SURROGATE MODELING AND ACTIVE SUBSPACES FOR EFFICIENT OPTIMIZATION OF SUPERSONIC AIRCRAFT

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF AERONAUTICS AND ASTRONAUTICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Trent William Lukaczyk
June 2015

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Juan Alonso, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Mykel Kochenderfer

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Ilan Kroo

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Paul Constantine

Approved for the Stanford University Committee on Graduate Studies.

Patricia J. Gumport, Vice Provost for Graduate Education

Abstract

This dissertation presents approaches for surrogate based optimization of supersonic vehicles analyzed with high fidelity flow simulations. It integrates several developments in surrogate modeling to enable a robust regression procedure in the presence of sparse data and inaccurate gradients. A series of hyperparameter constraints are developed which encourage the learning process to generate a physically representative fit of the data. It identifies the existence of subspaces based on linear combinations of inputs called “active subspaces” that reasonably model the behavior of objectives within aerospace design problems such as lift coefficient, drag coefficient and an equivalent area functional. Coherent physical features were found across several design problems for both two and three dimensional geometries. This dissertation further proposes an approach for adaptive refinement by conditioning the traditional expected improvement sampling criterion to avoid exploration of the design bounds. To begin work on applying active subspaces to optimization, inverse maps were developed to enable the linking of separate active subspaces for objectives and constraints, enabling surrogate based optimization in high dimension. Several design problems are explored, and it is shown that surrogate based optimization in active subspaces could enable the optimization of problems otherwise intractable via gradient based optimization alone.

Acknowledgement

First, I would like to thank my advisor, Juan Alonso, for his support and guidance. I am grateful that he gave me the freedom to pursue several projects in which I found interest and that he always supported my career choices. He has been a role model for me not only in terms of a researcher but as a lead engineer with broad and deep expertise.

I would also like to thank my dissertation committee, Paul Constantine, Mykel Kochenderfer, and Ilan Kroo for their input and support. They have contributed a diverse set of perspectives and projects for which I deeply value as a crucial part of my graduate experience.

I thank Francisco Palacios for his intense guidance during a large part of my graduate career, and the members of the Aerospace Design Lab for their advice and help with experiments for this dissertation. Especially, Thomas Economon, Sean Copeland, Brendan Tracey, Santiago Padron, Tom Colvin, Tom Taylor, Francisco Capristan, Alejandro Campos, and Andrew Wendorff.

My work would not have been possible without the support and funding of the NASA N+2 Supersonics Program and the Lockheed Martin Corporation. I am especially grateful to Michael Buonanno, John Morganstern, and Nicole Norstrud for their input and enlightening discussions.

I owe a great deal of gratitude to the teams of SUAVE-club and SUAVE-code, who helped make our respective visions of these projects a reality.

Finally, I thank my friends, Tamaki Bieri, and my parents for their invaluable emotional support.

Nomenclature

Aerodynamic quantities of interest

C_D	Drag coefficient
C_L	Lift coefficient
ΔA_e	Equivalent area functional

Design spaces

$\mathbb{R}^D$	Set of real numbers of dimension D
lb, ub	Lower and upper bounds
$\mathbf{x}$	Full-space design vector
$\mathbf{y}$	Active subspace design vector
$\mathcal{X}$	Space of $\mathbf{x}$
$\mathcal{Y}$	Space of $\mathbf{y}$
$f(\cdot)$	Scalar function in full space
$g(\cdot)$	Scalar function in active subspace
$\nabla_{(\cdot)}$	Gradient of scalar function

Active subspace formulation

$\boldsymbol{C}$	Average outerproduct of the gradient with itself
$\mathrm{E}\left[\cdot\right]$	Expected value of a random quantity
$\boldsymbol{W}$	Column matrix of eigenvectors
$\boldsymbol{\Lambda}$	Diagonal matrix of eigenvalues
$\boldsymbol{U}$	Active subspace basis, column matrix
$\boldsymbol{V}$	Inactive subspace basis

m	Design full-space dimension size
n	Active subspace dimension size

Surrogate modeling

$\text{Cov}\left[\cdot,\ \cdot\right]$	Covariance between two random quantities
$\mathcal{R}$	Response surface estimator
$\mathcal{N}$	Normal distribution
μ	Mean
$[\sigma]$	Standard deviation
$k(\cdot,\cdot)$	covariance function
s	Estimated variance
θ	Hyperparameter
$\approx$	Approximation
$\sim$	Less accurate Approximation
M	Number of training samples
N	Number of predicted samples

Subscript

i, j, p, q	Variable number
∂	Gradient model
ϵ	Noise model

Superscript

$*$	Estimated variable

Design Vector Subscripting

$\mathbf{x} = x_j, \quad j = \{1, ..., N\}$

Design Matrix Subscripting

$\boldsymbol{X} = \mathbf{x}_i = x_{i,j}, \quad i = \{1, ..., N\},\ j = \{1, ..., M\}$

Contents

Abstract **v**

Acknowledgement **vii**

Nomenclature **ix**

1 Introduction **1**

1.1 Motivation 2

1.2 Low-Boom Design 3

1.3 Optimization Approaches 6

1.4 Surrogate Modeling 8

1.5 Active Subspaces 10

1.6 Contributions 11

1.7 Outline 12

2 Design Cases **15**

2.1 Supersonic Boom Relationships 16

2.2 Analytic Examples 17

2.2.1 Test Functions 18

2.2.2 Thin Supersonic Airfoil 21

2.3 Freeform Deformation 22

2.4 Two Dimensional Flow Examples 23

2.4.1 NACA 0012 Airfoil 23

2.4.2 Biparabolic Airfoil 25

2.5 Three Dimensional Flow Examples 31
2.5.1 ONERA M6 Transonic Wing 31
2.5.2 Langley Supersonic Business Jet 33
2.5.3 N+2 Supersonic Passenger Jet 38

3 Surrogate Modeling **43**
3.1 Background 43
3.2 Method Formulation 46
3.2.1 Bayesian Foundation 46
3.2.2 Covariance Function 47
3.2.3 Fit Evaluation 56
3.2.4 Hyperparameter Selection 57
3.3 Numerical Experiments 59
3.3.1 Analytical Test Problem 60
3.3.1.1 A Fitting Example 60
3.3.1.2 Characterization of Learning Robustness 62
3.3.1.3 Characterization of Learning Efficiency 66
3.3.1.4 Effects of Simulated Gradient Errors 70
3.3.2 Surrogates on Flow Simulations 71
3.3.2.1 Gradient Error Characterization 71
3.3.2.2 Response Surfaces with Noise Tolerance 73
3.4 Summary 76

4 Active Subspaces **77**
4.1 Background 77
4.2 Method Formulation 78
4.2.1 Subspace Construction 78
4.2.2 Subspace Selection 83
4.2.3 Deformation Modes 84
4.2.4 Methods Summary 85
4.3 Numerical Experiments 86
4.3.1 Biparabolic Airfoil 86

4.3.1.1 Active Subspace Properties 87
4.3.1.2 Connection to Fundamental Aerodynamics 97
4.3.2 Langley Supersonic Business Jet 100
4.3.2.1 Active Subspace Properties 100
4.3.2.2 Physical Insights 104
4.3.3 N+2 Supersonic Passenger Jet 107
4.3.3.1 Active Subspace Properties 107
4.3.3.2 Physical Insights 110
4.4 Summary 113

5 Optimization 115
5.1 Background 115
5.2 Method Formulation 117
5.2.1 Surrogate Based Optimization 117
5.2.2 Optimization in Active Subspaces 121
5.3 Numerical Experiments 128
5.3.1 Random Start Gradient Based Optimization 128
5.3.2 Surrogate Based Optimization 132
5.3.2.1 Rosenbrock Function - Minimization 132
5.3.2.2 Rosenbrock Function - Minimization with Constraint on the Rastrigin Function 134
5.3.2.3 NACA 0012 Airfoil - Drag Minimization with Lift Constraint 137
5.3.2.4 Supersonic Biparabolic Airfoil - Drag Minimization with Equivalent Area Constraint 138
5.3.2.5 N+2 Supersonic Passenger Jet - Unconstrained Drag Minimization 142
5.3.3 Active Subspaces for Surrogate Based Optimization 144
5.3.3.1 Analytical Test Case 144
5.3.3.2 ONERA M6 - Unconstrained Drag Minimization . . 146
5.3.3.3 ONERA M6 - Drag Minimization with Lift Constraint 148

5.3.3.4 Biparabolic Airfoil - Drag Minimization with Equivalent Area Constraint 153
5.3.3.5 Langley Supersonic Business Jet - Drag Minimization with Lift and Equivalent Area Constraints 157
5.4 Summary 162

6 Conclusion 163
6.1 Summary 163
6.2 Contributions 165
6.3 Future work 167

A Appendix 169
A.1 Supersonic Thin Airfoil Theory 169
A.1.0.6 Lift Coefficient 170
A.1.0.7 Drag Coefficient 171
A.1.0.8 Equivalent Area 171
A.1.1 Freeform Deformation 173

Bibliography 175

List of Tables

3.1 Kernel function data layers. 54
3.2 Non-linear hyperparameter constraints for MLE maximization. 58
3.3 Error data for example surrogate model with increasing bias error. . . 71
3.4 Baseline mesh gradient errors for the NACA0012 example. 72

5.1 Comparision of computational cost for Active Subspace- and Gradient-Based Optimization for the ONERA M6 problem. 152
5.2 Configuration of z-space constraints. 154
5.3 Vehicle performance results for the optimization problem. 159

List of Figures

1.1 Schematic of CFD-based Supersonic Boom Propagation Simulations . 4

2.1 Example test functions in two dimensions. 19
2.2 Relevant geometry for the biparabolic design problem. 20
2.3 Sample solution for direct and adjoint problems of the NACA 0012 airfoil. 24
2.4 Parameterization and example deformation for the NACA 0012. . . . 24
2.5 Parameterization and example deformation 26
2.6 Relevant geometry for the biparabolic design problem. 27
2.7 Mesh topology for the biparabolic design problem. 27
2.8 Gradient verification for the biparabolic design problem. 29
2.9 Flow and adjoint results for the baseline biparabolic airfoil. 30
2.10 Relevant geometry for the biparabolic design problem. 32
2.11 An example deformation of the ONERA M6 using freeform deformation boxes. 32
2.12 Langley Supersonic Business Jet (LSBJ) geometry 33
2.13 Parameterization of the Langley supersonic business jet. 34
2.14 Selected views of the LSBJ mesh. The mesh contains 1.3 million cells. 36
2.15 Example of mirrored surface solutions for the Langley SBJ. 37
2.16 N+2 Supersonic Passengjer Jet (SPJ) geometry 38
2.17 Baseline N+2 Supersonic Passenger Jet Geometry 39
2.18 Mesh and parameterization for an N+2 drag minimization 40
2.19 A visualization of the flow solution and mesh for the N+2 Supersonic Passengjer Jet (SPJ) geometry, including engines. 41

2.20 Control points for the N+2 supersonic passenger jet problem. Features are mirrored. 42

3.1 Plots of a gradient-enhanced GPR model of the Rosenbrock function. 60
3.2 Plots of a GPR model of the Rosenbrock function, *without gradients.* 61
3.3 Plots of a gradient-enhanced GPR of the modified Rastrigin function. 61
3.4 Surrogate model of a parabola $f = x^2$ with 10 samples. 62
3.5 Learning history for the parabolic problem, using GPML, without hyperparameter constraints. 63
3.6 Learning history for the parabolic problem, using VyPy, without hyperparameter constraints. 64
3.7 Learning history for the parabolic problem, using VyPy, including hyperparameter constraints. 65
3.8 Variation of training error with increasing number of samples, while fitting a GPR surrogate model to the Rosenbrock function in different numbers of dimensions. 67
3.9 Variation of testing error with increasing number of samples, while fitting a GPR surrogate model to the Rosenbrock function in different numbers of dimensions. 69
3.10 Response surfaces for an example regression with inaccurate gradients. 70
3.11 Baseline mesh objective and gradient sweeps 73
3.12 Detail of finite differencing noise. 74
3.13 Example RSM with adjoint gradients, noise tolerance=1×10^{-10} . . . 75
3.14 Example RSM with adjoint gradients, noise tolerance=1×10^{-3} . . . 75
3.15 Example RSM with finite differencing, noise tolerance=1×10^{-3} . . . 75

4.1 Eigenvalue spectra of active subspaces for the biparabolic airfoil. . . . 90
4.2 Sufficient summary plots for the first active subspace of the biparabolic airfoil. 91
4.3 Physically relevant deformation modes for the biparabolic airfoil. . . . 92
4.4 Cross sections of surrogate models plotted in the active subspace. . . 94

4.5 Training and testing error trends for surrogates built in active subspaces of the biparabolic airfoil. 95

4.6 Comparison of thin airfoil theory for lift with the first direction's sufficient summary plot. 97

4.7 The first four drag active subspaces combined to yield an angle of attack mode. 98

4.8 Comparison of thin airfoil theory with biparabolic airfoil active subspaces. 99

4.9 The first four drag active subspaces combined to yield a camber mode. 100

4.10 Eigenvalue spectra for the langley supersonic business jet. Only the first 20 of 192 eigenvalues are displayed. 102

4.11 Sufficient summary plots for the first active subspace of the langley supersonic business jet. 103

4.12 First deformation modes for the Langley supersonic business jet discovered by an active subspace analysis. 105

4.13 Physically relevant deformation modes for the Langley supersonic business jet discovered by an active subspace analysis. 106

4.14 Eigenvalue spectra for the N+2 supersonic passenger jet. 108

4.15 Sufficient summary plots for the first active subspace of the N+2 supersonic passenger jet. 109

4.16 First deformation modes for the N+2 supersonic passenger jet. 112

5.1 Surrogate based optimization process. 118

5.2 The inverse mapping problem. 122

5.3 Summary of random-start gradient based optimizations. 130

5.4 Gradient based optimization corresponding to sample 5 in Figure 5.3a. 131

5.5 Gradient based optimization corresponding to sample 7 in Figure 5.3b. 131

5.6 Convergence of Unconstrained 2D Rosenbrock Problem 133

5.7 Comparison of Unconstrained GBO and SBO 134

5.8 Comparison of Constrained GBO and SBO 135

5.9 Example learning surfaces made with 6 samples for the optimization problem using Rosenbrock and Rastrigin test functions. 136

5.10 Example learning surfaces for the NACA0012 constrained problem at the conclusion of the optimization. 137
5.11 Biparabolic airfoil pressure contours. 139
5.12 Equivalent area distributions for constrained and unconstrained drag SBO. 140
5.13 Convergence history: biparabolic airfoil. 140
5.14 Objective and constraint history: biparabolic airfoil. 141
5.15 Comparison of unconstrained drag GBO and SBO 142
5.16 Original and deformed N+2 surfaces, unconstrained drag minimization. 143
5.17 Contours of pressure coefficient for the baseline and optimized design after surrogate based optimization with GPR. 143
5.18 Surrogate models in the original domain. 145
5.19 Surrogate models in the active domain. 145
5.20 Results for an unconstrained drag minimization in with active subspaces.147
5.21 Results for a lift-constrained drag minimization in active subspaces. . 149
5.22 Gradient based optimization trajectories compared to active subspaces. 151
5.23 Summary of GBO optimizations started from locations predicted by surrogate based optimization in active subspaces, for various configurations of the inactive space constraint. 155
5.24 Results for a surrogate based optimization in active subspaces, used to start a gradient based optimization. 156
5.25 Deformation of SBO estimated optimum. 160
5.26 Comparison of baseline and optimized designs. 161

Chapter 1

Introduction

Optimization is a fundamental tool in aircraft design. It drives improvements in performance metrics by iteratively changing design parameters, allowing algorithms to automatically find a favorable design. In the context of engineering, tools like these identify non-intuitive results during the early stages of design. Traditionally in the "preliminary" design stage, broad stroke choices are made using handbook correlations and analytical relationships. High fidelity analysis was not appropriate here because it was slow and expensive by comparison. But when a truly unique configuration is needed to solve a real world problem, high fidelity simulations almost become a requirement.

This dissertation is concerned with enabling high fidelity design optimization in the preliminary design phase. Such an application is possible today because simulation tools have matured in efficiency both in terms of designer time and computer time. The time required of a designer to move a napkin sketch to computer modeled geometry and then to discretized fluid ready for simulation is considerably short, thanks to interfaces that help one express more design information with fewer mouse clicks. The time required to run a simulation on modern clusters is also considerably short, and it is now reasonable to simultaneously run multiple multi-million cell fluid simulations.

But resources are still limited, and pressure grows stronger to discover new designs faster, which drives a need to remain efficient. Furthermore, in the conceptual design stage it is important to explore the global space of designs, lest one misses an island of significantly better performance. This is why the community is investigating methods for efficient global design optimization.

1.1 Motivation

For five years, NASA has sponsored an advanced concept study for commercial supersonic transports entering service in the 2018-2020 period. A cornerstone objective for the project was to validate integrated airframe and propulsion technologies, as well as validate design methodologies to realize a supersonic vehicle capable of meeting the "N+2" environmental and performance goals [1].

The motivating problem of this thesis is the shape design of the N+2 Supersonic Passenger Jet. The N+2 goals specify it is to be capable of flying over land at Mach 1.6-1.8, with a range of at least 4000 nmi and a capacity of approximately 35-70 passengers. The loudness target is a perceived loudness level of 85 PLdB, which is slightly quieter than a power lawnmower [2]. The cruise emissions target is lower than 10 grams of nitrous oxides per kilogram of fuel burned, and the fuel efficiency target is more than 12 passenger-kilometers per kilogram of fuel burned (20 passenger-miles per gallon) [1].

These criteria have been built out of a concerted effort by the community to restore civil supersonic transport capability since the retirement of Concorde. Before its final flight in 2003, this vehicle could carry between 92 and 120 passengers at twice the speed of sound from London to the Caribbean. Only 14 were sold, short of a goal of at least 100 [3]. Two major reasons hampered its widespread sale. First was its expensive level of fuel consumption at 8 pax-km/kg-fuel (15 pax-MPG). For comparison the Boeing 737 can fly at 45 pax-km/kg-fuel. Perhaps most importantly however was Concorde's very loud sonic boom, which led governments to ban it from flying over land. Even while cruising at 50,000 feet, its perceived loudness on the ground of 120 PLdB was comparable to listening to a jackhammer one meter away, which was unacceptable for routine flights over populated areas.

To advance the design of low-boom supersonic passenger jet technology, Stanford University collaborated with the Lockheed Martin Corporation as part of the N+2 Supersonics Project. Stanford University contributed to this effort by developing an aerodynamic optimization 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 work presented in this dissertation has grown within the supersonics project to demonstrate a real application for new optimization methods.

1.2 Low-Boom Design

As an aircraft flies supersonically, it sheds a system of shock waves that generate a signature of compressed and rarefied pressure. As depicted by Figure 1.1, these shocks propagate through the atmosphere, and under various processes will coalesce and stretch as they travel to towards the ground. If this behavior is unaccounted for in design, when the pressure signature reaches the ground it will have an N-shape wave, with two very strong shocks. These disturbances travel along cones characteristic to the flight Mach number, which reduces the spatial attenuation of noise. For example in a constant medium the signature would attenuate in amplitude only as a function of the square root of the distance from the vehicle. This is much slower compared to spherical attenuation which would occur subsonically, a function of the cubic root of distance. Noise propagation through a layered atmosphere is considerably more complex, as temperature and density changes contribute to attenuation. However the combination of coalescing shocks and reduced spatial attenuation mean that supersonic pressure disturbances can still be quite strong by the time they reach the ground.

Boom reduction literature has grown with a rich history of contributions starting in mid-1900. The genesis of sonic boom design is accredited to the collective works of Jones [4], McLean [5], Seebass [6], and George [7], who extended linearized supersonic theory of Whitham [8] and Hayes [9] to predict pressure signatures in the nearfield based on equivalent bodies of revolution, and then apply non-linear shock steepening

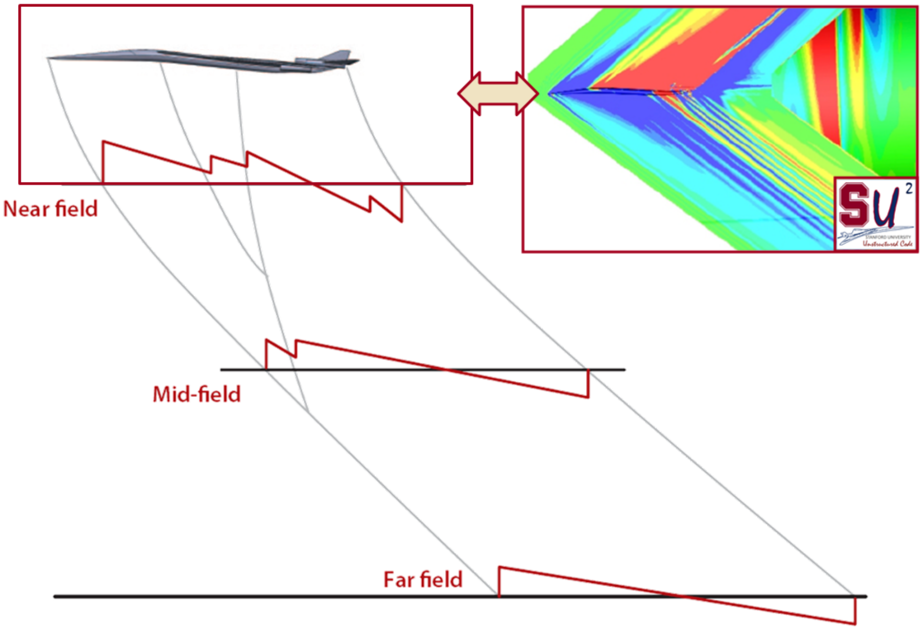


Figure 1.1: Schematic of CFD-based Supersonic Boom Propagation Simulations

methods to estimate the evolution of the signatures as they reach the ground. For a time, it was thought that the N-wave was an inevitable result of supersonic flight, but an important realization by McLean identified that in a stratified atmosphere with isothermal regions it is possible to find signatures that propagate slowly enough such that they only evolve partially before reaching the ground (noted by the midfield region shown in Figure 1.1). This enabled many early efforts to propose nearfield signatures optimized to reduce boom noise on the ground [7, 10, 11].

Propagating pressure signatures is an important component of sonic boom analysis and has been extensively studied. Most modern approaches build from an algorithm originally proposed by Thomas [12]. Relevant implementations today include PC-BOOM [13], and NFBOOM [14], among others [13]. In general this propagation procedure is executed for multiple azimuthal angles, and is used to characterize a full "boom carpet" on the ground. This carpet is typically not evaluated past 60 degrees from the symmetry plane because non-linear atmospheric effects actually refract the signature upward into space.

Quantifying boom loudness is an on going area of research, though the community has begun settling on a measure of "perceived loudness in decibels", or PLdB, as with for example the work by Morgenstern [15], Wintzer [16], Haas [17] and others [1, 18, 19, 20, 21]. Several flavors of percieved loudness exist [14, 22, 23, 24]. In general it is measured by extracting a pressure signature, processing it with Fourier analysis to find its power spectrum, and mapping this through a regularization strategy that is shown to correlate with the subjective human perception of loudness [25].

For applying supersonic boom theory to high-fidelity design, the community in general is currently separating the problem into two sub-problems. The first identifies nearfield pressure signatures that correspond to favorable perceived loudness [15, 26, 27], and the second identifies aircraft geometries that demonstrate a reasonable match to this signature [26, 28]. This decoupling allows the use of gradient-free optimization techniques on the propagation problem, which can identify optima even in the presence of noisy design spaces for example identified by Chung [29], while also allowing efficient use of high-fidelity flow simulation in the nearfield.

This dissertation will operate primarily in the nearfield, and use inverse shape design on an azimuthal series of target *equivalent area distributions*. An equivalent area distribution can represent the aircraft's pressure signature as a body of revolution that would generate the same signature, in a one-to-one and on-to mapping under linearized supersonic theory at a given azimuthal location[6]. Applying the equivalent area formulation integrates the pressure signature through an Abel transformation, and has the favorable effect of smoothing out the design space. The application of equivalent area is accordingly fundamental to the design approach executed in this dissertation.

The shape of the aircraft is changed to drive its equivalent area distribution to match the target, defined by the baseline design in this work. Flow analyses are evaluated all the way to a near-field location, typically between 2 and 3 body lengths beneath the aircraft, where the pressure signature is extracted. It is now common to not only apply Computational Fluid Dynamics (CFD) simulations to predict the nearfield equivalent area, but to also apply adjoint methods to efficiently calculate gradients needed for optimization algorithms. This has been demonstrated by the community for various design problems in supersonics [30, 31, 32, 33].

Many efforts have been built around these various simulation approaches to accomplish mission level design of supersonic vehicles especially by Kroo *et al.* [16, 34, 26]. The application of the approaches presented in this dissertation are within the context of aerodynamic design, and applications to mission level design and are left as an area of future work. For additional context of the background of supersonic design, many reviews of the literature are available [1, 35, 36].

1.3 Optimization Approaches

In the field of optimal shape design, the goal is to iteratively change 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 [31, 37]. The core impetus for these large dimension spaces is that one cannot know *a-priori* which variables are needed to most efficiently optimize the design. Thus parameterizations are constructed with many variables in order to finely control the shape of the vehicle. 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 [38]. Global optimizers including Genetic Algorithms (GAs) and Covariance Matrix Adaptation (CMA) can identify local basins that may contain the global minimum, but require large numbers of function evaluations, and are intractable in high-dimensional design spaces [39]. 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 [40, 41, 42].

To address the high computational cost of the CFD simulations and to enable global design space optimization, this dissertation presents surrogate modeling with gradient-enhanced Gaussian Process Regression (GPR), adapted for high dimensional problems using the Active Subspace Method (ASM). Gradient information available from an adjoint solution is used to increase the accuracy of the surrogate at low computational cost.

Alternatively to global optimization, there are many approaches built on 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. As a baseline method, this dissertation applies a sequential quadratic programming (SLSQP) optimizer built into the Scientific Python (scipy) toolbox [38, 43, 44]. 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 constrained 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 [38, 44].

The optimization methods in this dissertation rely on the calculation of sensitivities of the objective and constraints in the design space. Adjoint formulations and finite differencing are investigated as part of the goal of examining their effect on optimization efficiency. Finite differencing is generally associated with high computational costs. If all objectives are solved in a single flow solution, the sensitivity of n objectives to m design variable requires the solution of m additional mesh perturbations and flow solutions for a first order approximation. 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.

In comparison, adjoints are efficient and accurate, but 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 geometric 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[45].

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 [46]. It will be shown through this work the inaccuracy expected in continuous adjoints is sufficient to interfere with the construction of surrogate models, but that it can managed within the surrogate formulation so as to mitigate interference.

1.4 Surrogate Modeling

Instead of using the output of CFD-based function values and adjoint gradients directly as an input to a gradient based optimizer, it is possible to use the predictions and their gradients with respect to user-specified design parameters to create inexpensive representations of the behavior, known as surrogates models. These surrogate models can be adaptively improved to guide optimization efforts in a way that minimizes the overall computational cost required to complete an optimization run.

A large body of work has been built around surrogate modeling techniques, especially using a stochastic modeling technique known as Kriging [47, 48] and its gradient enhanced relative known as Co-Kriging [29, 41]. 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 [48, 49].

However, GPR is built within the context of machine learning, which allows more flexibility when dealing with complex design spaces [50]. Additional background for these approaches and further review of the surrogate modeling literature will be provided in Chapter 3.

To generate an initial response surface, a set of designs are sampled using a Design of Experiment (DOE) technique such as Latin Hypercube Sampling. Because the locations of these points are not dependent on each other, they can be evaluated in parallel, dramatically reducing wall-clock time if the resources are available. After this initial sample, additional design points can be chosen using Infill Sampling Criteria (ISC). It is common to take advantage of the uncertainty information available from stochastic response surface models like GPR to efficiently add points to improve the accuracy of the model near regions of optimal design [51].

An equally large body of work has been built around applying surrogate models to Surrogate Based Optimization (SBO). Jones and Forrester each provide a thorough summary of different infill sampling criterion [51, 52]. Several studies describe its application to aircraft design problems [42, 19, 16, 32]. In this work, the use of ISCs was explored to further improve the SBO process using a hybrid infill sampling criteria with expected improvement and estimated optimum [40].

