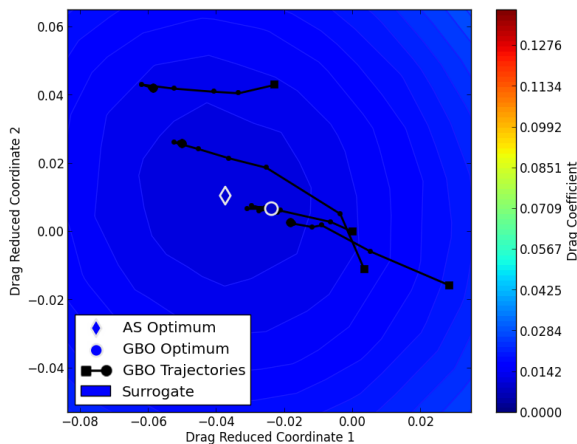


Figure 16: Surrogate model in the 2-D active subspace for drag without lift constraint.

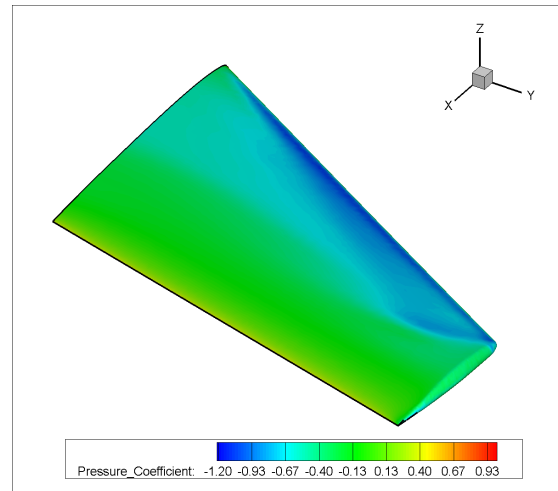


Figure 17: Surface contours of pressure coefficient for the sampled drag optimum.

$C_D = 0.0089$, (24.6% reduction)

$C_L = 0.2868$, (0.1% increase)

Each randomly started optimization converged to a slightly different minimum. This is identified by the different terminating locations plotted in the drag reduced space. It is possible that in these simulations, multiple local minima were found by the gradient based optimization because of numerical inaccuracies in the gradient information. The best minimum is plotted in Figure 17. It was able to achieve a 24.6% reduction in drag, with little violation of the lift constraint.

In the active subspace, this design was located closest to the minimum estimated by the surrogate model, which suggests that local refinement of the optimum estimated by active subspace could have found a nearly similar result.

Active Subspace		Gradient Based	
Step	Flow + Adjoint Evaluations	Step	Flow + Adjoint Evaluations
Initial Sample	300 x 3	GBO Start 1	56
Active Subspace Resample	22	GBO Start 2	28
Optimum Samples	5	GBO Start 3	41
		GBO Start 4	34
Total	925	Total	159

Table 1: Comparison of computational cost for Active Subspace- and Gradient-Based Optimization for the ONERA M6 problem.

A comparison of cost for the two approaches are shown in Table 1. The comparison is not especially fair, as more GBO restarts are needed to execute a global survey comparable to that of the active subspace approach. Additionally, the active subspace surrogate model is useful beyond this initial optimization, and could be provided to a multi-disciplinary analysis and optimization study, thus amortizing the computational costs across many studies.

V. Summary & Conclusions

In this paper we have extended the Active Subspace Method, and applied it to an example of high-dimensional constrained optimization of the ONERA-M6 wing. This approach was able to quickly identify designs with lower drag by interrogating the active subspace. The results are compelling because they suggest the existence of simple low-dimensional behaviors embedded in the high-dimensional space of a complex design problem. The method requires a large initial sample of design evaluations, which could make it more expensive than a gradient-based optimizer. However, these samples are made globally, within a large bounding hyper-box, and may enable the active subspace approach to avoid local minima that gradient-based approaches encounter with multi-modal response surfaces.

It is important to note that we have not expressed any guarantees of design optimality in the full space. Since we are conducting design optimizations above several levels of approximations and surrogate models, we believe this approach is best suited for estimating globally optimal regions with surrogate models in the active subspace, and following up with a local optimization in the full space with gradient based optimization for example.

Our application of the Active Subspace Method is enabled by an novel approach to mapping between coupled subspaces of a design problem. This is a major contribution of our work. We showed how it is possible to construct three separate reduced order surrogate models, and interrogate them in their respective subspaces, while linking them through their common embedding in the full design space.

Another contribution this paper is the application of the Active Subspace Method to enable surrogate based optimization in high dimensional aerodynamic shape design problems. It addresses problems we and many in the community have faced while using surrogate models in in high dimension. The future application of this contribution will be in multi-disciplinary optimization problems, where surrogates can enable distributed frameworks to more efficiently couple different design disciplines.

References

- ¹Palacios, 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.
- ²Kenway, G. and Martins, J., "Multi-point High-Fidelity Aerostructural Optimization of a Transport Aircraft Configuration," *Journal of Aircraft*, 2012.
- ³Kraft, D., "Algorithm 733: TOMP—Fortran modules for optimal control calculations," *ACM Transactions on Mathematical Software (TOMS)*, Vol. 20, No. 3, 1994, pp. 262–281.
- ⁴Igel, C., Suttrop, T., and Hansen, N., "A computational efficient covariance matrix update and a (1+1)-CMA for evolution strategies," *Proceedings of the 8th annual conference on genetic and evolutionary computation GECCO*, ACM, 2006, pp. 453–460.
- ⁵Lukaczyk, 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*.
- ⁶Yamazaki, 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.
- ⁷Han, 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.
- ⁸Constantine, P. G., Dow, E., and Wang, Q., "Active subspace methods in theory and practice: applications to kriging surfaces," *arXiv preprint arXiv:1304.2070*, 2013.
- ⁹Ng, A., "Principal component Analysis," *Stanford University CS229 Lecture Notes*, 2012.
- ¹⁰Ghisu, T., Parks, G., Jarrett, J., and Clarkson, P., "Accelerating Design Optimization Via Principal Components Analysis," *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Sept 2008.
- ¹¹Palacios, F., Alonso, J. J., Duraisamy, K., Colonno, M. R., Aranake, A. C., Campos, A., Copeland, S. R., Economou, T. D., Lonkar, A. K., Lukaczyk, T. W., and Taylor, T. W. R., "Stanford University Unstructured (SU2): An open source integrated computational environment for multiphysics simulation and design," *51st AIAA Aerospace Sciences Meeting and Exhibit*, Jan 2013.
- ¹²Russi, T. M., *Uncertainty Quantification with Experimental Data and Complex System Models*, Ph.D. thesis, UC Berkeley, 2010.
- ¹³Rasmussen, C. and Williams, C., *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, MA, 2006, pp. 13–30.
- ¹⁴Trent Lukaczyk, T. T., Palacios, F., and Alonso, J. J., "Managing Gradient Inaccuracies while Enhancing Response Surface Models," *Accepted, 51st AIAA Aerospace Sciences Meeting and Exhibit*, Grapevine, TX, January 2013.