A key assumption in gradient-enhanced surrogate modeling important to the current work is that the correlation of all input information can be modeled by a covariance function [53]. Typically when performing gradient enhanced surrogate modeling with Kriging or GPR, an exact correlation model is used to relate the function and its gradients [41]. As identified by Dwight [54], violations of this model have adverse effects on the quality of the fit. Decorrelation of objectives and gradients can affect 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.

A key problem that will identified, explored and addressed in this dissertation is the quality degradation of optimization methods that occurs when using function and gradient data that are decorrelated because they possess varying levels of exactness. 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. This factor 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.

1.5 Active Subspaces

Surrogate modeling techniques such as Gaussian Process Regression 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 dissertation 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) [55]. The approach learns the linear subspace that best describes the average response of the objective to perturbations of the input 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. These coordinates are referred to as the "active coordinates".

This approach bares well on 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.

The approach is contrasted to Principal Component Analysis (PCA), also known as Proper Orthogonal Decomposition (POD) [56]. 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) [57]. 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's 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. This takes advantage of the case that the input dimension reduction is intimately tied to the chosen quantity of interest. 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. Surrogates can be constructed, for example using Gaussian process regression, if one is willing to accept additional inaccuracies associated with collapsing dimensions. It will be shown that in aerospace problems that these additional inaccuracies are generally small enough to admit a useful surrogate that can accelerate optimization.

1.6 Contributions

This section provides a high-level summary of the primary contributions of this dissertation. The central theme of this work is to develop methods to accelerate optimization with high fidelity simulations.

First, this dissertation contributes the perspective that inaccurate gradients can and should be taken into account within gradient enhanced surrogate modeling. To that end a methodology is presented that can use inaccurate gradients under a new configuration of noise hyperparameters within Gaussian Process Regression (GPR). More broadly this enables the use of continuous adjoint gradients from computational fluid dynamics (CFD) simulations for surrogate based optimization of aerospace design problems.

Second, this dissertation builds upon existing surrogate based refinement strategies to suggest an algorithm for efficient optimization composed of two phases: a global refinement and a local refinement of the surrogate. A heuristic is developed for transitioning from the first phase to the second phase, and declaring convergence in the second phase.

Third, in applying the active subspace method this dissertation discovers the presence of active subspaces for supersonic design problems, enabling the reduction of large dimensional design spaces of order 200 variables to low dimensional subspaces of order 5 variables, and further enabling the construction of surrogate models for high dimensional design problems. This contribution continues on to characterize the physical behaviors found in these active subspaces and verifies their existence by drawing connections to fundamental aerodynamics.

Fourth, an algorithm is proposed for optimization in high-dimension with surrogate models built in reduced dimension using active subspaces. A core part of this algorithm is the construction of several inverse maps that enable the coupling of individual subspaces for each quantity of interest. This contribution identifies the potential for active subspaces to breach the curse of dimensionality within surrogate based optimization.

Finally, many of the algorithms of this dissertation have been deposited in open source, including interfaces for automatic evaluation of flow simulations, a new package for Gaussian process regression, and contributions to an existing package for active subspace methods.

1.7 Outline

This chapter has introduced the primary motivation and background of the dissertation. The remainder of the dissertation will proceed in the following manner.

Chapter 2 will present the relevant background information and configuration for analytic test cases, as well as cases simulated in two and three dimensions, which are interrogated for optimization in this dissertation.

Chapter 3 will present additions to the Marginal Likelihood Maximization learning algorithms for Gaussian Process Regression that increase the robustness of surrogate model generation, and enable the use of inaccurate gradients.

Chapter 4 will identify and characterize active subspaces for supersonic problems, including their connections to fundamental aerodynamics. Three levels of complexity within supersonic design are considered: a two dimensional airfoil, a simplified three dimensional business jet, and a fully complex three dimensional passenger jet.

Chapter 5 will present two optimization algorithms. First, an algorithm for efficient global optimization in low-dimension using surrogate models, composed of two phases: a global refinement and a local refinement of the surrogate. And second, an algorithm for efficient global optimization in high-dimension with active subspaces, using inverse maps that enable the coupling of multiple subspaces that are embedded in a full space.

Chapter 2

Design Cases

This dissertation will visit several classes of design problems. They are introduced here, including their setup, baseline solutions, and optimization problem formulations. There are three classes of examples - analytic functions, two dimensional flow problems, and three dimensional flow problems. The analytic examples contain standard test functions like the Rosenbrock and Rastrigin functions, as well as two functions constructed to demonstrate active subspaces. The two dimensional flow problems include the NACA 0012 and Biparabolic airfoils. The three dimensional flow problems include the ONERA M6 wing, the Langley supersonic business jet, and the N+2 supersonic passenger jet.

For all flow simulation work, this dissertation uses the open-source solver SU2 developed in the Aerospace Design Laboratory at Stanford University [58]. SU2 is a general purpose partial differential equation solver equipped with tools for optimal shape design including flow and adjoint solvers, free-form mesh deformation, 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.

2.1 Supersonic Boom Relationships

A design goal that is incorporated in several cases of this dissertation is to improve the drag performance of an aerodynamic shape, while maintaining a similar level of boom loudness on the ground. In order to avoid evaluating a full boom propagation analysis, this dissertation employs a formulation that maintains the boom loudness implicitly by constraining the nearfield equivalent area.

As introduced in the previous chapter, a significant amount of historical work has led to a strong understanding of the relationship between boom loudness and equivalent area. Specifically, it addresses the case that there are several nearfield pressure signatures that can map to a particular boom loudness. To an approximation, changes in the equivalent area distribution are known to correlate with changes in the boom loudness. A second level benefit of the equivalent area is that it smooths out the nearly discontinuous changes in nearfield pressure signatures. Several target equivalent area distributions that account for volume and lift requirements have been proposed based on linear supersonic potential theory. Recent efforts in the community have executed design studies in which have identified aircraft with favorable boom performance [18].

The equivalent area relationship is given as an Abel transformation on the static pressure disturbance p over freestream static pressure p_∞,

$$A_e(x,\phi;r) = \frac{4\sqrt{2\beta r}}{\gamma p_\infty M_\infty^2} \int_0^x (p(t,\phi;r) - p_\infty)(x-t)^{1/2} dt, \tag{2.1}$$

where $\beta = \sqrt{M_\infty^2 - 1}$ is the Prandtl-Glaurert factor, and M_∞ is the freestream Mach number [35]. The integral is evaluated along a line of interest parallel to the freestream, displaced along the line with distance x, at the radius r typically from the moment reference center, and azimuthal angle ϕ from the aircraft symmetry plane. In the case of a two-dimensional airfoil, the azimuthal angle is zero.

If an appropriate functional is constructed on equivalent area, it is possible to apply automatic shape design approaches to improve performance without compromising boom. In the case of CFD simulations within this dissertation, it is a primary requirement to have access to efficient gradients. Through the work of Palacios [31], an equivalent area formulation was constructed based on the equivalent area difference from a target distribution,

$$\Delta A_e = \frac{1}{2\Gamma L} \int_{-\Gamma}^{\Gamma} \int_{0}^{L} \left[A_e(x,\phi) - A_t(x,\phi)\right]^2 dx\, d\phi, \tag{2.2}$$

which is evaluated along the nearfield sampling cylinder with radius r, between azimuthal bounds Γ typically at $\pm 60^\circ$. This cost function is an L_2 norm of the current and target equivalent area, and becomes zero when the equivalent areas match.

It is possible to construct many other types of nearfield boom constraints. In the case of the work of Palacios, it was found that the above formulation made the development of a continuous adjoint much more tractable. Having such an adjoint capability is crucial as it makes the calculations of all surface sensitivities comparable to that of the direct flow solution.

In practical optimization studies, the equivalent area cost function ΔA_e is implemented as an inequality constraint in order to provide the optimizer freedom to explore designs. The threshold of the constraint is chosen by hand to indirectly control the azimuthal and streamwise equivalent area errors $A_e(x,\phi) - A_t(x,\phi)$. A rule of thumb used through this thesis is that an error of approximately 0.1% between the equivalent area and target at any streamwise and azimuthal location will result in a negligible change in boom loudness.

2.2 Analytic Examples

These analytic test functions are used to quickly evaluate the surrogate modeling and active subspace analysis methods. In general, analytic gradients are available, which provide smooth and noise-free results.

2.2.1 Test Functions

Rosenbrock

A standard analytical test case is the Rosenbrock function [59]. It is a function built of polynomials with a basin that is bent in a shape of a banana. Optimizers sometimes struggle with the differences in scales that occur around the minimum. The Rosenbrock function used in this dissertation is defined as follows:

$$\begin{gathered} f(\mathbf{x}) = \sum_{i=1}^{d-1} \left[100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2 \right], \\ -2.0 < x_i < 2.0, \quad i = \{1, ..., d\} \end{gathered} \tag{2.3}$$

over a design vector $\mathbf{x}$ with dimension d and scalar components x_i. Further, it has gradients defined as:

$$\frac{\partial f}{\partial x_i} = \left[-400 \left(x_{i+1} + x_i^2 \right) x_i - 2 \left(1 - x_i \right) \right]_{\text{if } i<d} \left[+200 \left(x_i - x_{i-1}^2 \right) \right]_{\text{if } i>1}. \tag{2.4}$$

Below four dimensions this function provably has a single minimum. Between four and seven dimensions it provably has two minima. Above seven there is no analytical proof for minima.

An example of the Rosenbrock surface in two dimensions is presented Figure 2.1a.

Rastrigin

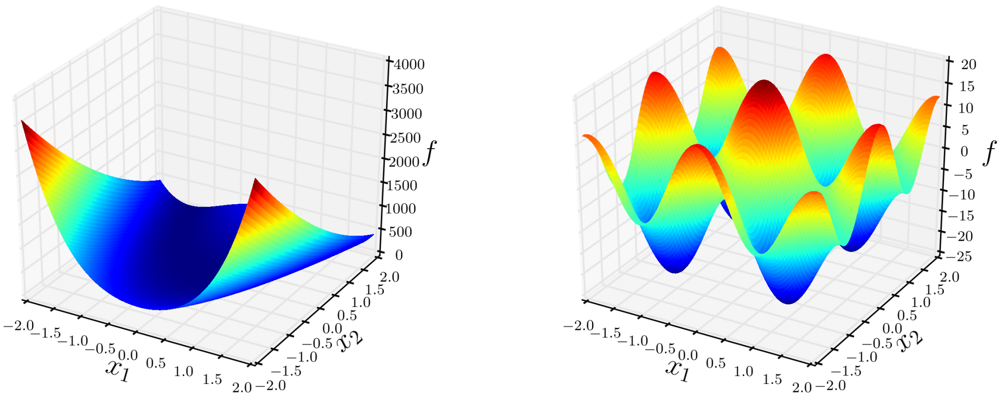
The Rastrigin function is a highly multi-modal function, with small scale variation driven by a cosine term, and large scale variation driven by a parabolic term [59]. This function is modified to serve as a constraint for the analytic SBO experiments. The function as modified is given with [59]:

$$\begin{gathered} f(\mathbf{x}) = \sum_{i=1}^{d} \left[x_i^2 - 10 \cos(\pi x_i) \right], \\ -2.0 < x_i < 2.0, \quad i = \{1, ..., d\}. \end{gathered} \tag{2.5}$$

The gradient is defined with:

$$\frac{\partial f}{\partial x_i} = 2x_i + 10\pi \sin(\pi x_i) \tag{2.6}$$

An example of the Rastrigin surface in two dimensions is presented in Figure 2.1b. It is modified from the original Rastrigin function to have four local minima in the range of $-2.0 < x_i < 2.0$, and to have the mean of the high frequency variation centered around $f = 0$, which will later serve as the constraint threshold.



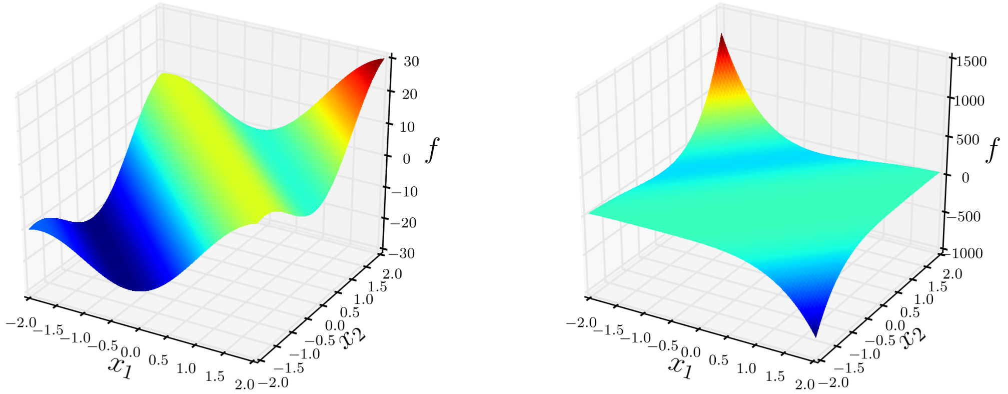
(a) Rosenbrock function in two dimensions.

(b) Modified Rastrigin function in two dimensions.

Figure 2.1: Example test functions in two dimensions.

Rotated Functions

These following two test functions were designed to have one global direction of variability, and one direction of no variability, rotated in the coordinate system.



(a) Visualization of Equation 2.7.

(b) Visualization of Equation 2.10.

Figure 2.2: Relevant geometry for the biparabolic design problem.

The first is a cosine function with a linearly increasing mean, rotated by 45° in the $x_1 - x_2$ plane. The sine behavior provides multiple local minima, necessitating a global optimization algorithm. The first rotated function is defined with:

$$f(\mathbf{x}) = 10 \cos\left(\pi \boldsymbol{A}^\top \mathbf{x}\right) + 10 \boldsymbol{A}^\top \mathbf{x} \tag{2.7}$$

$$\boldsymbol{A} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \tag{2.8}$$

$$-2 < x_i < 2, \quad i = \{1, 2\} \tag{2.9}$$

The second function is a quintic polynomial with three roots, rotated by 135° in the $x_1 - x_2$ plane. The three roots make an equality constraint difficult to find without global optimization. The second rotated function is defined with:

$$f(\mathbf{x}) = 50(\boldsymbol{B}^\top \mathbf{x} + 0.3)^5 - 39(\boldsymbol{B}^\top \mathbf{x} + 0.7)^4 + 50 \tag{2.10}$$

$$\boldsymbol{B} = \begin{bmatrix} -0.5 \\ 0.5 \end{bmatrix} \tag{2.11}$$

$$-2 < x_i < 2, \quad i = \{1, 2\} \tag{2.12}$$

These functions are visualized in Figure 2.2. Notice that there are clear directions with and without variability.

2.2.2 Thin Supersonic Airfoil

There are relatively simple analytic formulae available within supersonic aerodynamics, made available by linear supersonic potential theory. This can be used to find analytic performance results for thin airfoils.

A major result of thin airfoil theory is that the pressure change, and thus the pressure coefficient, on the surface of the airfoil is dependent on the surface gradient. A thorough description and background of the following relationships is provided in Appendix A.1.

The main relationships useful for this study are the analytic relationships for drag coefficient, lift coefficient, and the equivalent area error, on a symmetric two dimensional airfoil of unit chord, constructed of a biparabolic curve with a given thickness ratio τ, angle of attack α.

Under this problem configuration, the lift coefficient is given as:

$$C_l = \frac{4}{\sqrt{M_\infty^2 - 1}}\alpha, \tag{2.13}$$

which states that lift for supersonic thin airfoils, at a constant Mach number, is only dependent on angle of attack. This is in fact general to any linear supersonic potential theory thin airfoil.

The drag coefficient is dependent on the detailed curvature of the airfoil. If the biparabolic airfoil is parameterized by thickness and angle of attack, two common parameters for supersonic wing design, the drag coefficient is given as:

$$C_d = \frac{4\left(\frac{4}{3}\tau^2 + \alpha^2\right)}{\sqrt{M_\infty^2 - 1}} \tag{2.14}$$

which says that drag coefficient is dependent on the square of thickness and the square of angle of attack.

In the context of this biparabolic thin airfoil, the equivalent area functional can also be derived. Under supersonic linear potential theory the flow characteristics are parallel to the Mach angle, so there is no dependence on distance from the airfoil. A target equivalent area is chosen here by the pressure coefficient distribution that corresponds to an airfoil with a given thickness ratio τ_t and angle of attack α_t. The equivalent area functional is subsequently given as,

$$\Delta A_e = \frac{\sqrt{2\beta r}}{675}(44(\tau - \tau_t)^2 + 108(\tau - \tau_t)(\alpha - \alpha_t) + 75(\alpha - \alpha_t)^2), \tag{2.15}$$

which has a second order polynomial relationship on τ and α. The weights of this relationship show that the combination of angle of attack and thickness is the primary effector under the given airfoil parameterization.

2.3 Freeform Deformation

For the simulation based design in this dissertation, deformation of the relevant geometric surfaces is carried out by a separate geometry parametrization using a Free-Form Deformation (FFD) strategy [58]. First, an initial box encapsulating the surface 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

$$X(u, v, w) = \sum_{i,j,k=0}^{l,m,n} P_{i,j,k} B_j^l(u) B_j^m(v) B_k^n(w), \tag{2.16}$$

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 are then transformed into parametric coordinates within the Bézier box.

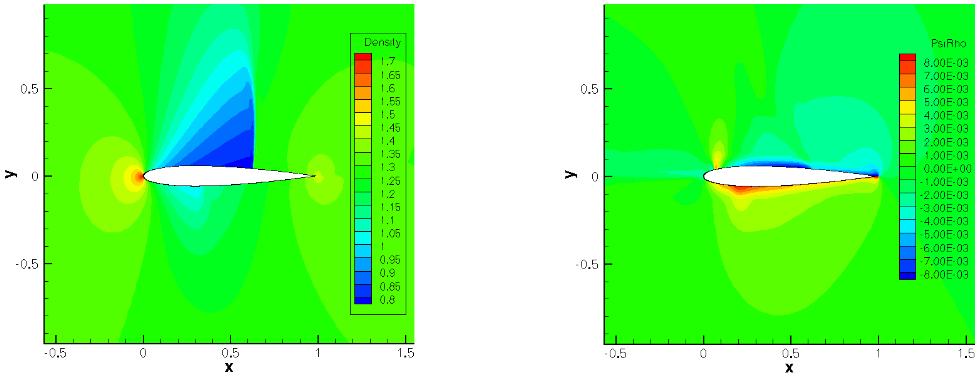
Control points of the box become design variables, as they control the shape of the solid, and thus the shape of the surface grid inside. The box enclosing the geometry is deformed by modifying its control points, with all the points inside the box inheriting a smooth deformation. Arbitrary changes to 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. 2.16.

2.4 Two Dimensional Flow Examples

In two dimensions there are interesting design problems available to exercise the optimization frameworks. These examples walk one category closer to high fidelity analysis. They introduce non-linear flow behaviors, as well as numerical errors that occur in discretized simulations.

2.4.1 NACA 0012 Airfoil

The NACA 0012 is a transonic airfoil. It is included here as a baseline test case of the surrogate modeling approaches in Section 3.3.2, and surrogate based optimization methods in Section 5.3.2.3. The test case here uses a freestream Mach number of 0.80, an angle of attack of 1.25 degrees, and is evaluated inviscidly using the Euler equations. Solutions are converged to residuals on density and energy of less than 1×10^{-13}. An example of the flow solution and adjoint solution for the drag coefficient are shown in Figure 2.3.



(a) Contours of density.

(b) Contours of adjoint density for drag coefficient.

Figure 2.3: Sample solution for direct and adjoint problems of the NACA 0012 airfoil.

The design problem of interest is to minimize drag while constraining lift:

$$
\begin{aligned}
\underset{\mathbf{x}}{\text{minimize}} \quad & C_D(\mathbf{x}) \\
\text{subject to} \quad & \Delta C_L(\mathbf{x}) \geq 1 \times 10^{-5} \\
& -0.03 < x_i < 0.03, \; i \in \{1, 2\}, \\
& \mathbf{x} \in \mathbb{R}^2
\end{aligned}
\tag{2.17}
$$

The airfoil was parameterized with Hicks Henne bump functions. These distribute a series of linearly combined basis functions across the upper and lower surface of the airfoil. An example of two located at the mid-chord of the airfoil are shown in Figure 2.4.

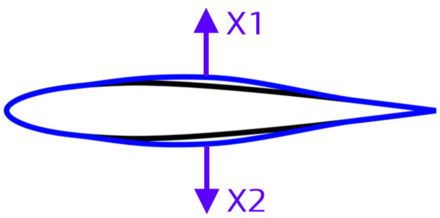


Figure 2.4: Parameterization and example deformation for the NACA 0012.

2.4.2 Biparabolic Airfoil

The biparabolic airfoil is a canonical two-dimensional geometry for supersonic aerodynamics. It is a symmetric airfoil composed of a parabolic arc on the upper and lower surfaces. In this study the airfoil has a chord of unit length, and a thickness of 2% on the chord. The geometry is identified in Figure 2.6a.

This geometry will be driven by the optimization problem shown in Equation 2.18, which attempts to minimize the drag coefficient while constraining equivalent area.

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & C_D(\mathbf{x}) \\ \text{subject to} \quad & \Delta A_e(\mathbf{x}) \leq 1 \times 10^{-5} \\ & -0.03 < x_i < 0.03,\ i \in \{1, ..., 20\}, \\ & \mathbf{x} \in \mathbb{R}^{20} \end{aligned} \tag{2.18}$$

As this is a two dimensional problem, an equivalent area distribution in fact defines the lower surface of the airfoil in a one-to-one mapping, to the approximation of linear supersonic potential theory. Because of this there is not enough freedom to separate the requirements of lift and equivalent area, so lift is omitted from this optimization problem.

There is an intuitively expected result for this problem, which makes validating optimization results more concrete. In this result, the upper surface is flattened so as to reduce wave drag by reducing thickness. The lower surface is unmoved however, so as to respect the equivalent area constraint. As will be shown in the results in Chapter 5, this behavior can be identified using surrogate based optimization.

Changes to the surface of the airfoil are parameterized using freeform deformation control points for a box tightly fitted around the airfoil. Figure 2.5 shows this, as well as an example deformation of one control point. There are twenty control points in total for this problem, which are only permitted to displace up or down in the vertical direction.

The flow domain is drawn to accommodate the objectives of lift and drag coefficient, as well as the equivalent area functional. As shown in Figure 2.6b, the nearfield pressure signature is sampled at two chord-lengths below the airfoil. and the farfield

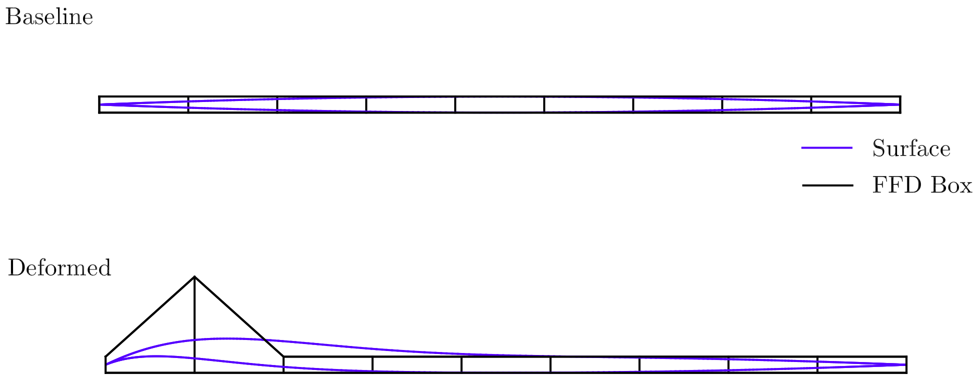
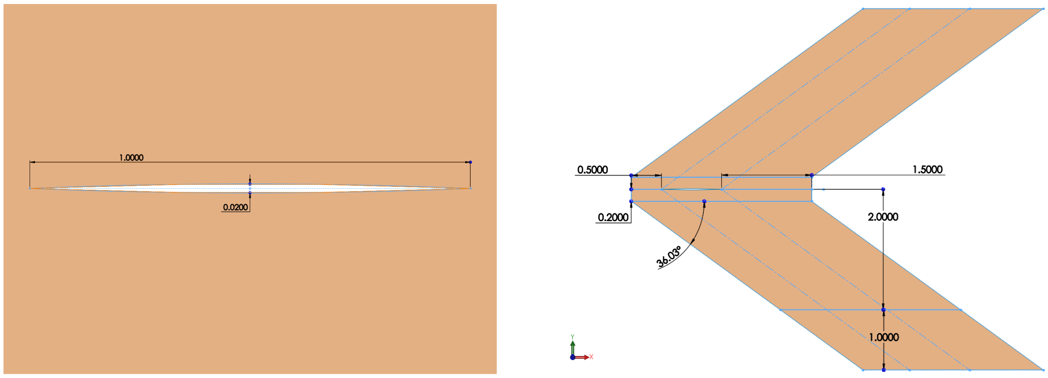


Figure 2.5: Parameterization and example deformation

region extends three chord-lengths below the airfoil. To maintain an efficient mesh, the region immediately around the airfoil is meshed with triangles, and the remainder of the mesh is composed of quadrilaterals that are aligned with the Mach angle of the problem. In total, the mesh presented here is well refined, having 160-thousand elements, with 500 points on the airfoil, and 350 points on the nearfield sampling marker.

The boundary conditions of the problem are a farfield Mach number of 1.7, with an angle of attack of 0.0 degrees. The freestream properties are set for an altitude of 15,240 meters (50,000 feet), and reference pressure, temperature and density are chosen based on the international standard atmospheric model.

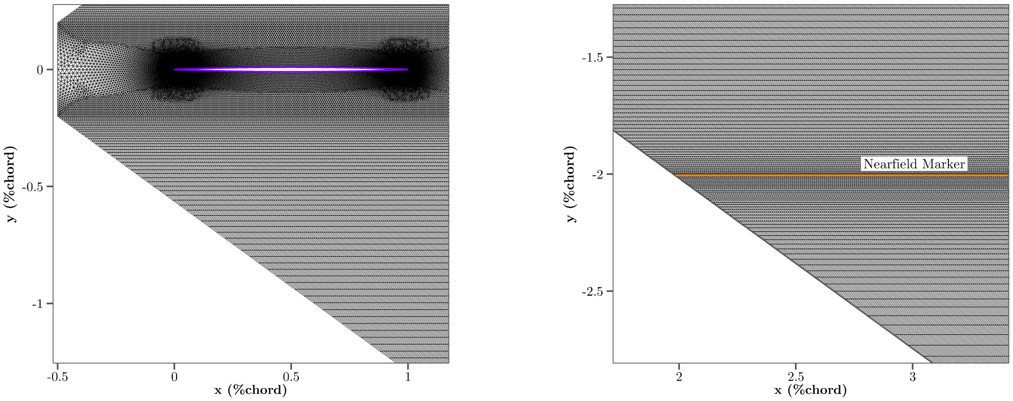
SU2 is used to solve the flow and adjoint problems. Within SU2, the problem is simulated with the Euler equations, which model supersonic flows with reasonable accuracy. The linear solver is FGMRES with LU-SGS preconditioning. The non-linear solver for the flow solution is JST (second order). The non-linear solver for the adjoint problems is Lax-Friedrich (first order). It will be shown in both gradient verification and active subspace results that the first order adjoint solution is sufficient for producing useful gradient results. Residuals are converged to 1×10^{-11} on density and energy.



(a) Airfoil geometry.

(b) Farfield geometry.

Figure 2.6: Relevant geometry for the biparabolic design problem.



(a) Mesh around the airfoil. The airfoil surface is marked in blue.

(b) Mesh around the nearfield marker.

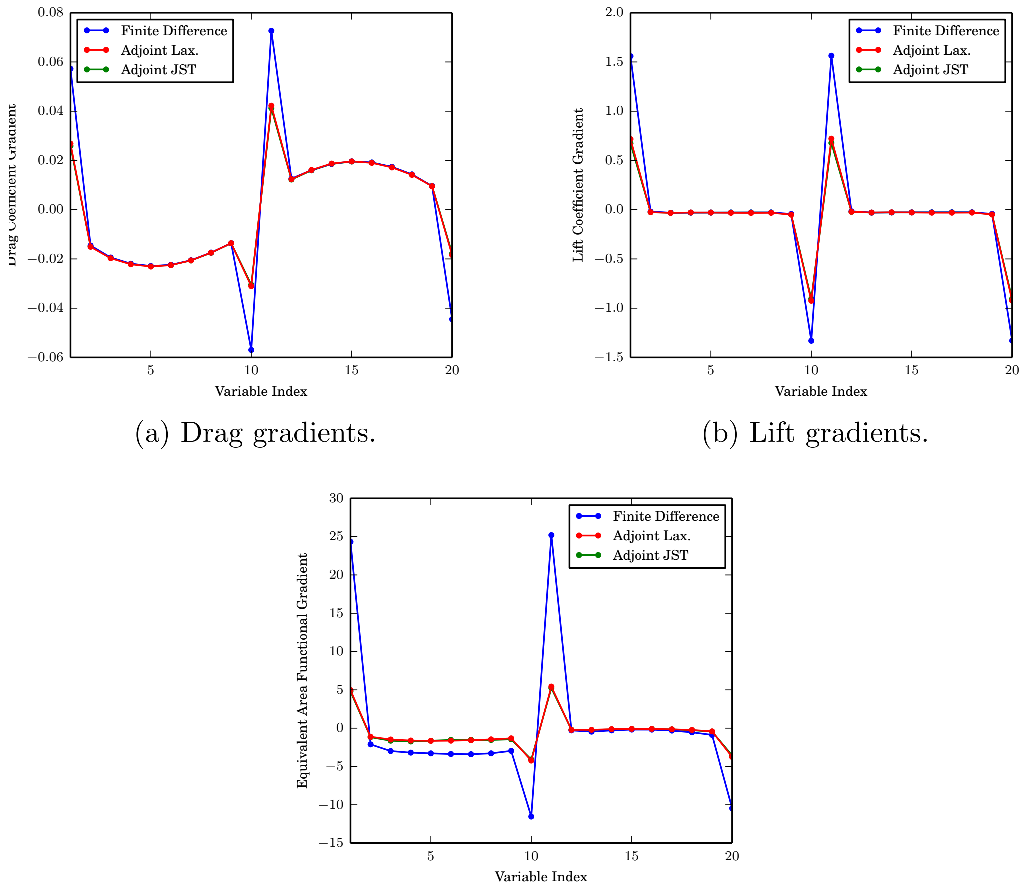
Figure 2.7: Mesh topology for the biparabolic design problem.

An example of the flow and adjoint solutions for the baseline geometry are shown in Figure 2.9. Salient flow features include two strong shocks that emit from the leading and trailing edges. Because this is a non-linear simulation, the incidence of the airfoil surface causes the compression shock to be slightly ahead of the Mach line, and the expansion shock to be slightly behind the Mach line.

The adjoint solution features for lift and drag emanate from the airfoil only, and follow the adjoint characteristics, which are essentially reversed on the flow direction. The equivalent area adjoint features are very much dependent on the shape of the airfoil, but are also tied to the flow at the nearfield sampling marker.

The pressure coefficient and equivalent area along the nearfield marker are shown in Figures 2.9e and 2.9f. As suggested by thin airfoil theory (see Appendix A.1), the constant surface derivative of the airfoil creates a linear pressure slope between two shocks. The mesh resolution and numerical scheme are responsible for the over- and under-shooting artifacts on either side of the discontinuity. Also note that the length of the signature grows by half a chord length by the time it reaches the near field. The equivalent area is shown for a similar region of the nearfield. Because there is no lift generated by this airfoil, if the equivalent area was followed out into well down stream, the trace would approach zero.

The gradients for the baseline geometry of this problem are shown in Figure 2.8. Finite difference steps of 1×10^{-4} chord are used for both the finite differencing of the flow solution, and the finite differencing of the surface when projecting the adjoint surface sensitivities into the design parameterization. The adjoint gradients were simulated using both the Lax-Friedrich (first order) and JST (second order) numerical schemes. These two schemes converged to very similar results, so the faster and more stable Lax-Friedrich was chosen. In general the gradients of finite differenced and adjoint methods are comparible, except near the sharp trailing and leading edges, which are known to be numerically unfavorable because the continuous adjoint formulation assumes a smooth continuous surface.

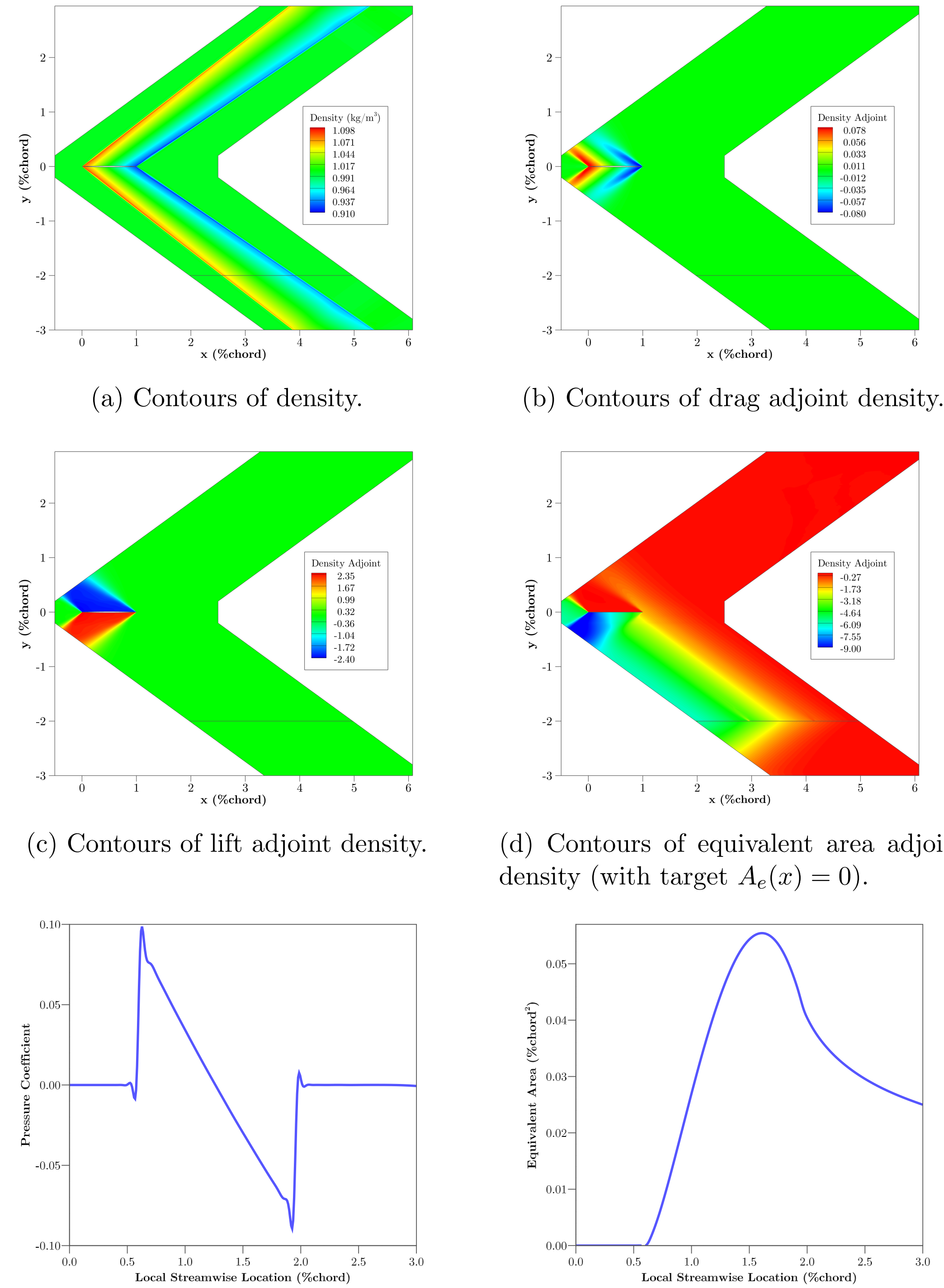


(a) Drag gradients.

(b) Lift gradients.

(c) Equivalent area gradients.

Figure 2.8: Gradient verification for the biparabolic design problem. Finite difference steps were $0.0001c$.



(a) Contours of density.

(b) Contours of drag adjoint density.

(c) Contours of lift adjoint density.

(d) Contours of equivalent area adjoint density (with target $A_e(x) = 0$).

(e) Pressure coefficient at the nearfield.

(f) Equivalent area at the nearfield.

Figure 2.9: Flow and adjoint results for the baseline biparabolic airfoil.

2.5 Three Dimensional Flow Examples

Three dimensional problems bring in additional complexity that can model realistic design problems. For the purpose of this dissertation, these examples are considered high fidelity. This comes at the additional cost of longer simulation times on more compute cores.

2.5.1 ONERA M6 Transonic Wing

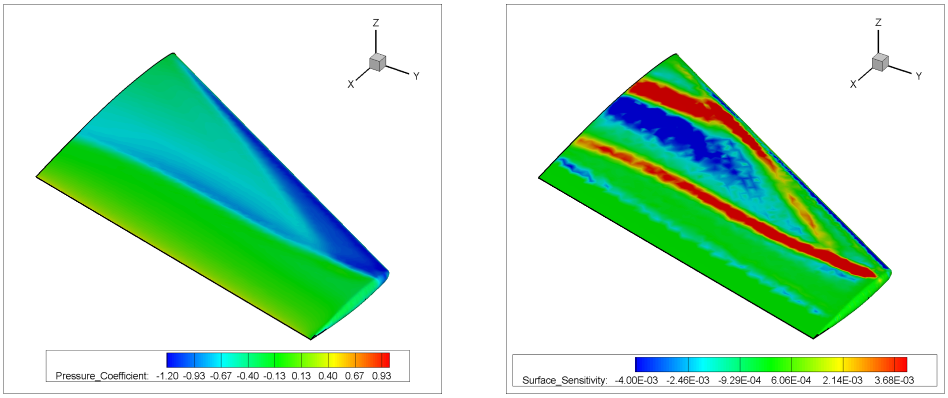
As a test example, work was completed on the optimal shape design of the ONERA-M6 fixed wing, a standard transonic test-geometry. The work is particularly concerned with the problem of maintaining a minimum lift while minimizing the inviscid drag, described in Equation 2.19.

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & C_D(\mathbf{x}) \\ \text{subject to} \quad & C_L(\mathbf{x}) \geq 0.2864 \\ & -0.06 \leq x_i \leq 0.06, \; i \in \{0, ..., 50\}, \\ & \mathbf{x} \in \mathbb{R}^{50} \end{aligned} \tag{2.19}$$

where each variable x_i is constrained within a lower bound $lb_i = -0.06$ and upper bound $ub_i = 0.06$. The bounds were chosen to permit enough variability in the quantities of interest while maintaining enough problem stability to converge most of the random samples, even when adjacent FFD control points move in opposite directions thus introducing sharp changes on the surface.

The flight conditions are a freestream Mach number of 0.8395, at an angle of attack of 3.06 degrees. There are 50 FFD control points in this problem, defined by a single box fit around the whole wing. The trailing edge points are omitted from the problem because the adjoint is not well defined around the sharp trailing edge, which introduces large gradient errors. An example of a deformation of the ONERA-M6 wing is shown in Figure 2.11.

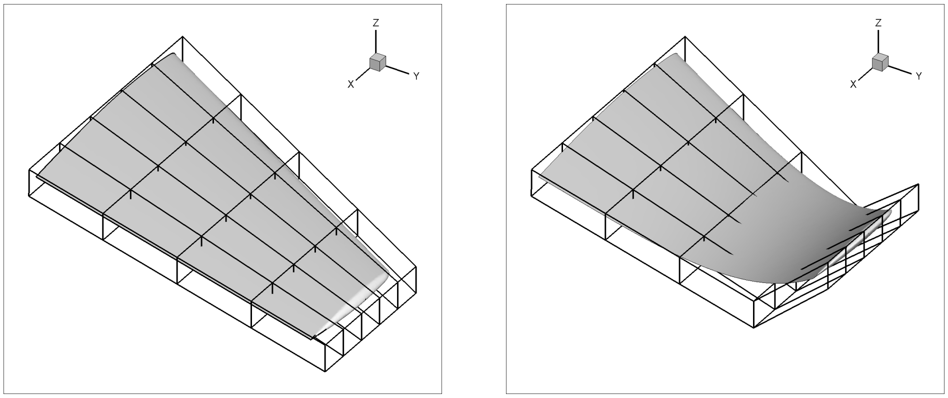
The surface contours in Figure 2.10 shows typical results for the ONERA-M6 wing from SU2. The direct problem (Figure 2.10a) solves for the design objective and constraints, in this case lift and drag. The pressure coefficient contours identify a lambda-shock along the mid-chord of the wing. This a flow feature that strongly contributes to drag, and should be removed by adjusting the wing's shape. The adjoint problem (Figure 2.10b) 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.



(a) Contours of pressure coefficient from a typical direct solution. $C_D = 0.0118$, $C_L = 0.2864$.

(b) Contours of surface sensitivity from a typical drag adjoint solution.

Figure 2.10: Relevant geometry for the biparabolic design problem.



(a) Original surface with FFD box.

(b) Deformed surface and FFD box.

Figure 2.11: An example deformation of the ONERA M6 using freeform deformation boxes.

2.5.2 Langley Supersonic Business Jet

As a major example of this dissertation, the behavior of a simplified supersonic business jet will be studied. Designed by NASA Langley Research Center [60], the geometry is simplified here to only include the fuselage and wing.

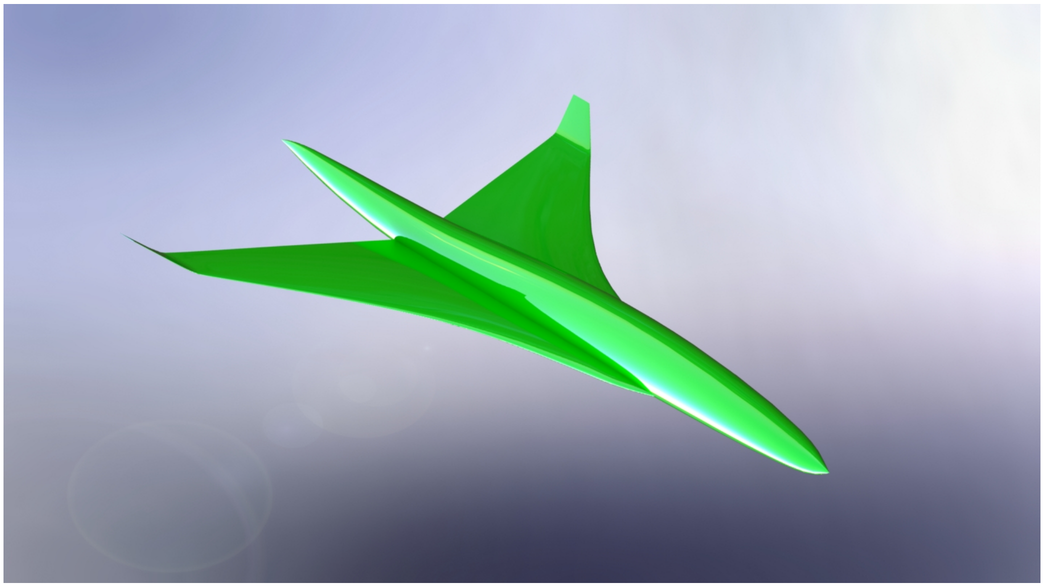


Figure 2.12: Langley Supersonic Business Jet (LSBJ) geometry

The flight conditions are a freestream Mach number of 2.0, at an angle of attack of 2.0 degrees, resulting in a cruise lift coefficient of 0.1. The vehicle has a span of 15.2 meters (50 feet), a root chord of 24.7 meters (81 feet), and a fuselage length of 36.6 meters (120 feet). It is sized to carry seven passengers.

The problem is simulated on a half-body vehicle with a symmetry plane. The results shown with a full body vehicle are mirrored.

For parameterization, there are 198 FFD control points that displace vertically, defined by a single box fit around the whole vehicle. The FFD box is visualized in Figure 2.13. While freestream velocity vector was held constant, this parameterization permits angle of attack changes. This box configuration is rather unusual for a tube and wing problem, since a large number of FFD points have little or no influence over the shape of the vehicle. However, this parameterization will be shown manageable by the active subspace method.

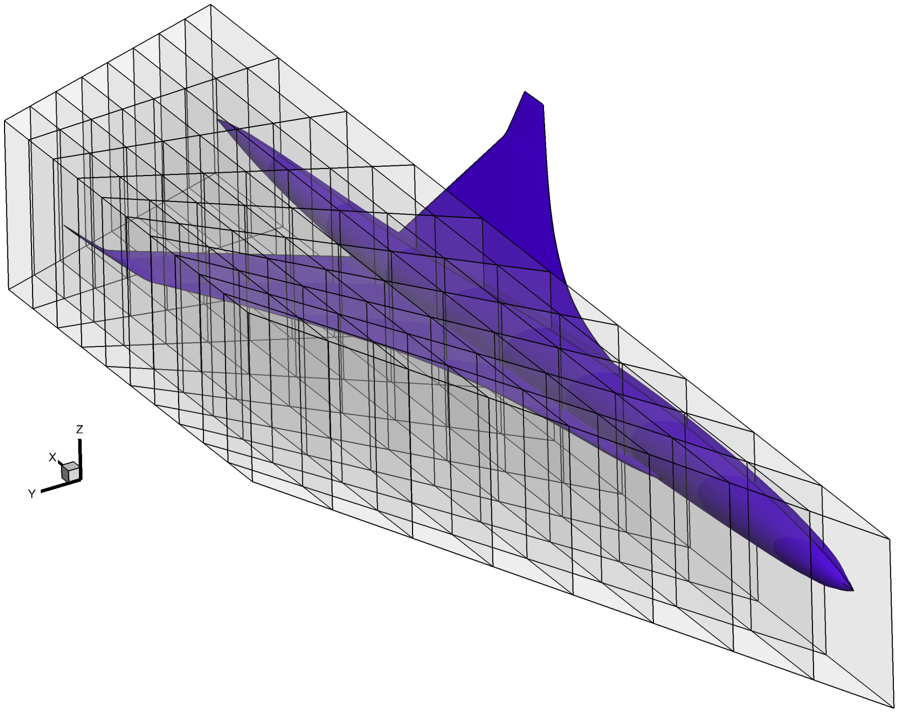


Figure 2.13: Parameterization of the Langley supersonic business jet. The surface is densely parameterized using one FFD box. The taper ratio of the box is restricted by the ability of the FFD algorithm to map the interior points into the internal polynomial basis.

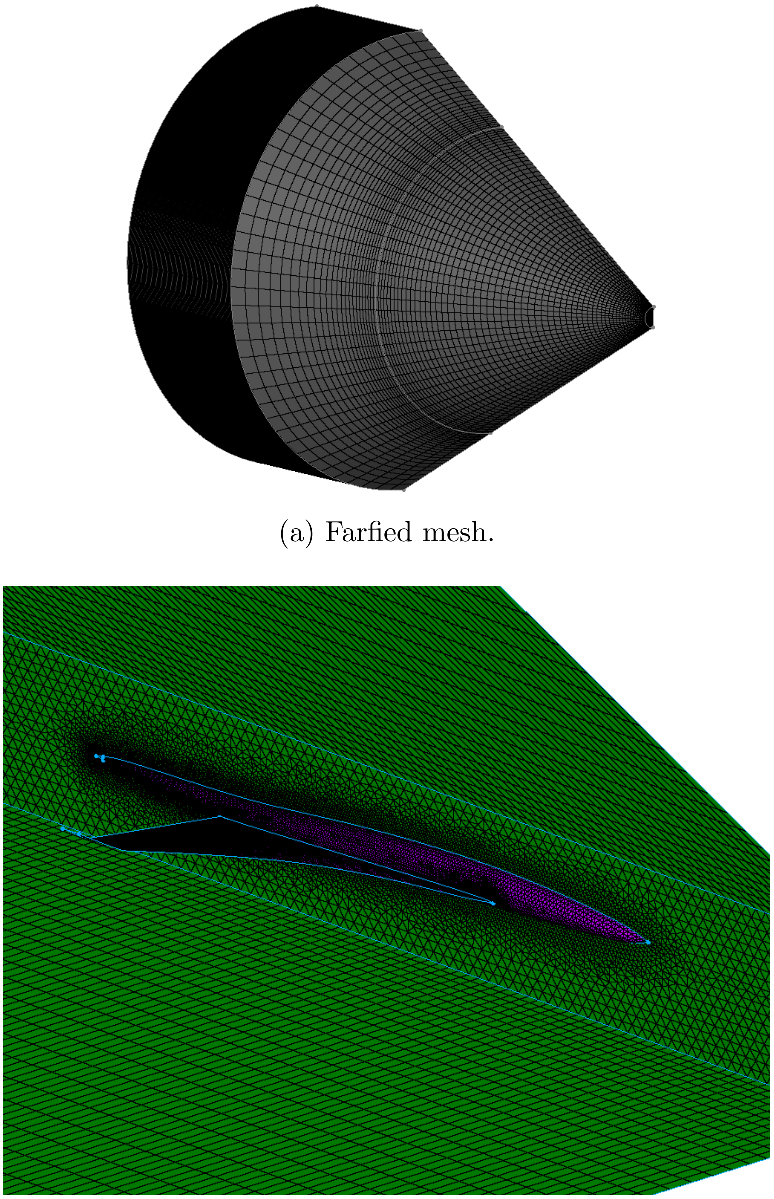
Three quantities of interest are needed here for a useful optimization problem. Drag is an important consideration for fuel burn, lift is important for carrying passengers and cargo, and equivalent area is important for managing boom noise. The optimization problem is listed in Equation 2.20.

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & C_D(\mathbf{x}) \\ \text{subject to} \quad & C_L(\mathbf{x}) \geq 0.10 \\ & \Delta A_e(\mathbf{x}) \leq 1 \times 10^{-4} \\ & -1.0 \leq x_i \leq 1.0,\ i \in \{0, ..., 198\}, \\ & \mathbf{x} \in \mathbb{R}^{198} \end{aligned} \tag{2.20}$$

The bounds of ±1.0 on design variables were chosen to permit enough variability in the quantities of interest while maintaining enough problem stability to converge most of the random samples, even when adjacent FFD control points move in opposite directions thus introducing sharp changes on the surface. The equivalent area target is taken to be that associated with the flow of the baseline design, and as noted in Equation 2.2 it is evaluated over an azimuthal carpet of $\phi = \pm 60°$. The target equivalent area constraint of 1×10^{-4} is a volume in m^3. The volume of the half-body aircraft here is 43.0 m^3. This constraint threshold is chosen to indirectly control the streamwise equivalent area errors. The particular value was found iteratively to yield small changes against the target distribution.

The model is meshed to medium refinement with 1.3 million cells, with unstructured tetrahedra and pyramids around the aircraft, and structured hexes in the region in which the aircraft's pressure signature propagates to the nearfield boundary. This boundary is located at a cylinder two body lengths radially from the vehicle's moment reference center and with an axis parallel to the freestream flow. The hexes are aligned with the Mach cones of the flow. Sections of the mesh are shown for reference in the Figure 2.14.

The surface contours in Figure 2.15 shows typical results for the LSBJ from SU2. A similar solver configuration is used as the biparabolic airfoil, except results are converged to 1×10^{-8}. The direct flow solution is presented in Figure 2.15a. The

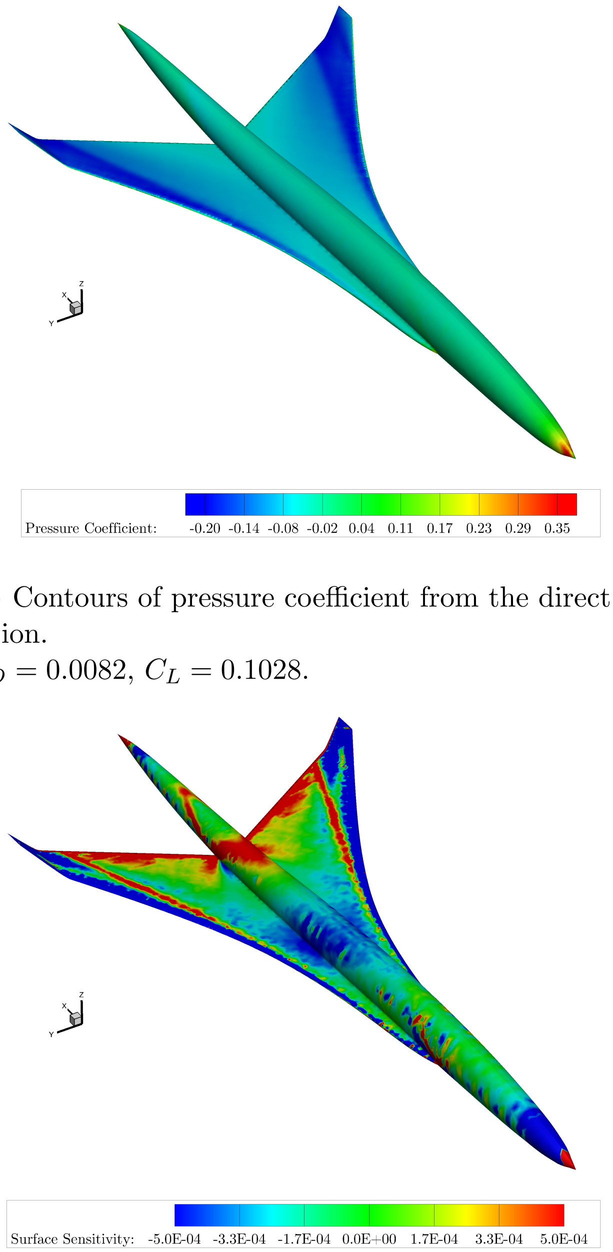


(a) Farfied mesh.

(b) Aircraft and symmetry plane mesh.

Figure 2.14: Selected views of the LSBJ mesh. The mesh contains 1.3 million cells.

pressure coefficient contours identify a shock system that emits from the knee of the cranked-delta wing. The adjoint solution for drag is presented in Figure 2.15b. The surface sensitivity contours identify locations that can be deformed to minimize drag.



(a) Contours of pressure coefficient from the direct solution.
$C_D = 0.0082$, $C_L = 0.1028$.

(b) Contours of surface sensitivity from the drag adjoint solution.

Figure 2.15: Example of mirrored surface solutions for the Langley SBJ.

2.5.3 N+2 Supersonic Passenger Jet

This design problem represents a complex geometry representative of a large supersonic passenger jet capable of carrying 70 passengers. The N+2 aircraft is meant to be the next generation of supersonic civil transport, and this particular design has been through extensive design studies through work with NASA, Lockheed, General Electric, and Stanford University.

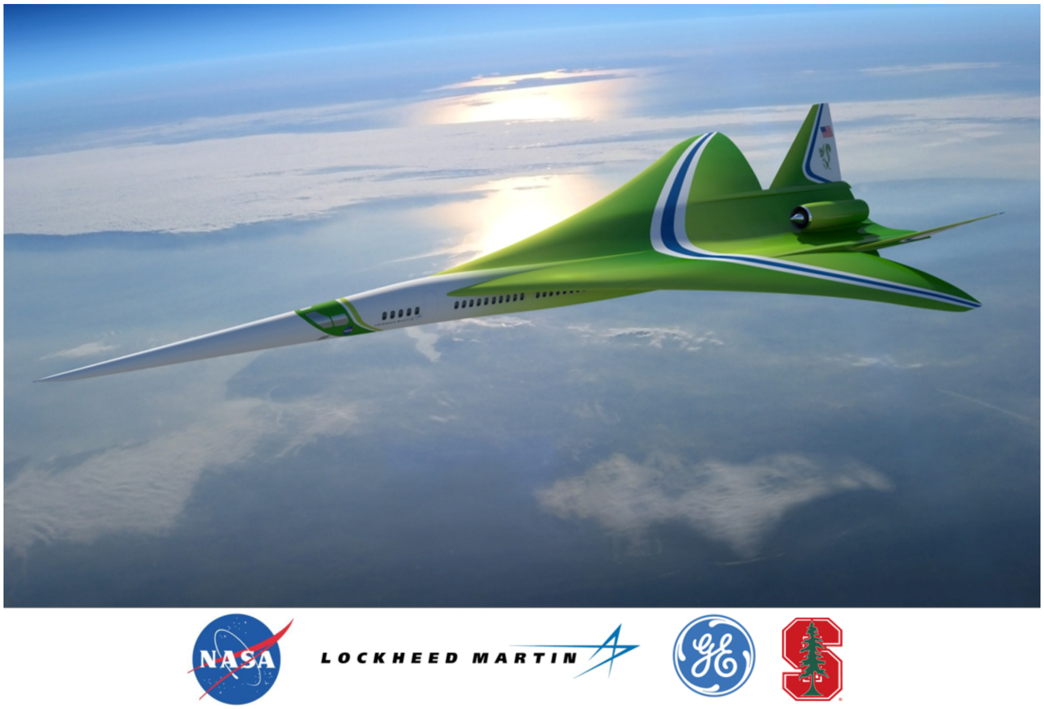


Figure 2.16: N+2 Supersonic Passengjer Jet (SPJ) geometry

The flight conditions are a freestream Mach number of 1.7, at an angle of attack of 2.1 degrees, resulting in a baseline cruise lift coefficient of 0.14. Flow simulations were performed with SU2.

For this dissertation, various studies were performed on the Lockheed N+2 aircraft. The primary optimization of interest here is similar to the Langley SBJ, to minimize drag while constraining equivalent area and lift. The problem is given in

Equation 2.21.

$$
\begin{aligned}
\underset{\mathbf{x}}{\text{minimize}} \quad & C_D(\mathbf{x}) \\
\text{subject to} \quad & C_L(\mathbf{x}) \geq 0.14 \\
& \Delta A_e(\mathbf{x}) \leq 1 \times 10^{-4} \\
& -3.0 \leq x_i \leq 3.0,\ i \in \{0, ..., D\}, \\
& \mathbf{x} \in \mathbb{R}^D
\end{aligned}
\tag{2.21}
$$

Two different configurations of deformation parameters and engine geometry were used in the course of this thesis. The first is simplified to be used only for drag minimization without engine effects. The second is fully complex as it samples the nearfield equivalent area and models engine effects. In both cases only components of the vehicle like the wing, tail and fuselage are modified, contrasting the full parameterization of the Langley Business Jet shown in the previous section. As a result, changes in vehicle angle of attack were not available in this design case.

2.5.3.1 Drag Mesh, Flow-Through Engines

The first set of studies used a geometry with flow-through engine nacelles, shown in Figure 2.17. In this configuration the engines do not generate thrust, which would otherwise affect the vehicle's pressure signature.

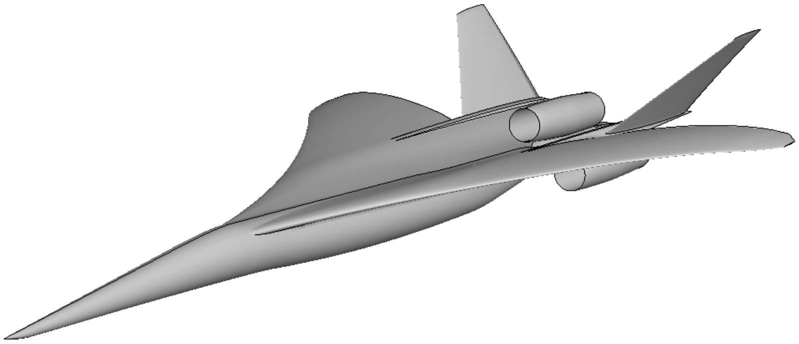
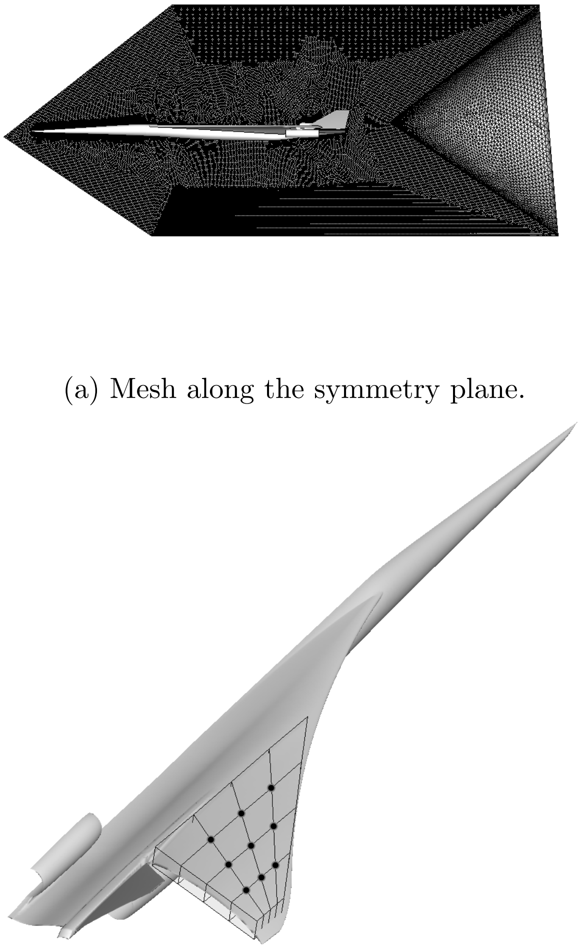


Figure 2.17: Baseline N+2 Supersonic Passenger Jet Geometry

This geometry was used to run an unconstrained drag minimization. A fine mesh of 4.3 million nodes without a nearfield region was used, shown in Figure 2.18a. There were only nine free-form deformation design variables spread across the top surface of the wing, shown Figure 2.18. This small parameterization was chosen to permit a traditional surrogate based optimization.



(a) Mesh along the symmetry plane.

(b) FFD control points, there are nine, marked with filled circles.

Figure 2.18: Mesh and parameterization for an N+2 drag minimization

2.5.3.2 Boom Mesh, Active Engines

The second configuration of this problem uses a geometry that includes the nearfield region, active engines and is parametrized with 105 freeform design variables. An example of the flow solution is presented in Figure 2.19, and the parameterization is presented in Figure 2.20.

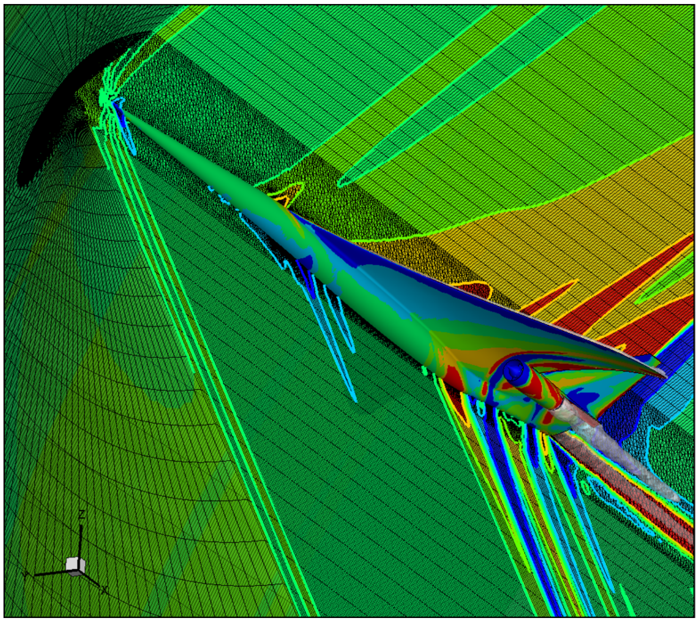


Figure 2.19: A visualization of the flow solution and mesh for the N+2 Supersonic Passengjer Jet (SPJ) geometry, including engines. The problem is simulated as a half-body.

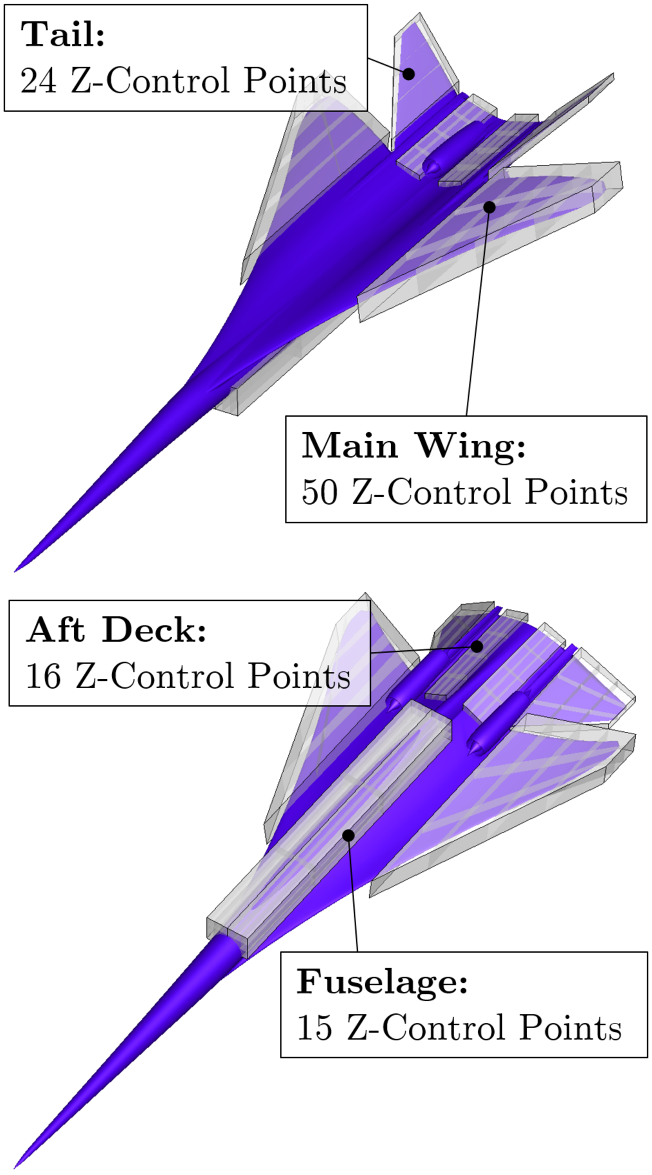


Figure 2.20: Control points for the N+2 supersonic passenger jet problem. Features are mirrored.

Chapter 3

Surrogate Modeling

This chapter will describe and demonstrate an approach to Gaussian Process Regression (GPR) built for surrogate based optimization. In general, surrogate modeling is concerned with predicting the response of an output quantity of interest at a particular location within the input parameterization, given a set of samples of the inputs and outputs called training data.

GPR is as super-set of Kriging. It approaches regression from a Bayesian standpoint by conditioning a probabilistic function to training data[61]. 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 [29], or inaccurate gradients as in the present work. This was one of the key motivations for this dissertation's exploration of Gaussian Process Regression.

3.1 Background

Surrogate modeling has a long and colored history. A classic approach to approximating experimental data is least-squares polynomial regression. In the context of optimization, using polynomial regression to generate a surrogate model for optimization was often called “Response Surface Modeling”, or the “Response Surface Method” [62, 63, 53]. Polynomial regression is part of a larger classification of “parametric” models, which have a fixed number of parameters that must be chosen, typically by minimizing the squared error of the model to the data. These techniques have a beneficial property of smoothing noisy data, but at the sacrifice of making strong assumptions about the shape of the data. This can prevent the model from identifying salient features, such as multiple local minima that can appear in complex design problems.

A landmark transition occurred from parametric to non-parametric probabilistic modeling (like GPR) when Sacks identified the usefulness of uncertainty information to refine response surfaces of expensive simulations [48]. Non-parametric methods make very general assumptions about the behavior of the fit, but effectively use the experimental data as parameters, yielding a valuable amount of generality when modeling new problems. Probabilistic regression methods can furthermore provide an estimated variance, or measure of uncertainty, as a function of the inputs. As will be seen in the discussion of efficient global optimization, this information can be used to identify areas of the model that are inaccurate and could be re-sampled.

An extraordinary number of useful non-parametric regression methods exist. Major approaches include splines, neural networks, and Kriging.

Splines are piecewise polynomial functions that exhibit a high degree of smoothness [64]. They are powerful interpolators in one or two dimensions, but are difficult to regress when using more than three dimensions.

Neural networks model responses using a composition of interconnected nodes, where input nodes are mapped through a set of “hidden layer nodes” that each apply a weighted “activation function”, and yield values on a set of output nodes. Neural networks are especially useful in classification problems, but are applicable to regression especially when the nodes employ a Radial Basis Function (RBF). Neural networks have seen use in optimization [65].

In the case that one extends neural networks to include an infinite number of RBF hidden layer nodes, it can be found that a class of interpolator called “Gaussian

Process Regression" (GPR) is available [61]. GPR is a probabilistic predictor derived using Bayesian statistics to condition a spatially varying mean and variance based on the training data. This posterior mean is used as an approximation of the data, and the posterior variance as an approximation of the uncertainty of the fit at a particular location. A nearly identical formulation can be arrived from a "frequentist" view by minimizing the square error of a predictor given spatially correlated samples, in a derivation proposed by Matheron which he called Kriging [66], named after his mentor.

As an approximation of an expensive function, surrogate models can be interrogated with an optimizer in what is known as Surrogate-Based Optimization (SBO) as a way to minimize expensive evaluations. Studies using surrogate modeling approaches grew substantially in the 1990's, especially in the area of Multi-Disciplinary Optimization of NASA's High Speed Civil Transport [67, 68, 69]. This resulted in a Surrogate Modeling Framework formalizing the approach [70]. Additional work was done in the 2000's in this area under the context of aircraft [71, 19, 72, 26, 29].

As an approach for increasing the accuracy of surrogates in higher dimensions, previous literature has explored using gradient information to enhance Gaussian Process Regression surrogate models [53, 29, 41, 73]. Multiple fidelity information has also been used to enhance the accuracy of the fit while displacing the need for expensive simulations to numerous low-fidelity simulations [74, 71, 41, 75].

Several difficulties with GPR occur in practice and have been a topic of research in the community. These include noisy data that can corrupt the response surface [76, 77, 78], the tendency for the surrogate model to become numerically unstable [79, 80], and most notoriously is the curse of dimensionality, which expresses the difficulty surrogate models have regressing data that becomes sparse in high dimension [81].

3.2 Method Formulation

A derivation of Gaussian Process Regression will now be presented. Though presented in numerous works, it is reproduced and extended here to (1) motivate the use of noise models to manage gradient inaccuracies, (2) identify a new approach to condition the hyperparameter tuning problem with *a-priori* constraints, and (3) unify notation to accomplish the first two and integrate with the use of the active subspace method in Chapter 4.

3.2.1 Bayesian Foundation

Following the derivation given by Rasmussen [61], Gaussian Process Regression is approached by conditioning a multivariate normal distribution.

$$f \sim \mathcal{N}\left(\mu, [\Sigma]\right), \tag{3.1}$$

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

For this work, take a uniformly zero mean vector, and populate the standard deviation with a covariance matrix composed of sub-matrices $k(\cdot,\cdot)$ that are a function of training and estimated data:

$$\begin{gathered} \begin{bmatrix} f_p \\ f_i^* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} k(\mathbf{x}_p, \mathbf{x}_q) & k(\mathbf{x}_p, \mathbf{x}_j^*) \\ k(\mathbf{x}_i^*, \mathbf{x}_q) & k(\mathbf{x}_i^*, \mathbf{x}_j^*) \end{bmatrix}\right), \\ \{\, f_t(\mathbf{x}_t) \,|\, t = 1, ..., M \,\} \ , \ \{\, f_r^*(\mathbf{x}_r^*) \,|\, r = 1, ..., N \,\}. \end{gathered} \tag{3.2}$$

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 $k(\mathbf{x}_p, \mathbf{x}_q)$ would be equivalent to the matrix $\boldsymbol{K} = k_{p,q}$. There are M training point horizontal vectors $\mathbf{x}$, with function values $f(\mathbf{x})$, and N estimated data point horizontal vectors $\mathbf{x}^*$ with function values, $f^*(\mathbf{x}^*)$.

Of the data, the estimated function values f^* are unknown. The training data with locations $\mathbf{x}$ and function values $f(\mathbf{x})$ are known, as are the desired estimated data locations $\mathbf{x}^*$. Following Rasmussen's derivation [61], GPR is developed by conditioning the normal distribution with the known data:

$$f|\mathbf{x}^*, \mathbf{x}, f \sim \mathcal{N}\left(f^*, \mathbb{V}[f^*]\right), \tag{3.3}$$

which allows the identification of useful relations for estimating a function fit,

$$\begin{aligned} f_i^* &= k(\mathbf{x}_i^*, \mathbf{x}_q)\, k(\mathbf{x}_p, \mathbf{x}_q)^{-1}\, f_p \\ s_{i,j}^* &= k(\mathbf{x}_i^*, \mathbf{x}_j^*) \;-\; k(\mathbf{x}_i^*, \mathbf{x}_q)\, k(\mathbf{x}_p, \mathbf{x}_q)^{-1}\, k(\mathbf{x}_p, \mathbf{x}_j^*) \end{aligned} \tag{3.4}$$

where the diagonal element $s_{i,i}^*$ is the estimated covariance of the estimated value f_i^*. These are the relations needed for coding a GPR program. Rasmussen provides an example algorithm that simplifies these relations by using Cholesky decomposition [61].

The choice of a uniform mean vector in Equation 3.2 brings in an important assumption of *stationarity*, which says that on average the fit is expected to be a flat function, with deviations above and below. Often in this class of surrogate modeling one chooses to only look at a windowed view of the regression. In this view it is possible to model functions that are are non-stationary, for example one that is constantly increasing. But if this model were viewed in a much larger domain, zoomed out from all of the data, the regression surface would be seen to equilibrate at zero far away from the data. It is possible to regress the mean value of the data [61], however in the practice of this dissertation it was not found necessary because the output range of the data is normalized. There are additional methods for regressing non-stationary mean models [61, 50], but this adds significant complexity when including gradient information so is avoided in this dissertation.

3.2.2 Covariance Function

The covariance function (also known as a kernel 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 that would require different choices of covariance functions. A common covariance function is the Gaussian function of the Euclidean distance between points:

$$\mathrm{Cov}\left[F(\mathbf{x}_p),\ F(\mathbf{x}_q)\right] = k(\mathbf{x}_p, \mathbf{x}_q) = \theta_1^2 \exp\left(-\frac{1}{2\theta_2^2}\sum_{v=1}^{D}(x_{p,v} - x_{q,v})^2\right), \tag{3.5}$$

where F is the random variable of f, D is the number of dimensions, and $\mathbf{x}_p$ and $\mathbf{x}_q$ are two position vectors chosen from the design space $\mathcal{X}$. There are two degrees of freedom in this covariance function. These are known as hyperparameters. 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.

While traditionally this kernel includes a length scale parameter for each dimension, the above function includes only one length scale parameter for all dimensions. This makes an assumption of *isotropy*: that on average the variation of the function is the same in each input direction. Most of the design problems in this dissertation exhibit such a condition because of the ubiquitous use of freeform deformation box parameterization. The assumption would breakdown for example if a subset of variables exhibits much higher frequency behavior then another subset. Under a reasonably dense sample of training data, the surrogate will still struggle to model the function along lower-frequency directions. In this case the hyperparameters typically favor tuning for the high-frequency behavior, and results in potentially large deviations in what should be flat regions of the model. This is in fact encountered in the design problem of the biparabolic airfoil presented in Chapter 5, and will be addressed by introducing active subspace dimensionality reduction.

Choosing the covariance function is an important choice when applying a GPR surrogate as it expresses a strong statement about the general behavior of the fit. In general these functions must be shown to have the property of being positive semi-definite. The Gaussian kernel is favored in many aerospace optimization problems where the functions are in general smooth and change gradually. Other common kernels include the exponential kernel or the Matérn kernel, which can be chosen to tune the surrogate for sharper changes [61, 53]. Polynomial radial basis covariance functions have been applied in similar work as well [41].

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[82] and Koehler [53] by deriving the covariance functions from the definitions of variance and derivative. Another approach suggested by Papoulis[83] and used for gradient enhanced GPR by Solak[84] exploits the theorem that the linear operation of an expected value is the expected value of the linear operation. In either case, the main result is to simply take the derivatives of the covariance function in order to include gradient information in the fit:

$$\begin{array}{rcccc} \mathrm{Cov}\left[\frac{\partial F(\mathbf{x}_p)}{\partial x_v},\, F(\mathbf{x}_q)\right] & = & k\left(\frac{\partial \mathbf{x}_p}{\partial x_v}, \mathbf{x}_q\right) & = & \frac{\partial k(\mathbf{x}_p,\mathbf{x}_q)}{\partial x_{p,v}} \\ \mathrm{Cov}\left[F(\mathbf{x}_p),\, \frac{\partial F(\mathbf{x}_q)}{\partial x_w}\right] & = & k\left(\mathbf{x}_p, \frac{\partial \mathbf{x}_q}{\partial x_w}\right) & = & \frac{\partial k(\mathbf{x}_p,\mathbf{x}_q)}{\partial x_{q,w}} \\ \mathrm{Cov}\left[\frac{\partial F(\mathbf{x}_p)}{\partial x_v},\, \frac{\partial F(\mathbf{x}_q)}{\partial x_w}\right] & = & k\left(\frac{\partial \mathbf{x}_p}{\partial x_v}, \frac{\partial \mathbf{x}_q}{\partial x_w}\right) & = & \frac{\partial}{\partial x_{q,w}}\left(\frac{\partial k(\mathbf{x}_p,\mathbf{x}_q)}{\partial x_{p,v}}\right) \end{array} \tag{3.6}$$

This is where a strong assumption is made on the correlation model between function value and its gradient. It is a natural and powerful assumption. However, because there will be D-times more gradient information than function values, inaccurate gradients can overpower the behavior of the fit and in the process ignore the information from the function values.

The notation above shows that the covariance between a derivative and a function value is found by taking the derivative of the covariance matrix between two function values. The notation $k\left(\frac{\partial \mathbf{x}_p}{\partial x_v}, \mathbf{x}_q\right)$ is short hand for covariance $\mathrm{Cov}\left[\frac{\partial F(\mathbf{x}_p)}{\partial x_v},\, F(\mathbf{x}_q)\right]$. The derivative of the kernel $\frac{\partial k(\mathbf{x}_p,\mathbf{x}_q)}{\partial x_{p,v}}$ requires a special attention, where the derivative is taken with respect to only one input vector of the kernel, and further with respect to only one component of the input space $\mathcal{X}$. This attention to the domain of the derivative is similarly required when interpreting the complementing function-derivative kernel. In the case of the derivative-derivative kernel the chain rule applies when working out the derivatives. In this dissertation these derivatives are non-zero since Gaussian kernels are infinitely differentiable.

It turns out for radial basis kernels that the function-derivative kernel is related to the negative of the derivative-function kernel,

$$k\left(\frac{\partial \mathbf{x}_p}{\partial x_v}, \mathbf{x}_q\right) = -k\left(\mathbf{x}_p, \frac{\partial \mathbf{x}_q}{\partial x_w}\right), \tag{3.7}$$

which can be helpful when programming these methods efficiently.

The detail behind the result of Equation 3.6 is presented here, following the approach shown by Koehler [53] and translating it into the present notation. First, take the definition for covariance between a gradient and function,

$$\operatorname{Cov}\left[\frac{\partial F(\mathbf{x}_p)}{\partial x_v},\, F(\mathbf{x}_q)\right] = \operatorname{E}\left[\frac{\partial F(\mathbf{x}_p)}{\partial x_v}\, F(\mathbf{x}_q)\right] - \operatorname{E}\left[\frac{\partial F(\mathbf{x}_p)}{\partial x_v}\right] \operatorname{E}\Big[F(\mathbf{x}_q)\Big]. \tag{3.8}$$

Under the assumption of stationarity introduced with the zero mean vector in Equation 3.2, the expected value of the gradient is zero. Imagine zooming far out from the function to see the assumption of a zero mean far from the data. Thus Equation 3.8 simplifies,

$$\operatorname{Cov}\left[\frac{\partial F(\mathbf{x}_p)}{\partial x_v},\, F(\mathbf{x}_q)\right] = \operatorname{E}\left[\frac{\partial F(\mathbf{x}_p)}{\partial x_v}\, F(\mathbf{x}_q)\right]. \tag{3.9}$$

Now apply the limit theorem to the gradient and rearrange,

$$\begin{aligned}
\mathrm{Cov}\left[\frac{\partial F(\mathbf{x}_p)}{\partial x_v},\, F(\mathbf{x}_q)\right] &= \mathrm{E}\left[\lim_{\Delta x_v \to 0} \frac{F(\mathbf{x}_p + \Delta x_v) - F(\mathbf{x}_p)}{\Delta x_v} F(\mathbf{x}_q)\right] \\
&= \mathrm{E}\left[\lim_{\Delta x_v \to 0} \frac{F(\mathbf{x}_p + \Delta x_v)F(\mathbf{x}_q) - F(\mathbf{x}_p)F(\mathbf{x}_q)}{\Delta x_v}\right] \\
&= \lim_{\Delta x_v \to 0} \frac{\mathrm{E}\left[F(\mathbf{x}_p + \Delta x_v)F(\mathbf{x}_q)\right] - \mathrm{E}\left[F(\mathbf{x}_p)F(\mathbf{x}_q)\right]}{\Delta x_v} \\
&= \lim_{\Delta x_v \to 0} \frac{\begin{array}{c}\mathrm{E}\left[F(\mathbf{x}_p + \Delta x_v)F(\mathbf{x}_q)\right] - \mathrm{E}\left[F(\mathbf{x}_p)F(\mathbf{x}_q)\right] \\ -\mathrm{E}\left[F(\mathbf{x}_p + \Delta x_v)\right]\mathrm{E}\left[F(\mathbf{x}_q)\right] + \mathrm{E}\left[F(\mathbf{x}_p)\right]\mathrm{E}\left[F(\mathbf{x}_q)\right]\end{array}}{\Delta x_v} \\
&= \lim_{\Delta x_v \to 0} \frac{\mathrm{Cov}\left[F(\mathbf{x}_p + \Delta x_v),\, F(\mathbf{x}_q)\right] - \mathrm{Cov}\left[F(\mathbf{x}_p),\, F(\mathbf{x}_q)\right]}{\Delta x_v} \\
&= \tfrac{\partial}{\partial x_v}\mathrm{Cov}\left[F(\mathbf{x}_p),\, F(\mathbf{x}_q)\right].
\end{aligned} \tag{3.10}$$

In this manipulation various identities for covariance and expectation are used, as well as the assumption of stationarity which is needed to expand the numerator in the fourth step.

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

$$f_p \rightarrow \begin{bmatrix} f_p \\ \frac{\partial f_p}{\partial x_v} \end{bmatrix} \tag{3.11}$$

$$k(\mathbf{x}_p, \mathbf{x}_q) \rightarrow \begin{bmatrix} k(\mathbf{x}_p, \mathbf{x}_q) & k\left(\mathbf{x}_p, \frac{\partial \mathbf{x}_q}{\partial x_w}\right) \\ k\left(\frac{\partial \mathbf{x}_p}{\partial x_v}, \mathbf{x}_q\right) & k\left(\frac{\partial \mathbf{x}_p}{\partial x_v}, \frac{\partial \mathbf{x}_q}{\partial x_w}\right) \end{bmatrix}, \tag{3.12}$$

where the gradient vector $\frac{\partial f_i}{\partial x_v}$ has dimension $M \cdot D \times 1$, sub-matrix $k(\mathbf{x}_p, \mathbf{x}_q)$ has dimension $M \times M$, and sub-matrix $k\left(\frac{\partial \mathbf{x}_p}{\partial x_v}, \frac{\partial \mathbf{x}_q}{\partial x_w}\right)$ has dimension $M \cdot D \times M \cdot D$. Each of the above updates must be patterned onto each element of Equation 3.2, and by extension Equation 3.3. Specifically, Equation 3.11 will update both f_p to include training gradient outputs, and f_i^* to predict training gradient outputs; and Equation 3.12 will update all four $k(\cdot,\cdot)$ kernel matrices, corresponding to the locations of the training and predicted gradients respectively. Note that the gradient data and function data need not be co-located, which offers interesting directions for future work. However in the present work adjoint simulations require a direct simulation so both pieces of information are collected in process.

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 (all the kernels and data with gradients and without $(\cdot)^*$) , but keeping those associated with the estimated data gradients (all the kernels and data *with* $(\cdot)^*$). 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.

Noise Models

When working with numerical experiments such as CFD simulations, simulations are deterministic and repeatable. However, they are built as finite approximations using fixed-precision operations. As a result, under small changes in inputs, there can be small non-smooth changes in objectives. If this is not accounted for, especially when a large amount of gradient information is present, then the accuracy of the fit suffers. The strategy in this dissertation is to model these small scale high-frequency variations as noise.

Several types of noise can be modeled within the GPR framework. Allowing noise can relax the assumption of exact correlation model between objective and gradient information. The effect on the response surface will have the form:

$$f_\epsilon^*(\mathbf{x}) = f^*(\mathbf{x}) + \epsilon, \tag{3.13}$$

where ϵ is a noise model. As explored by Moré, by inspecting the variation of an approximated function in a neighborhood of small perturbations, small local perturbations can be interpreted and analyzed as noise, despite being a stretch of the formal non-deterministic definition [85]. As will be shown in the numerical experiments section of this chapter, this sort of variation is present in aerospace design problems involving CFD simulations.

Adding noise to the model requires an update to the covariance matrix structure:

$$k(\mathbf{x}_p, \mathbf{x}_q) \rightarrow k(\mathbf{x}_p, \mathbf{x}_q) + k_\epsilon, \tag{3.14}$$

where k_ϵ 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[61]. This will only affect the self-correlated covariance terms along the diagonal of $[k_\epsilon]$. The noise covariance matrix will then take the form:

$$k_\epsilon = \begin{bmatrix} \theta_3^2 \, I_{a,a} & 0_{a,b} \\ 0_{b,a} & \theta_4^2 \, I_{b,b} \end{bmatrix}, \tag{3.15}$$

where I and 0 are identity and zero matrices with sizes $a = M$ and $b = M(1 + D)$. 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 hyperparameter, the fit will be allowed to stray a certain distance away from the data. *This will allow the model to account for numerical error in the gradients due to inaccuracies from mesh refinement or the particular sensitivity method.*

Note that there are two separate hyperparameters for the noise of the function values θ_3 and the noise of the gradients θ_4. As will be shown, this enables an important amount of control in the learning process, for example to encourage the fit to respect the function values before the gradients. Additionally, the presence of a single θ_4 hyperparameter for the gradients is a further extension of this derivation's assumption of isotropy. In this case a claim is made that on average the noise of each gradient component (which could come from different types of design variables) is of the same magnitude. This is a fair assumption when using the FFD design variables in this dissertation. The alternative extreme is to retain one noise parameter for each variable. Unfortunately this would result in a computationaly expensive learning process in design spaces with even ten design variables.

Covariance Matrix Construction

All this information must be neatly packed into the covariance matrix. This can be approached many ways depending on how the design vectors are organized. In this work, it is done by nesting layers of information. The first layer is the kernel layer, which relates any two points, regardless of type of information. The second is the derivative layer, which relates points, derivatives, and permutations of these. The third layer is the training layer, which relates training data locations with estimated data locations. Table 3.1 helps to further illustrate this dependence tree.

kernel	$k(\mathbf{x}_p, \mathbf{x}_q)$	$=$	$\begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \dots & k(\mathbf{x}_1, \mathbf{x}_M) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_M, \mathbf{x}_1) & \dots & k(\mathbf{x}_M, \mathbf{x}_M) \end{bmatrix}$
derivatives + noise	$k_{\partial\epsilon}(\mathbf{x}_p, \mathbf{x}_q)$	$=$	$\begin{bmatrix} k\left(\mathbf{x}_p, \mathbf{x}_q\right) & k\left(\mathbf{x}_p, \frac{\partial \mathbf{x}_q}{\partial x_w}\right) \\ k\left(\frac{\partial \mathbf{x}_p}{\partial x_v}, \mathbf{x}_q\right) & k\left(\frac{\partial \mathbf{x}_p}{\partial x_v}, \frac{\partial \mathbf{x}_q}{\partial x_w}\right) \end{bmatrix} + k_\epsilon$
training	$[\sigma]$	$=$	$\begin{bmatrix} k_{\partial\epsilon}(\mathbf{x}_p, \mathbf{x}_q) & k_{\partial\epsilon}(\mathbf{x}_p, \mathbf{x}_j^*) \\ k_{\partial\epsilon}(\mathbf{x}_i^*, \mathbf{x}_q) & k_{\partial\epsilon}(\mathbf{x}_i^*, \mathbf{x}_j^*) \end{bmatrix}$

Table 3.1: Kernel function data layers.

In the above, the size of the final matrix $[\sigma]$ is $(M+N)(1+D)$. Note that the kernel matrix becomes rectangular in the case of the training-prediction kernel $k(\mathbf{x}_p, \mathbf{x}_j^*)$ with size $M \times N$. This results in a derivative and noise kernel matrix $k_{\partial\epsilon}(\mathbf{x}_p, \mathbf{x}_j^*)$ of size $M(1+D) \times N(1+D)$. The matrix $[\sigma]$ can grow rapidly with training points and dimension, however the computational cost of manipulating it is much smaller than running more CFD solutions.

To use the gradient-enhanced covariance matrix, the GPR relations shown in Equation 3.4 need only be updated with the new kernel $k_{\partial\epsilon}(\cdot, \cdot)$ for each sub-block of $[\sigma]$.

Data Scaling

Steps are taken to improve the numerics and generality of the method by scaling the data based on the initial LHC sample. Data must be scaled to enable the application of the assumption of isotropy, in the case that the components of the input vector represent different catagories of variables (for example span and twist). The sampled objective function range, and design variable bounds are linearly scaled according to:

$$\begin{aligned}
&f' = (f - f_{off})/f_{ref} && s.t. \;\; [\min(f), \max(f)] && \rightarrow \;\; [0,1] \\
&x_i' = (x_i - x_{i,off})/x_{i,ref} && s.t. \;\; [lb_i, ub_i] && \rightarrow \;\; [0,1] \;\;, \; i = 1, ..., D \\
&\frac{\partial f'}{\partial x_i'} = \frac{\partial f}{\partial x_i} \cdot \frac{x_{i,ref}}{f_{ref}}.
\end{aligned} \tag{3.16}$$

Here, $(\cdot)_{ref}$ is a reference length and $(\cdot)_{off}$ is a reference offset. Together they scale and shift the data to a unit hypercube on inputs and outputs. The domain of inputs are estimated by the lower bounds lb_i and upper bounds ub_i of the original design of experiments, and the range of outputs is estimated by the the maximum and minimum sampled objective. The reference scales are also applied to the gradient to bring it into the same scaled space.

Several benefits are realized from scaling the data past improving the condition of the covariance matrix, if it is fair to assume the response surface is smooth with a nominal amount of variation. It first becomes reasonable to approximately claim the variation of data in all design variables is brought to be the same order of magnitude. This allows an assumption of isotropy for variation in each dimension, which reduces the number of length scale parameters to one, and significantly reduces the computational cost of hyperparameter learning. Second, this allows the claim that the scaled magnitude of the noise parameters for f and $\frac{\partial f}{\partial x}$ are of similar order of magnitude. This is important since the value of the noise in the gradients is difficult to estimate *a-priori*. In general a computational aerospace engineer will have a sense for the unit of precision of the direct analysis. For example in the results for lift and drag run via SU2 it can be said that fully converged results with a practically refined mesh will be able to represent variation in the drag or lift coefficient to about one-count, or 1×10^{-4}. 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.

3.2.3 Fit Evaluation

To evaluate the quality of fit, measures of training and testing error are used. Training error is a measure of the difference between the surrogate model and the training data. When a noise model is included the surrogate is permitted to deviate above and below training data, which allows the training error to grow. Testing error is a measure of the difference between the surrogate model and a separate set of data, called the test set, that was not used to create the surrogate. In principle, the samples should populate in the regions between the training data samples, and thus test the global accuracy of the fit.

The testing error is a stronger measure of accuracy of the surrogate because surrogate models have a risk of "over-fitting". Over-fitting happens when the learning process exploits the upper or lower bounds of the hyper parameters to increase the accuracy of the fit near the training data while sacrificing the accuracy of the fit between the training data. One way this can happen is by decreasing the length scale of the Gaussian kernel (θ_2 in this thesis), which results in a surrogate with that is almost exclusively flat at the mean value of the objective samples, with sharp peaky excursions to meet the individual values of the samples.

In this thesis, both training and testing error are measured using the root mean square difference between the sample set and the surrogate.

$$E_{rms} = \sqrt{\sum_{i=1}^{N} \left(f^*(\mathbf{x}_i) - f(\mathbf{x}_i)\right)^2} \tag{3.17}$$

Relative errors are calculated in the scaled space where objectives are linearly mapped to a range of [0,1], as described earlier in Section 3.2.2. This gives a measure of relative error on the scale of total variation of the data's objective.

3.2.4 Hyperparameter Selection

To use the covariance function, the hyperparameters θ_{1-4} must be chosen. Different values will yield different fits, each being a different view of the data. Presented here is the method of tuning the required hyperparameters by maximizing marginal likelihood[61].

Marginal likelihood measures how well a given set of hyperparameters describes the training data. A common approach is to use the log of the marginal likelihood:

$$\log p(f_p|\mathbf{x}_p, \theta_k) = -\frac{1}{2} f_p^\top [\sigma]^{-1} f_p - \frac{1}{2} \log |[\sigma]| - \frac{n}{2} \log 2\pi, \tag{3.18}$$

where θ_k is a vector of hyperparameters. Rasmussen provides an example algorithm that simplifies these relations by using Cholesky decomposition [61]

Finding the argument maximum is a common way to select hyperparameters for GPR. 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 a two step method of Covariance Matrix Adaptation followed by Sequential Least Squares Optimization for local refinement.

Marginal likelihood maximization is by far the most common method chosen for regression applications of probabilistic models, though two other approaches of note should be mentioned. First is the leave-one-out cross validation approach where by parameters are tuned (usually in an marginal likelihood approach) while leaving one or a subset of training samples out, and measuring the error of the surrogate model with these points. The draw of tuned hyperparameters with the lowest error or highest marginal likelihood is retained. A second approach is to retain a large test set and configure an optimization program to minimize the testing error given the hyperparameters. This can be a favorable approach if sufficient data is available because the testing error measures the interpolation power of the model.

Robustness

Further steps are taken to improve the robustness of the hyperparameter selection process by expressing additional *a-priori* knowledge of the problem in the form of constraints. These constraints are summarized in the Table 3.2.

Table 3.2: Non-linear hyperparameter constraints for MLE maximization.

Constraint	**Motivation**
$\dfrac{\min(\Delta f_i)}{W} < \theta_1 < \max(\Delta f_i) \cdot W$	Avoid under and overfitting due to the nominal variance
$\dfrac{\min(\Delta x_i)}{W} < \theta_2 < \max(\Delta x_i) \cdot W$	Avoid under and overfitting due to the length scale
$\dfrac{\theta_3}{\theta_1} > 1 \times 10^{-8}$	Maintain well conditioned numerics
$\dfrac{\theta_3}{\theta_1} < 1 \times 10^{-1}$	Avoid interpreting data as noise
$\theta_3 < \theta_4$	Trust objective function data before gradient data

These constraints add robustness to the learning process by encouraging not only favorable nominal variance and length scale parameters θ_1 and θ_2, as well as favorable noise hyperparameters, θ_3 and θ_4.

The first two input bounds are enforced on the hyperparameters to ensure that they stay within values relevant to the sampled data. These bounds are chosen to avoid modeling more, or less, variation than the training data could hope to predict. This is why they are tied to the maximum and minimum observed differences in both function value and data location. The buffer constant W is chosen to allow larger or smaller values. A value of 100 was found to be useful by trial and error and was applied consistently throughout this dissertation.

The third and fourth constraints couple the behavior of the nominal variance and noise parameter to avoid numerical edge cases. The two parameters are related because the noise parameter is added to the kernel's diagonal, which without the noise term will be equal to the nominal variance parameter. Thus the ratio of these two terms drive the numerical damping of the problem. If the ratio becomes too low, as in the case of the third constraint, then the problem is at risk of becoming numerically unstable in the presence of a large sample of functions and gradients. This problem was encountered in this dissertation, and has been expressed as a major issue in literature [79, 80]. The constraint of 1×10^{-8} was chosen by hand and in general is not encountered for most numerical experiments involving CFD data. On the other end of the ratio of noise and nominal variance parameters is the case where all the data is interpreted as noise, resulting in a fit that is essentially flat at the data mean. The fourth constraint addresses this by saying the noise ratio should not exceed 10% of the nominal variation in the data.

The last constraint $\theta_3 < \theta_4$ is a safe guard against the gradient data from overpowering the behavior of the fit such that it does not respect the function data. If gradient data had not been scaled to the same order of magnitude as the objective data, this constraint would not be appropriate, since it requires the two parameters to be of the same order of magnitude.

In the case of high dimensional problems, there is an especially large amount of gradient data. When numerically estimated gradients are not *exactly* correlated to numerically estimated functions, then the important assumption of Equation 3.6 is violated. In this case the gradient information, for which there is one component for each dimension, is numerically correlated with each other and will generally be honored before the function information, for which there is only one component. In flow simulations, this can happen when using gradients from continuous adjoints because they are not the numerically exact derivative of the direct solution, but rather they are numerically exact to the physical problem. Discrete adjoints on the other hand are formulated such that they are indeed numerically exact to the direct solution, and would potentially be better applied in the case of gradient enhanced regression. However these issues are effectively mitigated by the inclusion of the independent gradient noise parameter and this constraint, which encourages the optimization problem to land in a local minimum in which the function data is respected.

3.3 Numerical Experiments

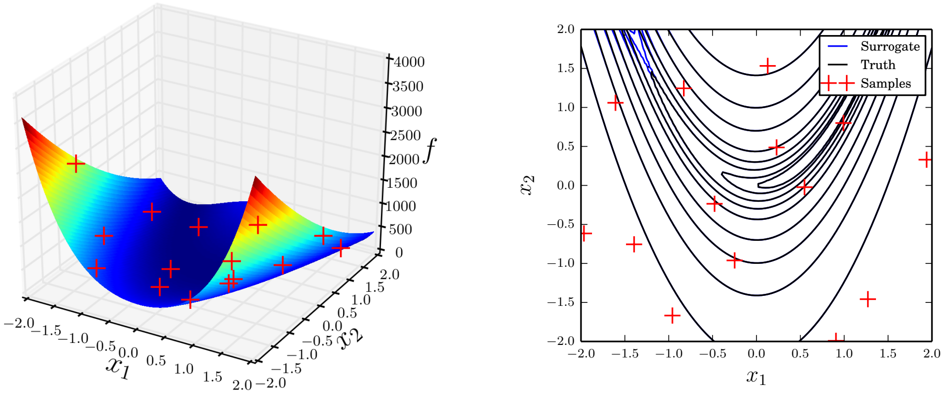
This section will show numerical experiments that demonstrate the algorithms presented in the previous section. An important theme here in is investigating trends in training and testing error as a measure of accuracy of the surrogate models. These errors can grow in unfavorable ways with the presence of gradient errors, since there are many gradients compared to objectives.

3.3.1 Analytical Test Problem

3.3.1.1 A Fitting Example

This section presents quick motivation for the usefulness of gradients. As discussed in the background, a large existing literature exists for gradient enhanced models, so the example will be brief.

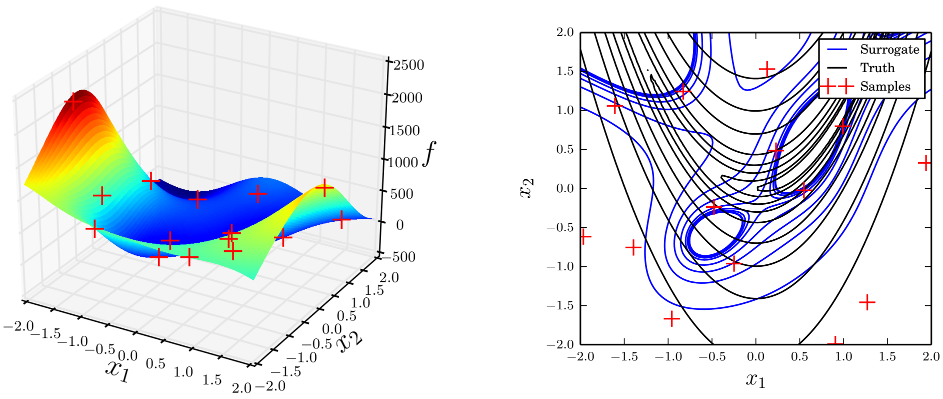
The Rosenbrock function is a special test function because of the large range in objective values, and the relatively flat region in which the minimum is found. Using gradients, it is possible to regress the Rosenbrock function with fifteen test points and yield accurate results. The plots of Figure 3.1 show the banana-shaped valley is modeled nicely. In the complementing Figure 3.2, gradients were omitted from the same training data. The accuracy of the fit clearly suffers. Nearly 40 samples of objective values would be needed here to make a comparable fit.



(a) GPR surface.

(b) Contours of the surrogate model and truth function.

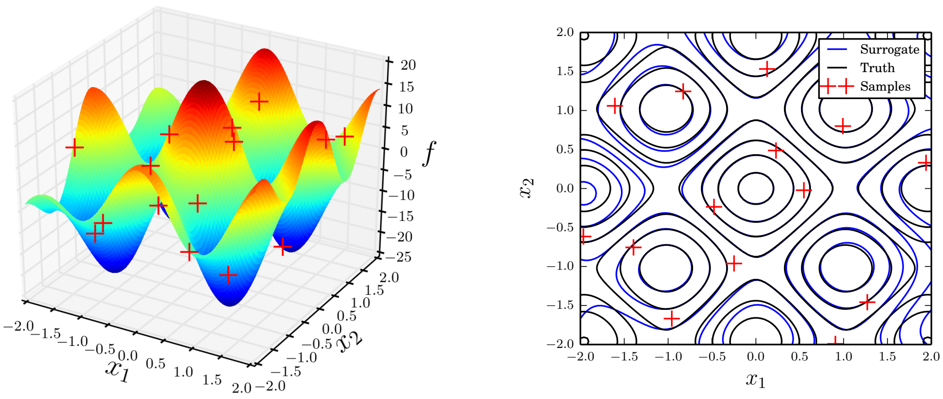
Figure 3.1: Plots of a gradient-enhanced GPR model of the Rosenbrock function.



(a) GPR surface.

(b) Contours of the surrogate model and truth function.

Figure 3.2: Plots of a GPR model of the Rosenbrock function, *without gradients*.



(a) GPR surface.

(b) Contours of the surrogate model and truth function.

Figure 3.3: Plots of a gradient-enhanced GPR of the modified Rastrigin function.

A gradient-enhanced fit of the modified Rastrigin function is also presented, again with fifteen samples, and is shown in Figure 3.3. In this case the multi-modality of the function is well modeled, with some deviations near the corners of the domain that indicate the need for minor refinement.

3.3.1.2 Characterization of Learning Robustness

An important component of the surrogate modeling formulation in this thesis is the use of constraints on the hyperparameters during the optimization problem that maximizes marginal likelihood. A surprisingly simple but problematic regression case exposes the tendency for the covariance matrix to become singular.

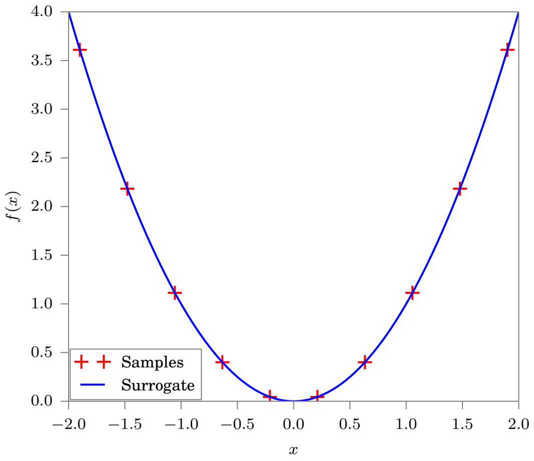


Figure 3.4: Surrogate model of a parabola $f = x^2$ with 10 samples.

Here a parabola in one dimension is fit using ten samples without gradients. In this example the samples are linearly spaced, though a space filling random sample can be used as well. This problem is setup to illustrate the tendency of the marginal likelihood process to drive into larger hyperparameters to create a fit, resulting in numerically unsolvable models.

A Gaussian process regression is applied to the data using various packages. In the first case, a standard regression tool written with Matlab, called GPML, is applied using a squared exponential kernel function with a Gaussian noise model, and a BFGS gradient based optimization algorithm to maximize marginal likelihood. During the hyperparameter learning process, important values are tracked, including the hyperparameters, the value of the marginal likelihood, and the condition of the kernel matrix (with respect to the norm ratio with an inverted matrix). As the condition number of the matrix increases to large values, the matrix is becoming singular and the linear algebra required to compute marginal likelihood becomes intractable.

This is what happens when running the marginal likelihood optimization without constraints for surrogates with reasonably dense amounts of data. The convergence history presented in Figure 3.5 shows that the nominal variance θ_1 and length scale θ_2 become very large, order 1×10^5 and 1×10^{10} respectively. This is possible because the behavior of the fit is strongly driven by the ratio of these two hyperparameters. From the perspective of the marginal likelihood there is a very small improvement that the gradient based optimizer exploits to drive to a maximum. Unfortunately as these parameters become large the covariance matrix becomes singular, shown with the condition number cond($\boldsymbol{K}$) becoming order 1×10^{18}. At this point error messages print warning that the Cholesky decomposition step is failing and the optimization stalls.

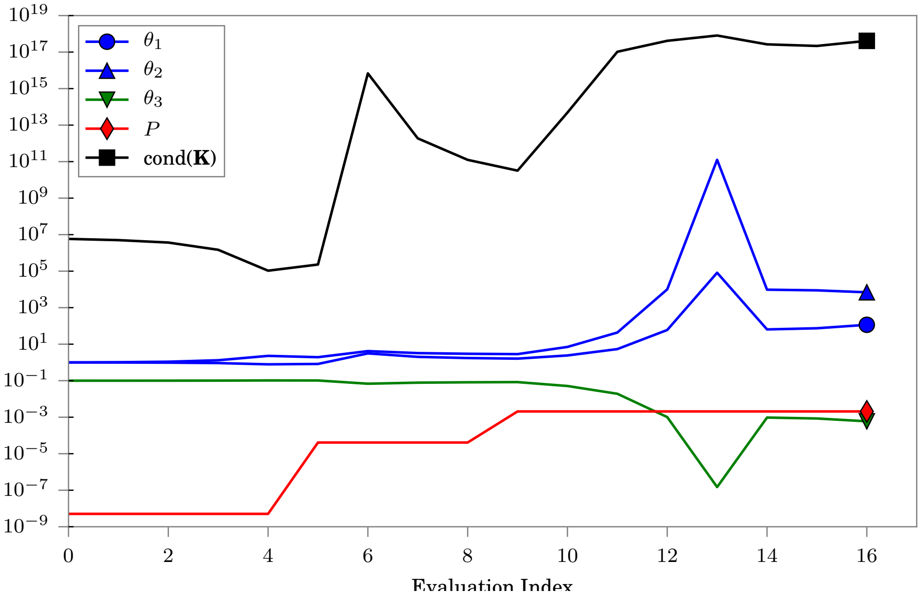


Figure 3.5: Learning history for the parabolic problem, using GPML, without hyperparameter constraints.

A similar result is encountered with the learning process developed for this thesis, when hyperparameter constraints are not applied. The learning history for the parabolic regression using VyPy [86], a toolbox developed as part of this dissertation, is shown in Figure 3.6. As with the previous case the hyperparameters drive into large values, causing the covariance matrix to become more singular, as indicated by the condition number to growing to 1×10^{18}.

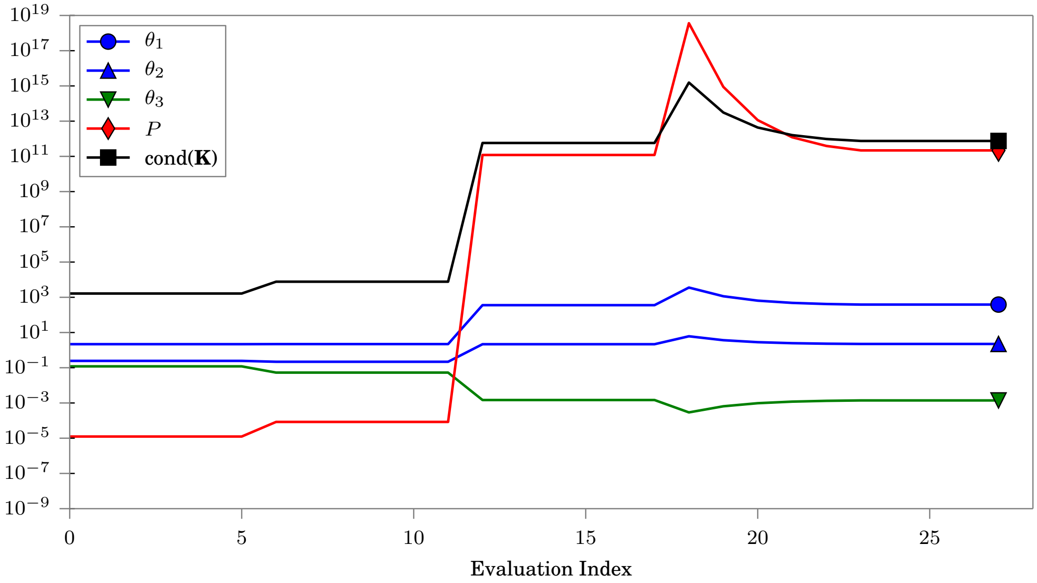


Figure 3.6: Learning history for the parabolic problem, using VyPy, without hyperparameter constraints.

It is conceivable that the singularity of the covariance matrix could be applied as a constraint to the hyperparameter learning problem. One would maximize marginal likelihood subject to a maximum condition number. However calculating the condition number is expensive because it requires calculating the full inverse of the matrix. This is why the ratio of the noise parameter θ_3 to nominal variance θ_1 is used as a surrogate, since the noise parameter adds numerical stability to the problem by increasing the norm of the trace.

As Figure 3.7 shows, this is able to keep the condition number of the matrix in numerically non-singular ranges. In this case the noise parameter was initialized at a low value, 1×10^{-6}, which results in a larger covariance matrix condition number. As the learning progressed the parameters are adjusted to not only increase the likelihood to similar or larger values as the previous examples, but maintain a condition number of less then 1×10^{10}, and not encounter any singular matrix errors.

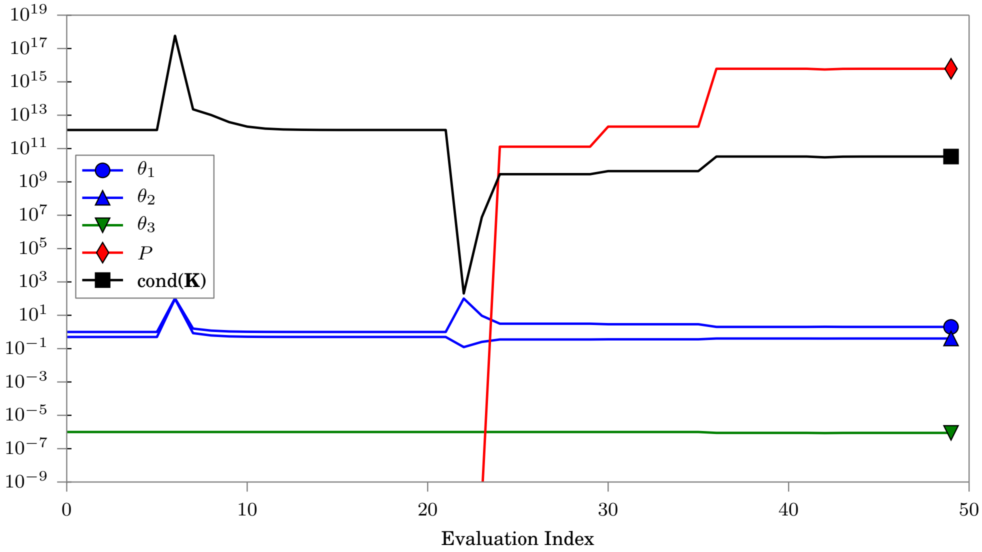


Figure 3.7: Learning history for the parabolic problem, using VyPy, including hyperparameter constraints.

3.3.1.3 Characterization of Learning Efficiency

When working with real world data, that comes from CFD simulations for example, a large premium is placed on the data. It is expensive to run realistic problems. In this dissertation some problems take three hours on 80 cores to collect one sample with objective and gradients, and there are problems that are much more expensive than this. So the fewer data needed to build an accurate fit, the better. This is why the constraints presented in Sections 3.2.4 were explored. By preconditioning the MLE optimization problem, fewer data are needed to make a good fit.

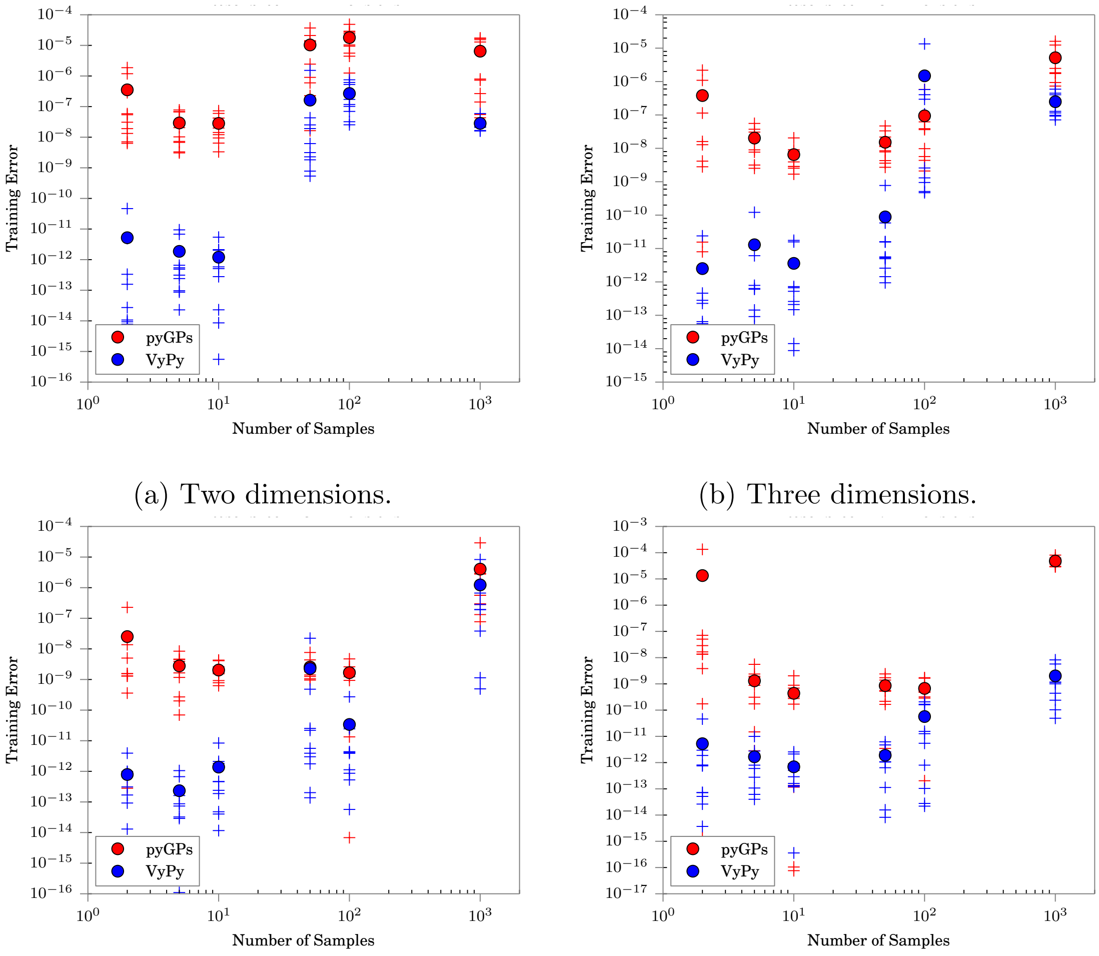
The benefit of the approach is examined with the Rosenbrock function. In this experiment, the number of samples vary between 2 and 1000 for the Rosenbrock described in different dimensions. The upper bound of 1000 samples is chosen as an example of being data-constrained during a design problem and is enforced regardless of the number of dimensions. Training and testing error are each compared between the methods described in this dissertation (implemented in a package called VyPy [86]), with the standard GPML methods described Rasmussen (implemented in a python package called pyGPs [87]).

Because this experiment is highly dependent on a random sample of training and testing data, each category of number of samples and problem dimension is repeated ten times. The result for each sub-experiment is reported with crosses and the average of the experiments is reported with a filled circle.

The y-axis shows the root mean square relative training error, where 1 (or 100%) would represent a difference magnitude that is comparable to the range of the training set objective. The range of the Rosenbrock function in for this problem (described in the design problem chapter), is on the order of 1×10^4. So a relative error of 1×10^{-5} is adequate but not ideal for a sparsely sampled surrogate.

In the first set of graphs presented in Figure 3.8, the training error is compared between this dissertation and GPML. One expected trend to observe for both approaches is the diminishing returns that come with more samples in higher dimension. With more samples to consider, the numerics of the problem prevent it from respecting every point accurately. This is because matrix $k(\mathbf{p}, \mathbf{q})$ becomes larger, and has to be inverted in an $\boldsymbol{A}\mathbf{x} = \mathbf{b}$ problem. As numerical limits creep in, the marginal likelihood surface to becomes noisy, and this affects the learning problem.

Another trend that that is apparent in the plots is that at lower number of samples the surrogate is able to achieve significantly smaller training errors. This accuracy gain reduces as the number of samples increases. This is one of the major results of having the hyperparameter constraints.



(a) Two dimensions.

(b) Three dimensions.

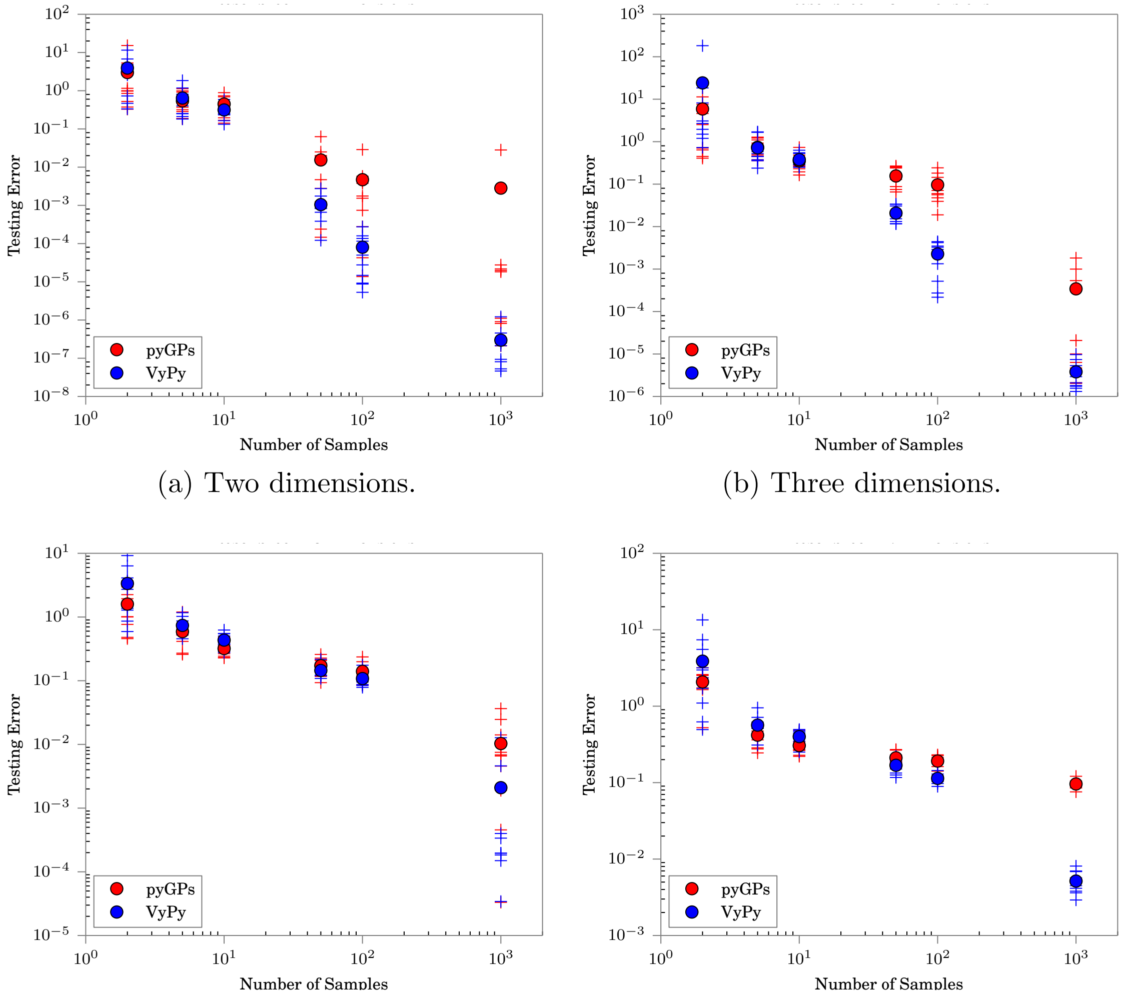
(c) Five dimensions.

(d) Seven dimensions.

Figure 3.8: Variation of training error with increasing number of samples, while fitting a GPR surrogate model to the Rosenbrock function in different numbers of dimensions. A comparison is shown between the implementation presented in this thesis (VyPy) and that based on a standard GPR package (pyGPs, based on GPML). The blue crosses indicate individual fit attempts, and the circles indicate the average of the ten attempts.

In this second set of graphs presented in Figure 3.9, the testing error is plotted with increasing number of training samples, for various dimensionalities of the Rosenbrock function. The test set size was scaled with increase dimension according to $100 \cdot 2^n$

In general, a machine learning problem can expect the testing error to decrease with increasing number of training samples. Gains are available in how fast the error can decrease with more training samples. The figures suggest that by conditioning the hyperparameter learning problem with physically insightful constraints, a gain of about one order of magnitude in test set accuracy is available in larger sizes of training sets, even in high dimension, without sacrificing the testing accuracy with smaller sizes of training sets, which in the earlier set of figures demonstrated a significantly reduced training error.



(a) Two dimensions.

(b) Three dimensions.

(c) Five dimensions.

(d) Seven dimensions.

Figure 3.9: Variation of testing error with increasing number of samples, while fitting a GPR surrogate model to the Rosenbrock function in different numbers of dimensions. A comparison is shown between the implementation presented in this thesis (VyPy) and that based on a standard GPR package (pyGPs, based on GPML). The blue crosses indicate individual fit attempts, and the circles indicate the average of the ten attempts.

3.3.1.4 Effects of Simulated Gradient Errors

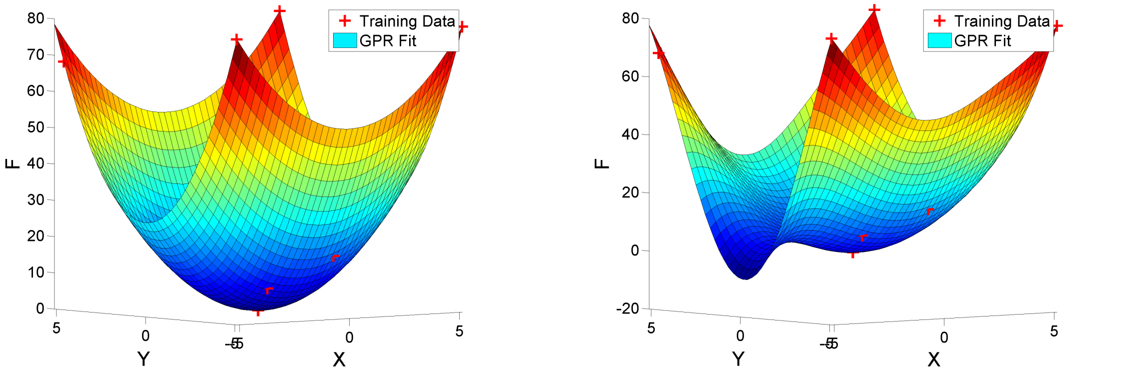
Given the presence of gradient bias error, it is useful to see its effect in a simple and controlled surrogate modeling test case. This example examines a response surface fitted to a two-dimensional paraboloid.

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,

$$\frac{\partial y'}{\partial x'} = (1+\Delta)\frac{\partial y}{\partial x}, \tag{3.19}$$

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

Figure 3.10 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. Table 3.3 quantifies this by showing where the surrogate's minimum is located. In this table one can see that this error grows with increasing gradient bias error.



(a) Parabolid surrogate with accurate gradients.

(b) Parabolid surrogate using gradients with simulated bias error.

Figure 3.10: Response surfaces for an example regression with inaccurate gradients.

Gradient	**X_{opt}**	**F_{opt}**
Bias Error	**Error**	**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

Table 3.3: Error data for example surrogate model with increasing bias error. The errors show the distance from the surrogate's prediction of the global minimum. In the case of the last three rows the error in the gradient data creates a new minimum.

3.3.2 Surrogates on Flow Simulations

With the introduction of computational flow simulations, noise considerations become important. This section will characterize the noise found in a test CFD problem based on the NACA 0012 airfoil, which was overviewed in Section 2.4.1.

3.3.2.1 Gradient Error Characterization

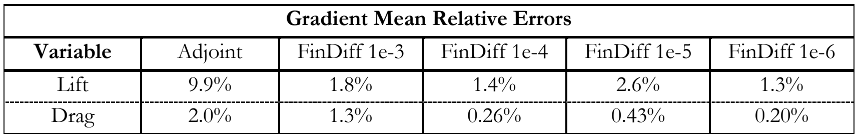
This experiment compares the trends and values of the gradients for continuous adjoint and finite difference based sensitivity analyses, at 41 different magnitudes of a single Hicks-Henne bump function, on the lower airfoil surface at the mid-chord. The reference for error estimation was the response surface based approach to estimating gradients in a low-dimensional design space. If one can accept the direct solutions as physically representative, then an accurate response surface based on only the training data should produce a gradient that is similarly accurate, because the GPR gradients are available analytically. *This makes surrogate modeling useful for evaluating numerically accurate gradients in low dimension for expensive function evaluations.*

Figure 3.11 compares 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 3.4. 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.

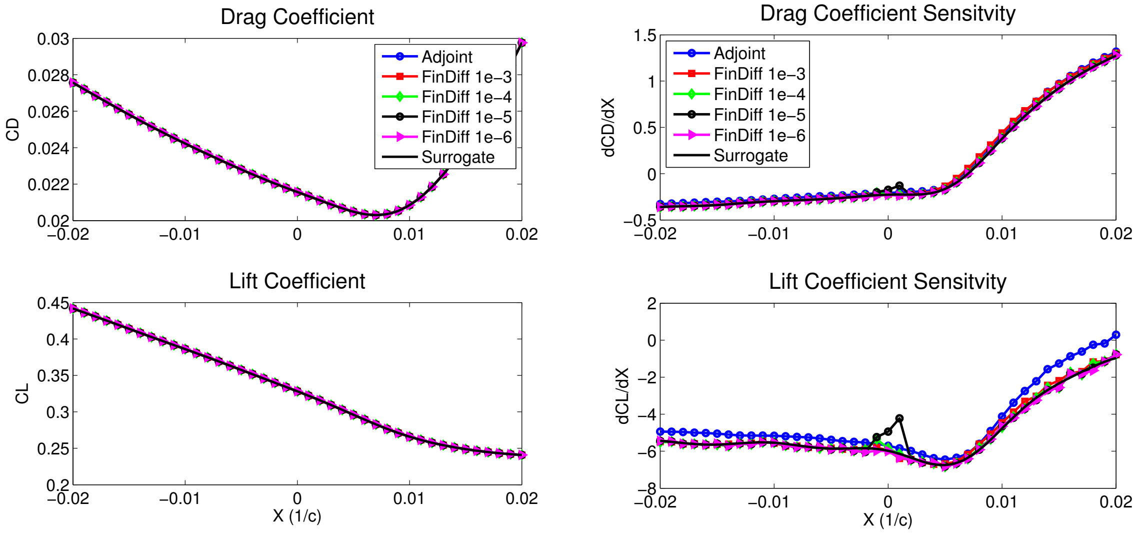
In contrast, low bias error but short-scale noise can be seen occurring in the finite difference gradients. There are especially large magnitude errors present in step lengths 1×10^{-4} and 1×10^{-5}. A closer look at the the noise is shown in Figure 3.12. The source of the noise is presumably from various shock lines moving between cells during perturbation. The accuracy of the shock is limited by the numerical resolution of the grid, as well as the need for three nodes to define the location. With small perturbations, this limitation begins to show itself as "noise" in the design space.



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 3.4: Baseline mesh gradient errors for the NACA0012 example.

The performance of the 1×10^{-3} step is likely due to the favorable amount of smoothing with its larger step length. This of course may trade accuracy in some design spaces that are not as smooth as the current. Caution is required when considering the 1×10^{-6} step 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, a



(a) Objectives.

(b) Gradients.

Figure 3.11: Baseline mesh objective and gradient sweeps

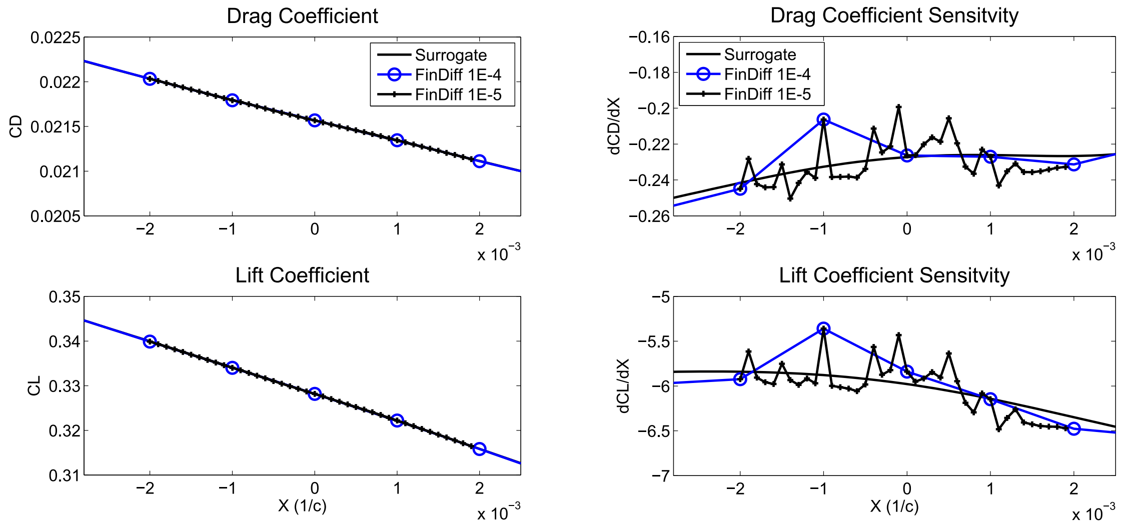
fair judgment can be made that the finite difference step of 1×10^{-3} results in the most physically representative estimation of the gradient.

Also of note is that the continuous adjoint estimation of the lift gradient is less accurate than the estimate of the drag gradient. This is because SU2 surface based adjoint approach relies on a central differencing scheme with artificial dissipation constants that were calibrated for the drag gradient. While it is conceivable to tune these parameters for the lift gradient, there is merit trying to avoid internal modifications to the adjoint method in favor of treating it as a black box.

3.3.2.2 Response Surfaces with Noise Tolerance

This experiment will apply the 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 3.13 - 3.15, a response surface is generated using 20 Latin hypercube sampled training data with continuous adjoint based gradients. Then the noise parameter of the surrogate model is modified to



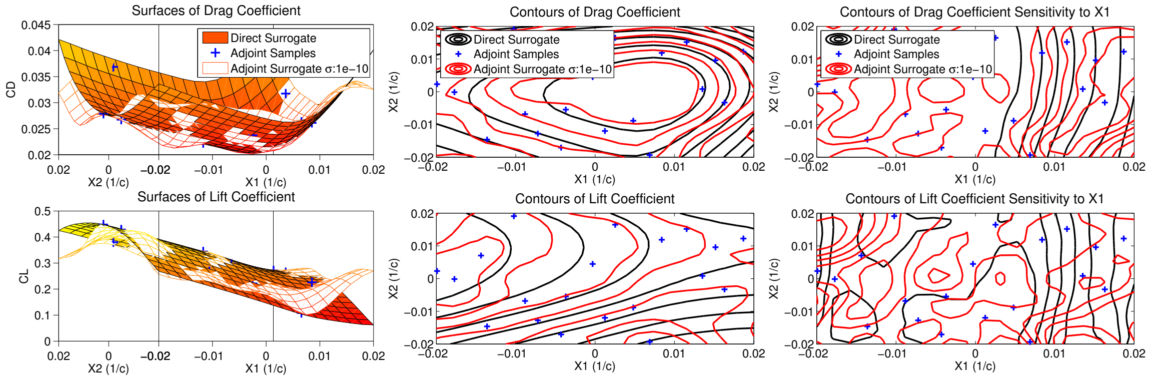
(a) Objectives of dense samples for drag and lift coefficient.

(b) Gradients by finite difference approximations using the samples in Figure 3.12a.

Figure 3.12: Detail of finite differencing noise. Note the scale of the input variable is 1×10^{-3}. The finite differenced gradients represented in (a) are found using forward differencing on the objectives of the corresponding marker in (b). This means that the blue circles have a finite difference step of 1×10^{-4} and the black points have a step of 1×10^{-5}. The surrogate gradient is found analytically from the surrogate fit on only the objective data in (a).

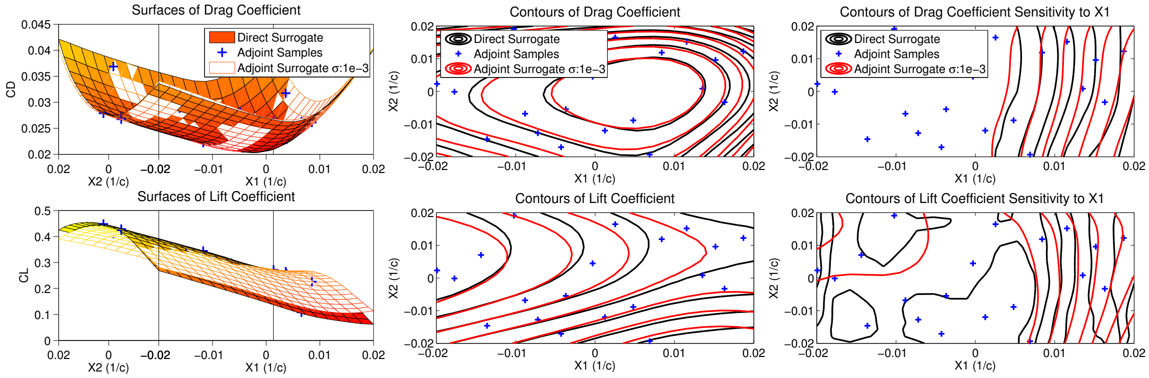
study the fitting performance. For reference, a comparison is made between the fits to a response surface generated from a 10x10 grid of direct solution data only.

The data shows in Figure 3.14 that if a noise of 1×10^{-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 3.15. If noise is not allowed, or restricted to be very low as in the case of Figure 3.13, then the accuracy of the response surface degrades significantly. Again as in the analytical case, additional local minima are generated. This reflects a major observation of this study that response surface with noise-tolerant covariance models are more robust to gradient inaccuracies.



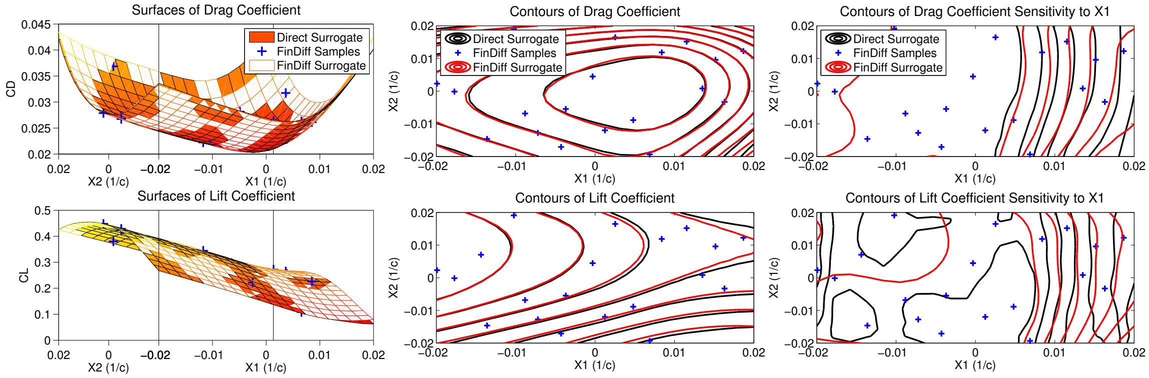
Mean errors: Lift objective: 5.5%; Lift gradient: 50.8%;
Drag objective: 4.8%; Drag gradient: 12.8%

Figure 3.13: Example RSM with adjoint gradients, noise tolerance=1×10^{-10}



Mean errors: Lift objective: 3.3%; Lift gradient: 10.9%;
Drag objective: 1.2%; Drag gradient: 3.4%

Figure 3.14: Example RSM with adjoint gradients, noise tolerance=1×10^{-3}



Mean errors: Lift objective: 0.31%; Lift gradient: 4.0%;
Drag objective: 0.75%; Drag gradient: 2.2%

Figure 3.15: Example RSM with finite differencing, noise tolerance=1×10^{-3}

3.4 Summary

A core goal of this chapter was to integrate several developments in surrogate modeling to enable a robust regression procedure in the face of sparse data and inaccurate gradients. To address the issue of multiple local minima in marginal likelihood surface used to learn hyperparameters, a series of constraints were developed to encourage the problem to land in a local minimum that was physically representative of the data.

The issue of gradient inaccuracies was presented with an analysis of where it impacts the surrogate modeling process. To characterize the gradient inaccuracies from adjoint methods used in this dissertation, a surrogate-based gradient estimation technique was applied by analytically solving 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 the estimation of a root-mean-squared error of the gradients across the entire design space. These results were used as motivation to apply a common noise model for GPR, modified to enable the independent treatment of objective and gradient noise terms, and constrained during the hyperparameter learning process to ensure that objective information was honored before gradient information. This was a core contribution enabling the robust regression of gradient information.

Chapter 4

Active Subspaces

This chapter describes the theory behind the active subspace method, and will then investigate the existence and properties of active subspaces for various design problems.

4.1 Background

Active subspaces are a recent development in the area of reduced order modeling. Reduced order modeling seeks to identify a lower dimensional representation of data that sufficiently describes the important behaviors of the data. Several strategies for reducing dimensionality have been explored in the past, including variable subset selection, linear decomposition, and non-linear decomposition methods.

Subset selection methods rely on measures of sensitivity of an output to the various inputs, potentially allowing some parameters to be omitted from the analysis with small sensitivity. Examples include ANalysis OF VAriation (ANOVA) decomposition [88, 89], Main Effects Screening (MES) [88], and Elementary Effects Screening (EES) [90]. These measures of sensititvity are typically formulated as integrals over the parameter space, so are difficult to evaluate in high dimension [91].

Linear decompositions look for subsets of linear combinations of the inputs that sufficiently describe the data. This generalizes subset-based dimension reduction which otherwise would set the weights of the linear combination all to zero except for the one associated with the parameter of interest. A well known linear dimensionality reduction method is Prcinipal Component Analysis (PCA) [92]. PCA is typically used to reduce the dimensionality of an output space that contains a high degree of correlation. However in the context of optimization in this dissertation, the goal is to reduce the dimensionality of the input space. Proper Orthogonal Decomposition (POD) is closely related but often is applied with basis functions chosen to be suitable for reducing the size of large scale systems such as CFD solvers [93, 94, 95, 96]. These methods are intrusive to the analysis method, and is avoided in this dissertation in favor of leveraging existing analysis tools as black boxes.

Non-linear decomposition is an area of active research, which seeks to represent data using non-linear basis functions. This enables for example the discovery of helical curves in three input dimensions. In the context of this dissertation such behavior is not expected for the problems under consideration. A review of current non-linear reduction methods is available [97].

Active subspaces applies PCA to operate on gradient information of scalar-valued outputs, allowing the reduction of the input space. The approach was first proposed and coined by Russi [98], and demonstrated in the analysis and optimization context by Constantine [55, 99]. It has been demonstrated in aircraft surrogate based optimization with prior work of this dissertation [100]. Independent work in the area of applying PCA to gradients has been accomplished in the context of accelerating gradient based optimization [101, 102] in aircraft.

4.2 Method Formulation

4.2.1 Subspace Construction

Consider a scalar function f of a column vector $\mathbf{x}$, and its gradient $\nabla_{\mathbf{x}} f$ oriented as a column vector:

$$f = f(\mathbf{x}), \quad \nabla_{\mathbf{x}} f = \frac{\partial f}{\partial x_v}, \quad \mathbf{x} \in \mathcal{X} = [-1, 1]^m. \tag{4.1}$$

where m is the dimension of the design vector $\mathbf{x}$, and $v = 1, ..., m$.

Here assumed is 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 during an active subspace analysis. The task at hand is to determine the orthogonal directions that most effectively describe the variability of the objective f.

The first step is to encode the global correlation of the gradients in a way that will lend itself to an singular value decomposition (SVD) analysis. The matrix $\boldsymbol{C}$ is defined as the outer product of the gradient with itself:

$$\boldsymbol{C} \;=\; \int \left(\nabla_{\mathbf{x}} f\right) \left(\nabla_{\mathbf{x}} f\right)^{\top} \rho \, d\mathbf{x}, \tag{4.2}$$

where in this dissertation $\rho = \rho(\mathbf{x})$ is chosen to be a uniform probability density on $\mathcal{X}$. Each element of $\boldsymbol{C}$ is the average of the product of partial derivatives. The matrix is positive symmetric and positive semi-definite of size $m \times m$. If we consider the gradients to be randomly sampled, then the $\boldsymbol{C}$ is the uncentered covariance matrix of the gradient. As a result, this matrix is sensitive to choice of scale, and the data should be centered and normalized before performing active subspace analysis.

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

$$\boldsymbol{C} \;\approx\; \frac{1}{M} \sum_{i=1}^{M} \nabla_{\mathbf{x}} f_i \, \nabla_{\mathbf{x}} f_i^T, \tag{4.3}$$

where $\nabla_{\mathbf{x}} f_i = \nabla_{\mathbf{x}} f(\mathbf{x}_i)$, and $\mathbf{x}_i$ is drawn at random according to density function ρ from $[-1, 1]^m$. This approach compliments well with surrogate modeling, which also requires a well spread set of functions and gradients. This enables an important amount of data reusability.

The sampling requires M evaluations of the objective f and its gradient $\frac{\partial f}{\partial \mathbf{x}}$. When interrogating aircraft designs with CFD simulations, adjoint methods reduce the computational expense because they 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, find the eigenvectors of the covariance matrix. This matrix is symmetric and positive semidefinite, so it admits a real eigenvalue decomposition,

$$\boldsymbol{C} \;=\; \boldsymbol{W}\boldsymbol{\Lambda}\boldsymbol{W}^T, \tag{4.4}$$

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

Note that for many aerospace 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.

To form a reduced-order basis, sort the eigenvectors according to descending eigenvalues, then select the first n eigenvectors. The heuristic for selecting the size of the subspace dimension will be discussed shortly. After choosing n directions to keep, the eigenvectors and eigenvalues can be partitioned accordingly:

$$\boldsymbol{W} \;=\; \begin{bmatrix}\boldsymbol{U} & \boldsymbol{V}\end{bmatrix}, \qquad \boldsymbol{\Lambda} \;=\; \begin{bmatrix}\boldsymbol{\Lambda}_{\mathbf{y}} & \\ & \boldsymbol{\Lambda}_{\mathbf{z}}\end{bmatrix}, \tag{4.5}$$

where $\boldsymbol{U}$ contains the first n columns of $\boldsymbol{W}$, and defines the active subspace of the input's full space.

With the basis identified, the sample locations can now be mapped forward to the active subspace,

$$\mathbf{y} = \boldsymbol{U}^\top \mathbf{x}, \quad \mathbf{y} \in \mathbb{R}^n, \tag{4.6}$$

and the function f can be approximated in this active subspace,

$$f(\mathbf{x}) \approx g(\boldsymbol{U}^\top \mathbf{x}) = g(\mathbf{y}). \tag{4.7}$$

The intuition for this stems from the following relationship of eigenvectors, and eigenvalues [99],

$$\int \left((\nabla_{\mathbf{x}} f)^\top \mathbf{w}_i\right)^2 \rho \, d\mathbf{x} = \lambda_i, \quad i = 1, ..., m, \tag{4.8}$$

which says that the derivative in the direction of the eigenvector, mean-squared over the input domain, is equal to the eigenvalue. Thus if λ_i is zero, then the mean-square of the gradients are zero, and the function is completely flat in the direction of the eigenvector. Thus it is possible to describe the behavior of the objective function by projecting the full-space design variables into this *active subspace* [98]. *This enables the dimensionality of the design problem to be reduced.* Low eigenvalues suggest that the corresponding vector is in the null-space of the covariance matrix, and that these vectors can be discarded to form an approximation. An important judgment call must be applied however to choose the dimension of the active-subspace basis. A heurisitic to inform this judgement will be presented in Section 4.2.2.

The domain of g, the function approximated in the active subspace, is

$$\mathcal{Y} \;=\; \left\{ \mathbf{y} = \boldsymbol{U}^T \mathbf{x}, \quad \mathbf{x} \in \mathcal{X} \right\} \;\subset\; \mathbb{R}^n, \tag{4.9}$$

which indicates that the bounds of the full space $\mathcal{X}$ must be respected when exploring the active space $\mathcal{Y}$. The bounds of the full space are defined as an m-dimensional hypercube. This hypercube projects into the active subspace to form a “zonotope”. The hull of the zonotope is convex, but identifying the corners is a difficult problem when projecting from high dimension. In practice this can be addressed by retaining information of the forward map, as will be shown in Section 5.2.2.

In addition to the active subspace, the inactive subspace is defined with the complimenting map,

$$\mathbf{z} = \boldsymbol{V}^\top \mathbf{x}, \quad \mathbf{z} \in \mathbb{R}^{m-n}. \tag{4.10}$$

It is permissible to apply a surrogate model in the active subspace,

$$g(\mathbf{y}) \approx g^*(\mathbf{y}) \;\equiv\; \mathcal{R}(\mathbf{y};\, g_1, \ldots, g_N), \tag{4.11}$$

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 $g_1, \ldots, g_N$, where each training point is evaluated $g_i = g(\mathbf{x}_i)$.

Gradients can be projected

$$\frac{\partial g}{\partial \mathbf{y}} = \boldsymbol{U}^\top \frac{\partial g}{\partial \mathbf{x}}, \quad \frac{\partial g}{\partial \mathbf{z}} = \boldsymbol{V}^\top \frac{\partial g}{\partial \mathbf{x}}. \tag{4.12}$$

Caution is needed when using gradients projected into the active subspace. Gradients near corners of the full space for example will project into the active subspace at similar locations with different directions. Thus a large noise parameter may be needed for the gradient component of the surrogate. It would not be surprising if the noise parameter must be so large that the gradient information does not contribute to the surrogate model in the active subspace.

To estimate the covariance matrix and accomplish the analysis above, a total of M gradient samples are required. It is important to have an idea of how large this sample should be, in order to enable a reasonably accurate eigenvalue analysis. Work by Constantine [99] has shown that number of samples needed scales with

$$M = a\ k \log(m), \tag{4.13}$$

Where the constant a is a chosen oversampling factor between 2 and 10, k is the number of eigenvalues that are to be approximated, and m is the dimension of the input space. To tie this to the problems exercised in this dissertation, in the case of 100 design variables and trying to estimate 10 eigenvectors, anywhere between 92 and 460 gradient samples could be needed to predict the eigenvectors with reasonable accuracy.

The accuracy of the eigenvectors can be assessed using the bootstrap method. In this method, gradients are randomly re-sampled from the original training set. Duplicated gradients are allowed. These re-sampled gradients are used to calculate the eigenvectors and values. The process is repeated several times, typically between 100 and 10000. The results of these repeated analyses can be processed to yield statistics such as the variance, as well as upper and lower bounds of the eigenvalues and eigenvector components.

4.2.2 Subspace Selection

Two heuristics are available for selecting the dimension n of the active subspace. In the first approach, one extends Equation 4.8 to find that the sum of the eigenvalues of the active and inactive spaces represent the mean-square of the gradients in that space:

$$\begin{aligned} \int (\nabla_{\mathbf{y}} f)^\top (\nabla_{\mathbf{y}} f) \rho \, d\mathbf{x} &= \lambda_1 + \cdots + \lambda_n, \\ \int (\nabla_{\mathbf{z}} f)^\top (\nabla_{\mathbf{z}} f) \rho \, d\mathbf{x} &= \lambda_{n+1} + \cdots + \lambda_m. \end{aligned} \tag{4.14}$$

One can then look at the ratio of the sum of the inactive eigenvalues to the sum of all the eigenvalues as a relative measure of how much variability of the gradients is left unmodeled in a mean-squared sense,

$$\varepsilon_{\nabla n} = \frac{\sum_{j=n+1}^{m} \lambda_j}{\sum_{i=1}^{m} \lambda_i}. \tag{4.15}$$

One can thus plot ε_∇ for varying n and choose a threshold value, for example 0.01 or 1%, to partition the active and inactive spaces.

The second approach works on the result from Constantine [99] that shows under certain assumptions the distance between eigenvalues is related to the error of the subspace described by the retained eigenvectors.

$$\varepsilon_{\Delta n} \leq \Delta \boldsymbol{W}_n \sqrt{\sum_{i=1}^{n} \lambda_i} + \sqrt{\sum_{j=n+1}^{m} \lambda_j}. \tag{4.16}$$

Here the subspace error $\Delta \boldsymbol{W}_n$ is estimated using the bootstrap method, also known as resampling with replacement. Constantine [99] provides a thorough background and algorithm on applying this to active subspaces. This relation is a heuristic that shows that a tradeoff exists between the accuracy of the prediction of the active subspace basis, and the remaining variability in the inactive space. Intuitively this is saying that if the eigenvectors point in the wrong direction, then the data will not collapse as well into a manifold.

In practice the result of this analysis shows "gaps" in eigenvalues when viewed on a logarithmic scale are strongly correlated with the error first major opportunity for reducing the dimensionality of the problem. The gap becomes a sensor that indicates enough eigenvectors have been retained to sufficiently approximate the function. This behavior is also observable by looking for troughs in the spectrum of $\varepsilon_{\Delta n}$. If eigenvectors before the gap are omitted, the ability of the basis to collapse the data into a manifold is reduced. Adding subsequent eigenvectors indicates that one is looking to capture the next order of behavior, and at least as many eigenvectors as those needed to reach the next gap should be retained for this purpose. In another sense, one should avoid choosing an active space that has a dimension falls between major eigenvalue gaps.

4.2.3 Deformation Modes

For problems involving geometry, the active subspace information can be used to visualize shape changes that result in the greatest changes in an objective on average.

In setting up aircraft geometry-based design problems, there is an important map, that takes the full space design vector and returns a three dimensional surface. This is the "parameterization" of the problem.

$$\mathcal{X} \xrightarrow{s(\mathbf{x})} \mathcal{S} \tag{4.17}$$

The eigenvectors chosen by the active subspace analysis are embedded in the full space and points along the direction that results in the largest change in the objective on average. For visualization one can pick individual eigenvectors, $\mathbf{u}_i$ from the eigenbasis $\boldsymbol{U}$. Choosing y, the coordinate plotted along this basis vector, the eigenvector is scaled to modify the surface under the given parameterization,

$$\mathcal{X} \xrightarrow{s(y\mathbf{u}_i)} \mathcal{S}. \tag{4.18}$$

Now as the active subspace coordinate y is varied, one can compare the objective's behavior in the active subspace to the shape change that accompanies it in the surface parameterization.

An important initial condition here for visualization is that the eigenvector is centered on the baseline design. For all of the problems in this dissertation, the baseline design is at $\mathbf{x} = \mathbf{0}$. Another important assumption is that only one eigenvector is being used to visualize deformations.

The deformation approach here can accordingly be generalized to remove these assumptions by specifying a centering coordinate $\mathbf{x}_0$, and using a multiple dimension subspace coordinate $\mathbf{y}$,

$$\mathcal{X} \xrightarrow{s(\mathbf{x}_0+\boldsymbol{U}^T\mathbf{y})} \mathcal{S}. \tag{4.19}$$

Combinations of the active subspace basis can be combined to bring the result back under a single deformation parameter,

$$\mathcal{X} \xrightarrow{s(\mathbf{x}_0+y'\mathbf{a}^\top\boldsymbol{U}^\top)} \mathcal{S}. \tag{4.20}$$

Where $\mathbf{a}$ is a vector specifying a linear combination of an active subspace vector, and y' is the scalar along which the deformation is actuated this collapsed space.

4.2.4 Methods Summary

The active subspace method allows the construction of surrogate models in a subspace with reduced dimension. If f varies primarily along the coordinates $\mathbf{y}$, expect that modeling over $\mathbf{y}$ using the response surface will yield a good approximation of the optimum of f over its domain $\mathcal{X}$. To approximate f using the active subspace accordingly, this dissertation employs the following procedure.

1. Choose M points of $\mathbf{x}_i$ according to a well-spread sampling strategy on the normalized input domain $\mathcal{X} = [-1, 1]^m$. Scale these to the model's natural inputs, then compute $f_i = f(\mathbf{x}_i)$ and $\nabla_{\mathbf{x}} f(\mathbf{x}_i)$.

2. Compute the eigendecomposition as in Equation 4.4, using the gradient data expressed in the normalized domain. Inspect the eigenvalue spectrum, use one of the heuristic described in Equation 4.16 to choose a subspace dimension n, and retain that many eigenvectors. These eigenvectors form the active subspace basis $\boldsymbol{U}$.

3. Project the sampled data into $\mathcal{Y}$ using the forward maps $\mathbf{y} = \boldsymbol{U}^\top \mathbf{x}$; $g(\mathbf{y}) = f(\mathbf{x})$; and $\nabla_{\mathbf{y}} g(\mathbf{y}) = \boldsymbol{U}^\top (\nabla_{\mathbf{x}} f(\mathbf{x}))$.

4. Construct a response surface $g^*(\mathbf{y}^*) = \mathcal{R}(\mathbf{y}^*; \mathbf{y}_i, g(\mathbf{y}_i))$ with the projected training data.

5. Use the response surface g^* to interrogate the design space, i.e. for minimizing the estimated function.

Many of the above steps are implemented in the open-source "Python Active-subspaces Utility Library" (PAUL), available on GitHub [103].

4.3 Numerical Experiments

The construction of active subspaces for various design problems will now be demonstrated.

4.3.1 Biparabolic Airfoil

In this experiment, the biparabolic design problem described in Chapter 2 is used. There are 20 design variables, where shape parameterizations are made using one freeform deformation box over the airfoil.

The parameterization is interesting because the upper and lower control points have influence over both the top and bottom of the airfoil. A major theme will be discovering if active subspaces can still find modes that would require the decoupling of the upper an lower surface. For example, in this two-dimensional problem the lower part of the airfoil is the only surface that effects the equivalent area.

As a two-dimensional physics problem, a rather large set of strongly converged samples was obtainable. There are two sets - a 962 sample training set, and a 3859 sample test set. These samples are those retained from a 1000, and 4000 sample set respectively, where samples with residuals greater than 1×10^{-10} were discarded.

4.3.1.1 Active Subspace Properties

What follows is the active subspace analysis on the full set of data of 4821 samples. The eigenvalue plots featured in Figure 4.1 preview a few phenomenon that will appear in later examples. In general it is important to identify gaps in the eigenvalues, as they partition the collections of important eigenvectors that describe the problem. The bootstrap ranges of the eigenvalues are shown by the polygonal patches around the eigenvalues.

One of the more favorable results that can be discovered in active subspace analysis is shown for the lift objective Figure 4.1c. A large gap, approximately two-orders of magnitude, is found after the first eigenvalue . This means that it might be possible to expect a significant amount of the problem to be described in one dimension. This correlates with a small heuristic measure of error that decays very slowly, shown in Figure 4.1d.

A similar result can be found for equivalent area in Figure 4.1e, where a large gap of about one order of magnitude appears after the first variable. Since the gap is smaller, this suggests that the next set of modes, up to perhaps the sixth, will be important to a second order effect. This is represented in the heuristic measure of error in Figure 4.1f with the order of magnitude decay available by increasing dimension. It suggests that a subspace dimension of nine will likely result in favorably accurate response surfaces.

Drag's eigenvalue analysis in Figure 4.1a shows much more behavior in the earlier modes. The first weak gap appears after the fifth dimension. As a result it could be expected that a large contribution to drag behavior exists among these variables. The next chunk of variables, 6-11, should also be examined for a second order influence. The heuristic for this objective also shows the large amount of behavior in the function by starting with a larger magnitude 3×10^{-1} that decays slowly with increasing dimension. In this case a decision on dimensionality has to be made by also considering the largest dimension that the surrogate modeling approach can comfortably handle. For this study ten dimensions will be treated as a safe upper bound. Though this is a highly problem-dependent decision. With more variability in the function, denser data is needed, and so only lower dimensions will be tractable.

The "sufficient summary plot" of each objective are shown in Figure 4.2. These plots are created by projecting the full-space coordinates of the samples into a chosen active subspace basis vector using the forward map in Equation 4.6. In the case of the following plots, the chosen basis is the first eigenvector. In these sufficient summary plots, one is looking for data to collapse into manifolds that could, for example, be used to build reasonably accurate surrogate models.

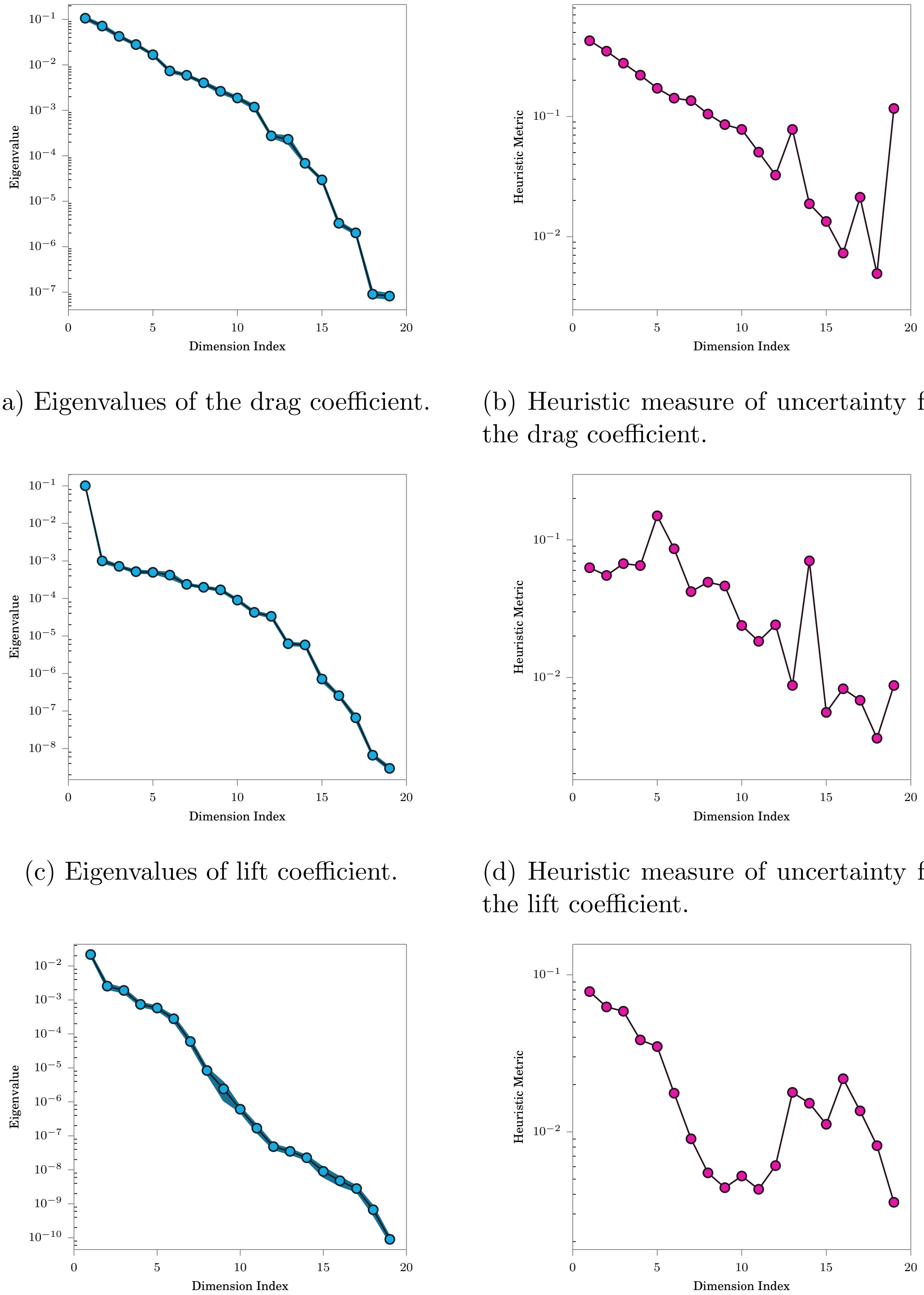
This is certainly the case for the lift sufficient summary plot in Figure 4.2b. It shows that all 4821 samples not only fall into a strongly collapsed manifold, but also shows that the trend is linear. The eigenvalue analysis for this problem, in having a strong gap between the first and second dimension, thus predicts the strongly collapsed manifold. The eigenvalue analysis however cannot predict the linear trend. This is just a fortunate result, and one that can be viewed in context to first principle aerodynamics. This will be discussed in more detail in Section 4.3.1.2.

Equivalent area shows a collapsed manifold in one dimension in Figure 4.2c. It has a larger spread however, which was indicated by the eigenvalue analysis. The global shape of the manifold is interpretable as quadratic. This is also a physically insightful result, recalling that the analytic result was quadratic in Equation 2.15.

The drag sufficient summary is much more spread by comparison in Figure 4.2a. This is because it is visualizing in one dimension what might be a function of five variables. The problem must be further explored to understand how the data is behaving. A first impression based on the lower manifold of the data is a quadratic function of parameters. This should not be surprising for drag, from a physical standpoint, given the quadratic result from first principle aerodynamics in Equation 4.22.

The next available analysis is to plot insightful deformations in Figure 4.3, that accompany the sufficient summary plots. In the sufficient summary plots, the vertical line is located at the active subspace coordinate that is used to generate the deformed design, as described in Section 4.2.3, and corresponds with the deformation plotted in blue. The baseline design is plotted in green. The airfoil thicknesses have been amplified for visualization.

For drag in Figure 4.3a, the fifth mode is plotted, which is the last mode of the first group of modes that is shown by the eigenvalue analysis. By looking at the shape that accompanies deformation along the fifth active subspace vector, it is observable that the analysis has identified what an engineer would describe as a thickness mode. This is a very interesting result because it matches with many physical intuitions.



(a) Eigenvalues of the drag coefficient.

(b) Heuristic measure of uncertainty for the drag coefficient.

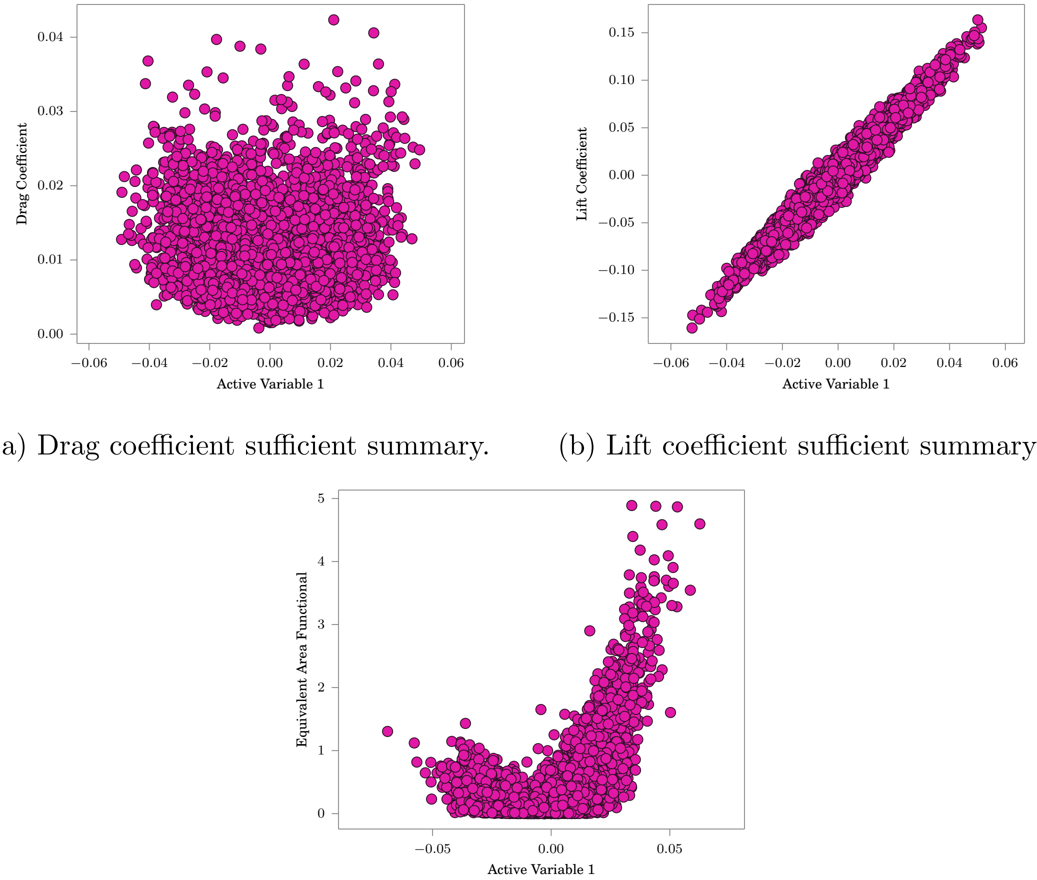
(c) Eigenvalues of lift coefficient.

(d) Heuristic measure of uncertainty for the lift coefficient.

(e) Eigenvalues of the equivalent area functional.

(f) Heuristic measure of uncertainty for the equivalent area functional.

Figure 4.1: Eigenvalue spectra of active subspaces for the biparabolic airfoil.

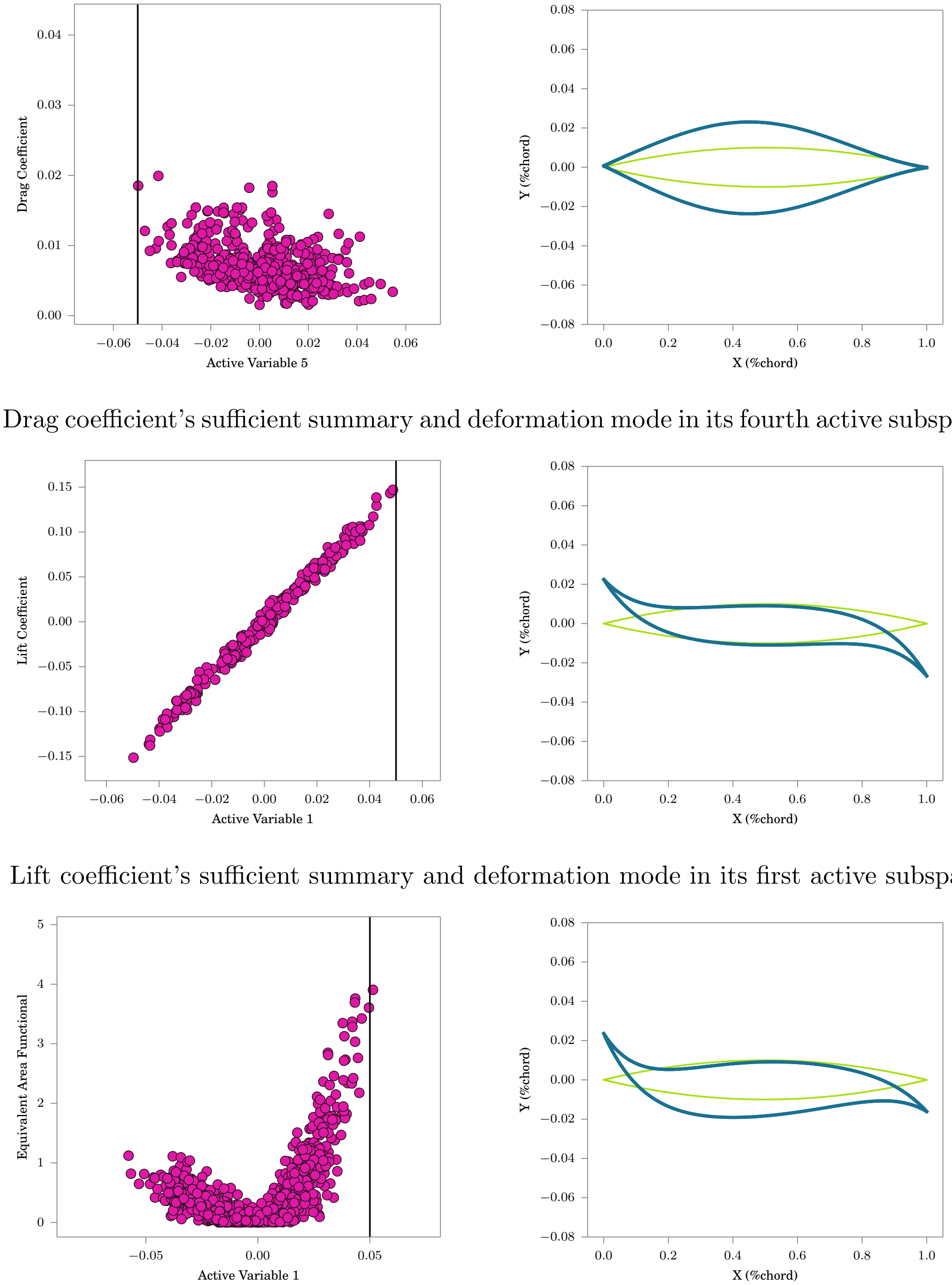


(a) Drag coefficient sufficient summary.

(b) Lift coefficient sufficient summary.

(c) Equivalent area functional sufficient summary.

Figure 4.2: Sufficient summary plots for the first active subspace of the biparabolic airfoil.



(a) Drag coefficient's sufficient summary and deformation mode in its fourth active subspace.

(b) Lift coefficient's sufficient summary and deformation mode in its first active subspace.

(c) Equivalent area functional's sufficient summary and deformation mode in its first active subspace.

Figure 4.3: Physically relevant deformation modes for the biparabolic airfoil.

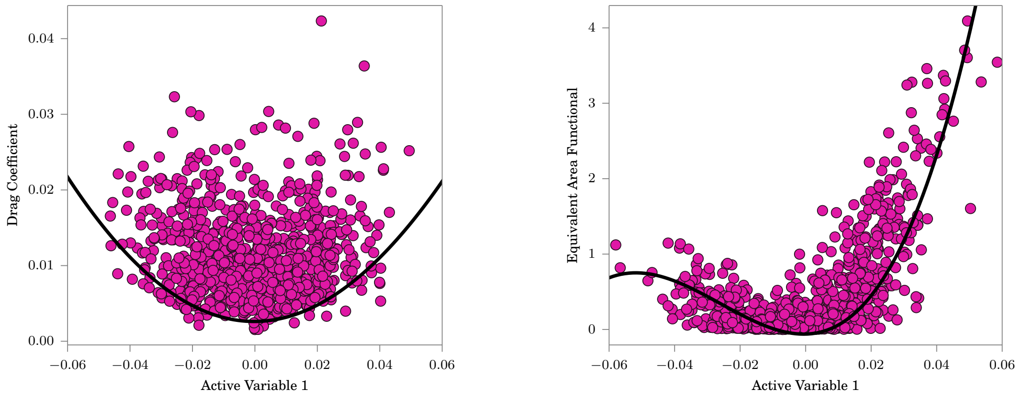
For lift in Figure 4.3b, the first mode is plotted, as this was alone was the strongest mode. The deformation in this mode can be interpreted as an angle of attack mode when acknowledging the definition of the angle of attack is based on the line drawn between the leading and trailing edge of the airfoil. An engineer would expect a rigid rotation however for an angle of attack mode. The behavior here may be explained here because the simulations involve nonlinear physics and a large range in deformations, which violates the linear supersonic potential theory.

For equivalent area in Figure 4.3c, the first mode is plotted. This shows some insightful behavior too. The upper part of the surface is largely left unmoved, reflecting that the upper part of the airfoil does not affect the physics of the problem in two dimensions. It was able to find this independent of the coupled parametrization of the problem, which requires both upper and lower control points to move in order to make a shape like this. The lower surface is expanded out, which reflects the physical intuition that volume makes a strong contribution to equivalent area changes. Additionally, the tips of the airfoil are deflected in an angle of attack mode, which reflects that other half of the physical intuition, that equivalent area is strongly affected by changes in lift.

Surrogate models can also be applied in these active subspaces. Figure 4.4 presents an example of surrogate model cross sections in the first active subspace for drag and equivalent area.

In the following experiments, the surrogate models will be evaluated by building an active subspace analysis of of the training set along (962 points), and then fitting a GPR surrogate model to the data for varying numbers of active subspace dimensions. These fits are performed without gradients. For each chosen dimension, the training error, the testing error, and the heuristic measure of uncertainty (Equation 4.16) were evaluated.

This is an interesting experiment on its own because it is a useful opportunity to examine the fitting power of a GPR model with varying dimension on a physically realistic problem. In order to do this sort of experiment otherwise, one would have to parameterized the problem for each draw of dimension, and run a new set of training and test samples. The active subspaces method is able short circuit a lot



(a) Surrogate model cross section in the first active subspace for drag.

(b) Surrogate model cross section in the first active subspace for equivalent area.

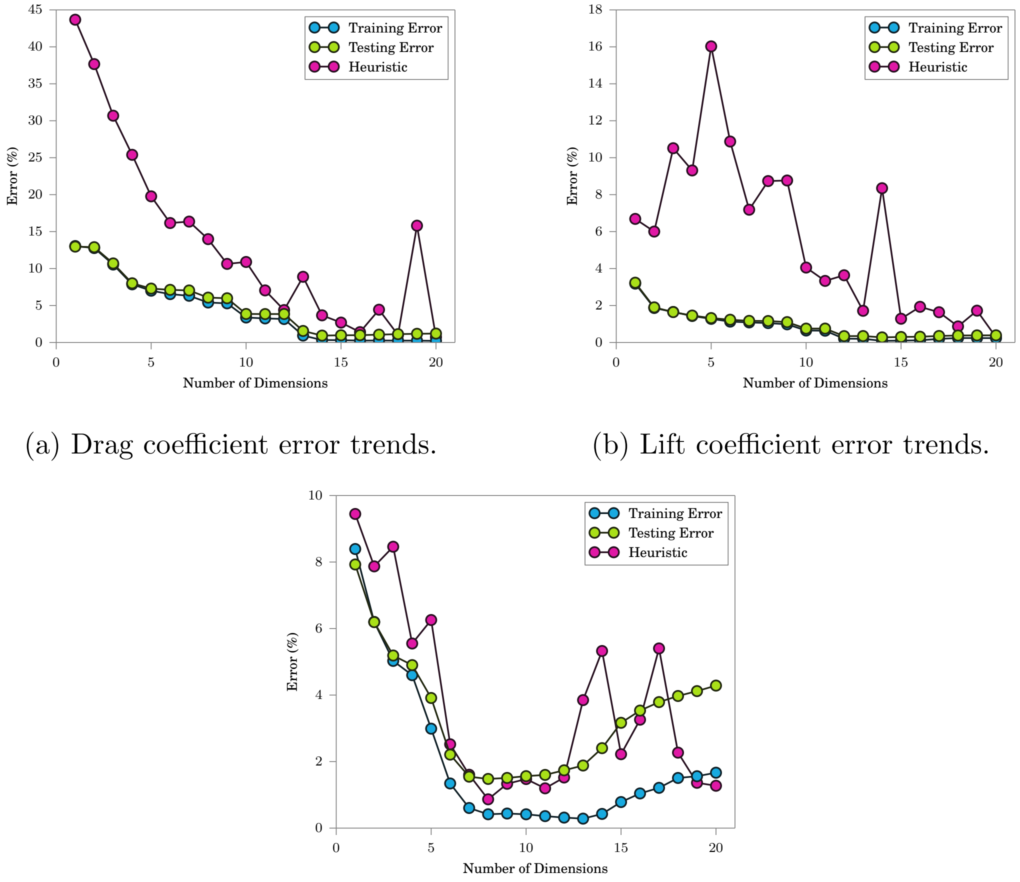
Figure 4.4: Cross sections of surrogate models plotted in the active subspace. The surrogates are plotted by displacing along the line directed by the subspace's eigenvector, starting at the fullspace's origin.

of this work. It should be noted however that there is now an additional effect contributing to modeling errors here, namely the error resulting from the lack of manifold collapsedness from mapping data into lower dimension bases.

The results of the experiment are presented in Figure 4.5. In general, as the dimension of the active subspace basis increases, it can be observed that the training and testing errors fall. This happens because the model has more dimensions over which to collapse the data into a surrogate model. As the dimension of the basis increases, the fitting power of the training data becomes thinned out, and eventually, the errors begin rising again.

This is the trade between the collapsing power of the active subspace method and the fitting power of the surrogate modeling method. In general this starts in the range of 10-15 dimensions.

The predictive nature of the heuristic measure of uncertainty can also be gauged from these results. Notable, the equivalent area heuristic tracks the testing error very well. This is a convenient but not expected result, as the heuristic is an upper bound of the error. In the case of drag the heuristic predicts the trend that the drag decreases across the the total dimensionality of the problem. The lift heuristic is less



(a) Drag coefficient error trends.

(b) Lift coefficient error trends.

(c) Equivalent area error trends.

Figure 4.5: Training and testing error trends for surrogates built in active subspaces of the biparabolic airfoil. The surrogates are generated using the first n active dimensions, indicated by the x-axis in the plots above.

predictive, but is at least bounding the testing and training errors in this case.

There are some characteristics in these error traces specific to the objective functions. The lift training and testing errors in Figure 4.5b start off low at 3.3% (90 coefficient counts) , and fall quickly with the addition of a few dimensions. The minimum testing error achieved by this problem is 0.3% (9 coefficient counts) with 14 variables. The testing error starts rising marginally after this point.

The equivalent area errors in Figure 4.5c fall in a similar fashion, starting at 8% (0.32 chord^2), and reaching a minimum of 1.7% (0.07 chord^2) with eight variables. The testing error rises quickly starting at 13 variables. This is reflective of how difficult it can be to fit the equivalent area function, which will be further encountered in Section 5.3.2.4. Since part of the airfoil does not effect the equivalent area function, parts of the design space are very flat. In order for a stationary surrogate model (like the one used in this thesis) to fit this type of behavior, a large amount of samples are needed in the flat region. Active subspaces make a significant contribution here by collapsing the behavior in those flat regions into fewer dimensions.

The drag errors in Figure 4.5a provide a little more detail on top of the eigenvalue analysis. It shows that the second variable does not contribute much to regressing the problem past the first variable. A large contribution to regressing the problem is made by using the first four variables. The fifth variable is the last variable that contributes to reducing the testing error before reaching a second plateau. From there, plateaus of two and three variables are needed to reduce the testing error. The minimum testing error is found at 1.1% (4.5 coefficient counts) with 14 variables.

4.3.1.2 Connection to Fundamental Aerodynamics

Thin airfoil theory suggests that the lift coefficient of a thin supersonic airfoil is a linear function on the angle of attack, for a given mach number.

$$C_l = \frac{4}{\sqrt{M_\infty - 1}}\alpha \tag{4.21}$$

As seen previously, there is a strong linear function present in the first active subspace for lift. To discover a connection it is important to work on the dimensionalized active subspace, by unscaling the normalization on the training data performed during the eigenvector analysis, but maintaining the same data centering. Since the experiment was defined as centered on the baseline design, no special treatment is needed here.

Next, an important hypothesis is needed, which supposes that this coordinate is in fact angle of attack. Thus, plotting Equation 4.21 over the sufficient summary plot, it is possible to discover the following relationship, shown in Figure 4.6.

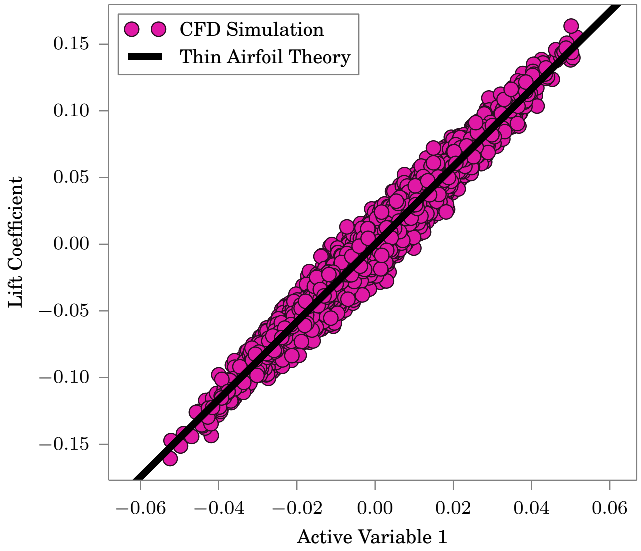


Figure 4.6: Comparison of thin airfoil theory for lift with the first direction's sufficient summary plot.

This plot shows that the active subspace analysis for the biparabolic airfoil was able to discover a fundamental aerodynamic behavior, and also identify it as the dominant factor for this design problem. This is a well known behavior among the aerospace community. The significance of this connection is that it was discovered by a machine learning based heuristic.

Now recall the thin biparabolic airfoil result for drag when parameterized by thickness and angle of attack.

$$C_d = \frac{4\left(\frac{4}{3}\tau^2 + \alpha^2\right)}{\sqrt{M_\infty^2 - 1}} \tag{4.22}$$

To evaluate the relation between thin airfoil theory and the active subspace analysis for drag, additional work is needed. In the case of this analysis, the angle of attack mode is buried within the first four modes. It can be recovered in the present case by combining the four basis vectors with this sub-map:

$$y^* = [-1, -1, +1, +1]\,\mathbf{y} \tag{4.23}$$

The resulting deformation mode is found in Figure 4.7, and the sufficient summary plot is compared to the thin airfoil theory result in Figure 4.8a, assuming that the active subspace coordinate is actually the angle of attack variable.

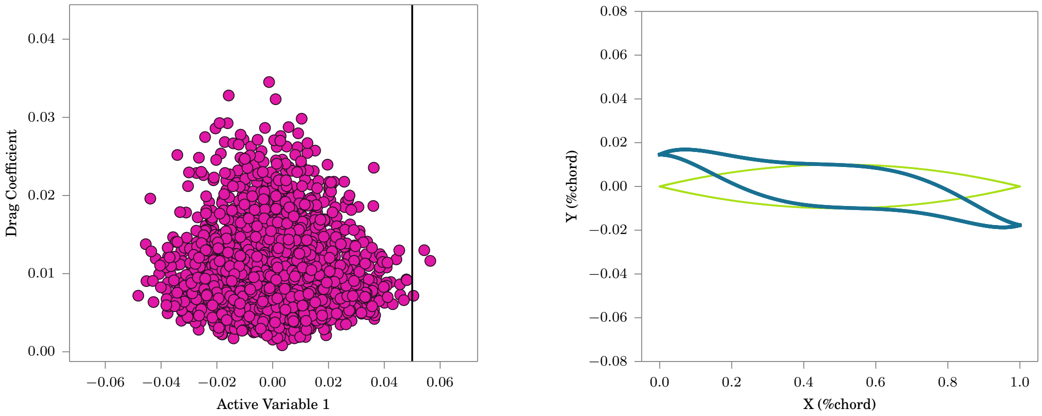
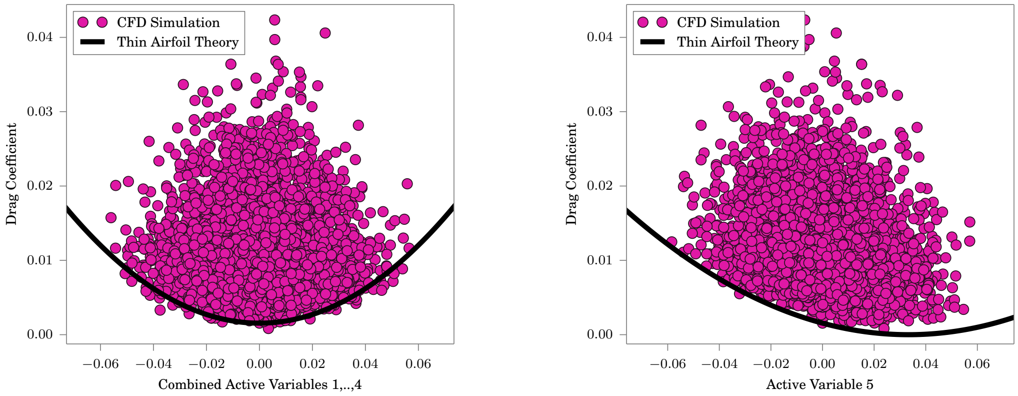


Figure 4.7: The first four drag active subspaces combined to yield an angle of attack mode.



(a) Thin airfoil theory for drag dependence on angle of attack compared to the combined first through fourth active subspaces.

(b) Thin airfoil theory for drag dependence on thickness compared to the fifth active subspace basis.

Figure 4.8: Comparison of thin airfoil theory with biparabolic airfoil active subspaces.

For thickness, the active subspace coordinate system needs to be scaled and shifted to match the thickness that results along this direction. In this case, at $y_0 = 0.0$ there is a thickness of 2%, and at $y_0 = 0.06$ there is a thickness of 5.4%. Scaling and shifting the input to thin airfoil theory accordingly, the resulting comparison to thin airfoil theory is presented in Figure 4.8b.

While not parameterized by the particular thin airfoil theory presented in this thesis, camber is also an important mode for drag. The camber mode can be constructed in a similar way as the angle of attack mode, by using the following sub-map:

$$y^* = [-1, -1, -1, -1]\,\mathbf{y}. \tag{4.24}$$

The resulting deformation mode is shown in Figure 4.9.

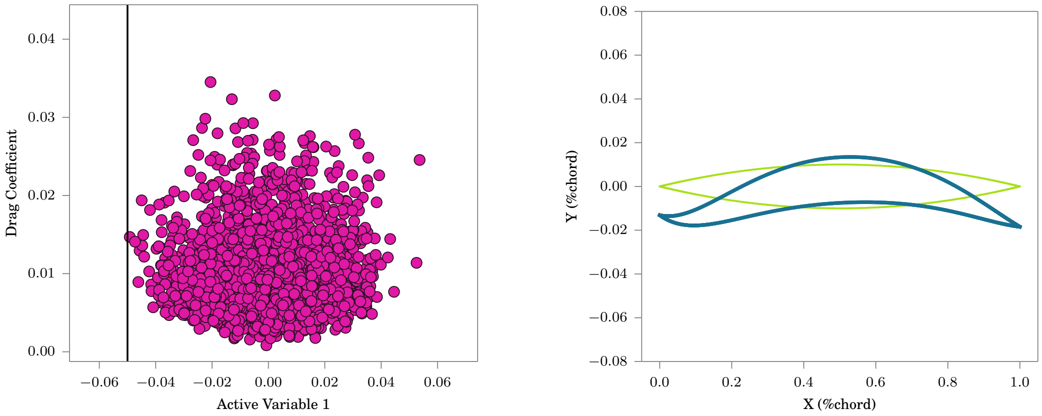


Figure 4.9: The first four drag active subspaces combined to yield a camber mode.

4.3.2 Langley Supersonic Business Jet

In this next problem, the active subspaces of the simplified Langley supersonic business jet are examined. Like the biparabolic problem, the quantities of interest here are lift coefficient, drag coefficient, and the equivalent area functional.

This problem is more expensive as it is a three-dimensional problem. Samples take 110 minutes on 54 cores to collect objectives and three gradients. There are 192 design variables for this problem, but only 172 samples were collected after filtering for results that converged more than 1×10^{-7}. It will be shown that this was still enough data to perform an active subspace analysis that results in strongly collapsed data in one dimension.

4.3.2.1 Active Subspace Properties

Looking at the eigenvalue analysis, interesting results appear. All three objectives show large gaps after the first eigenvalue, which suggests that the first dimension has a strong influence on the objectives.

The drag eigenvalues, shown in Figure 4.10a, have a single order of magnitude primary gap. The second and third active variables probably are important since they have sizable gaps as well.

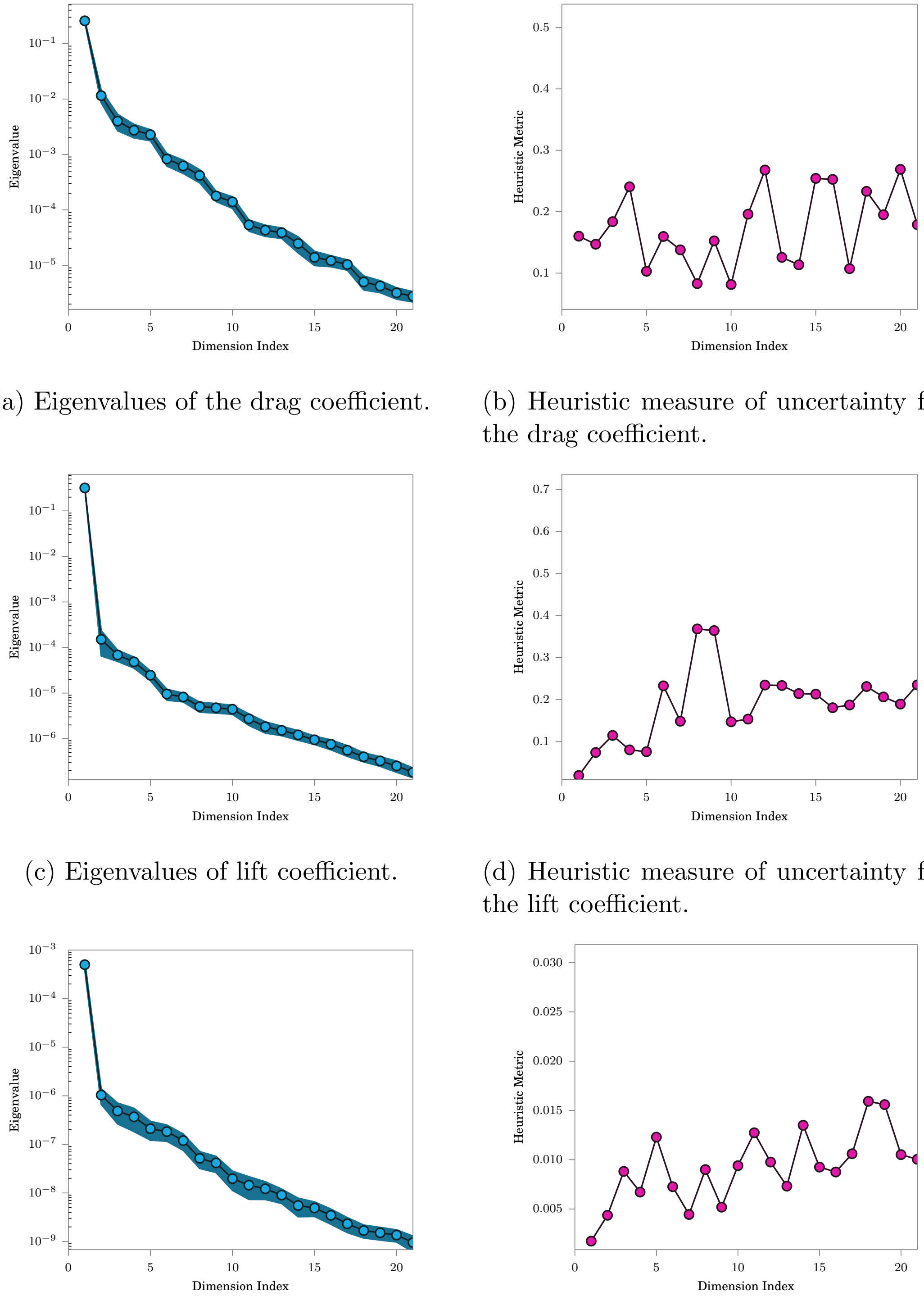
The lift eigenvalues, shown Figure 4.10c, have a three order of magnitude gap, so one might expect an especially well collapsed manifold in the first dimension. Then there is a cluster of four variables that could have a higher order effect.

Equivalent area's eigenvalues shown Figure 4.10e, like drag, have a large gap that levels off into a cluster with variables two through four.

All of this analysis is heuristic, and meant to be a preview of what may appear in sufficient summary plots and deformation mode visualizations.

Many of the trends suggested by the eigenvalues appear in the sufficient summary plots. Figure 4.11 shows the sufficient summaries for drag, lift and equivalent area within the first active subspace dimension, compared with the deformation that is associated with subspace. As before, the vertical line is plotted at the scalar factor by which the basis vector is scaled to generate the deformation mode plotted.

What is found is that indeed all three quantities of interest can be approximately described in one dimension. In the case of drag and lift, the behavior is nearly linear. In the case of equivalent area, the behavior is quadratic, with a few handfuls of outliers.



(a) Eigenvalues of the drag coefficient.

(b) Heuristic measure of uncertainty for the drag coefficient.

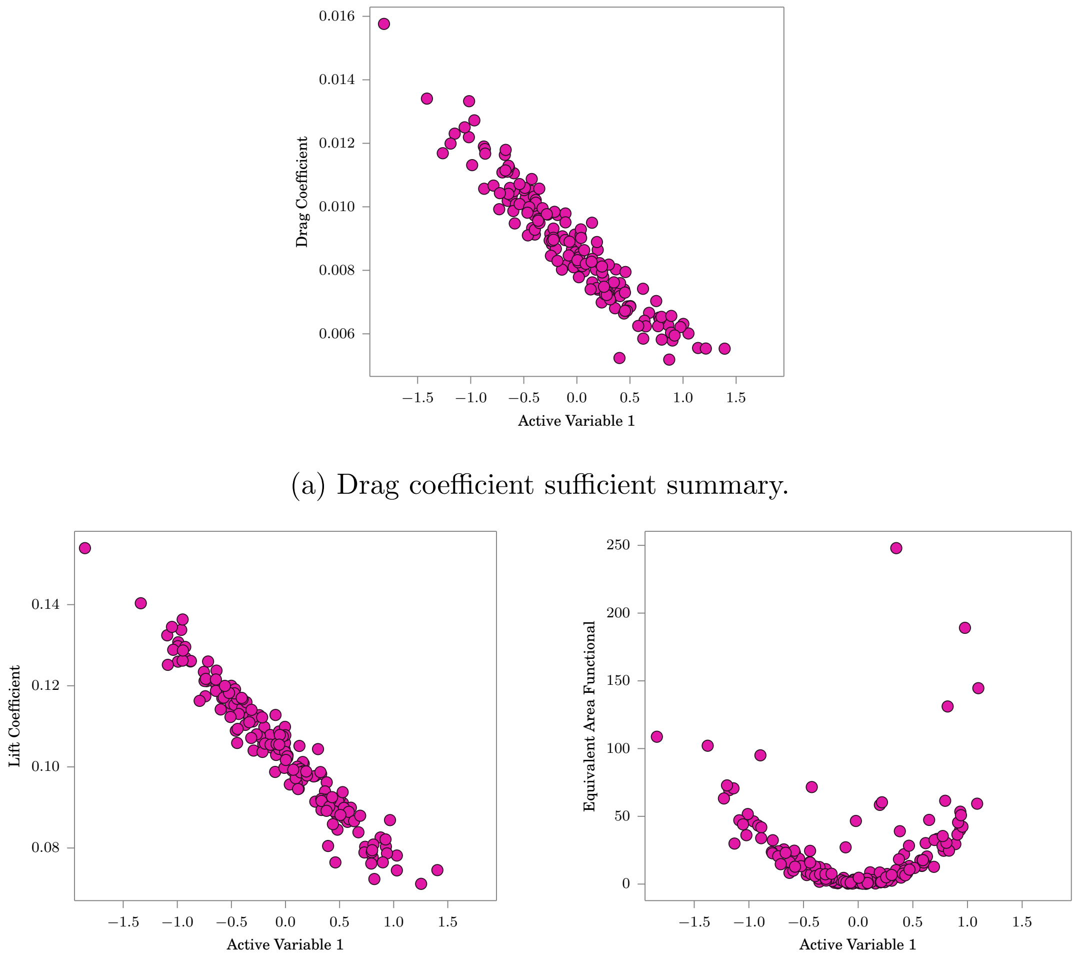
(c) Eigenvalues of lift coefficient.

(d) Heuristic measure of uncertainty for the lift coefficient.

(e) Eigenvalues of the equivalent area functional.

(f) Heuristic measure of uncertainty for the equivalent area functional.

Figure 4.10: Eigenvalue spectra for the langley supersonic business jet. Only the first 20 of 192 eigenvalues are displayed.



(a) Drag coefficient sufficient summary.

(b) Lift coefficient sufficient summary.

(c) Equivalent area functional sufficient summary.

Figure 4.11: Sufficient summary plots for the first active subspace of the langley supersonic business jet.

4.3.2.2 Physical Insights

The modes associated with these active subspace bases are plotted in Figure 4.12. With only minor variation, the modes are primarily angle of attack modes along the main wing. The fuselage and tail bend to permit the rotation of the main wing and body approximately around the moment reference center. Physically, this is consistent with first order aerodynamics. These samples were taken at a cruise angle of attack, so the drag behavior is strongly driven by the larger quadratic term here, compared to the thickness term. Lift as described earlier is dependent on angle of attack. And equivalent area is very much tied to lift, so the angle of attack mode is a natural result.

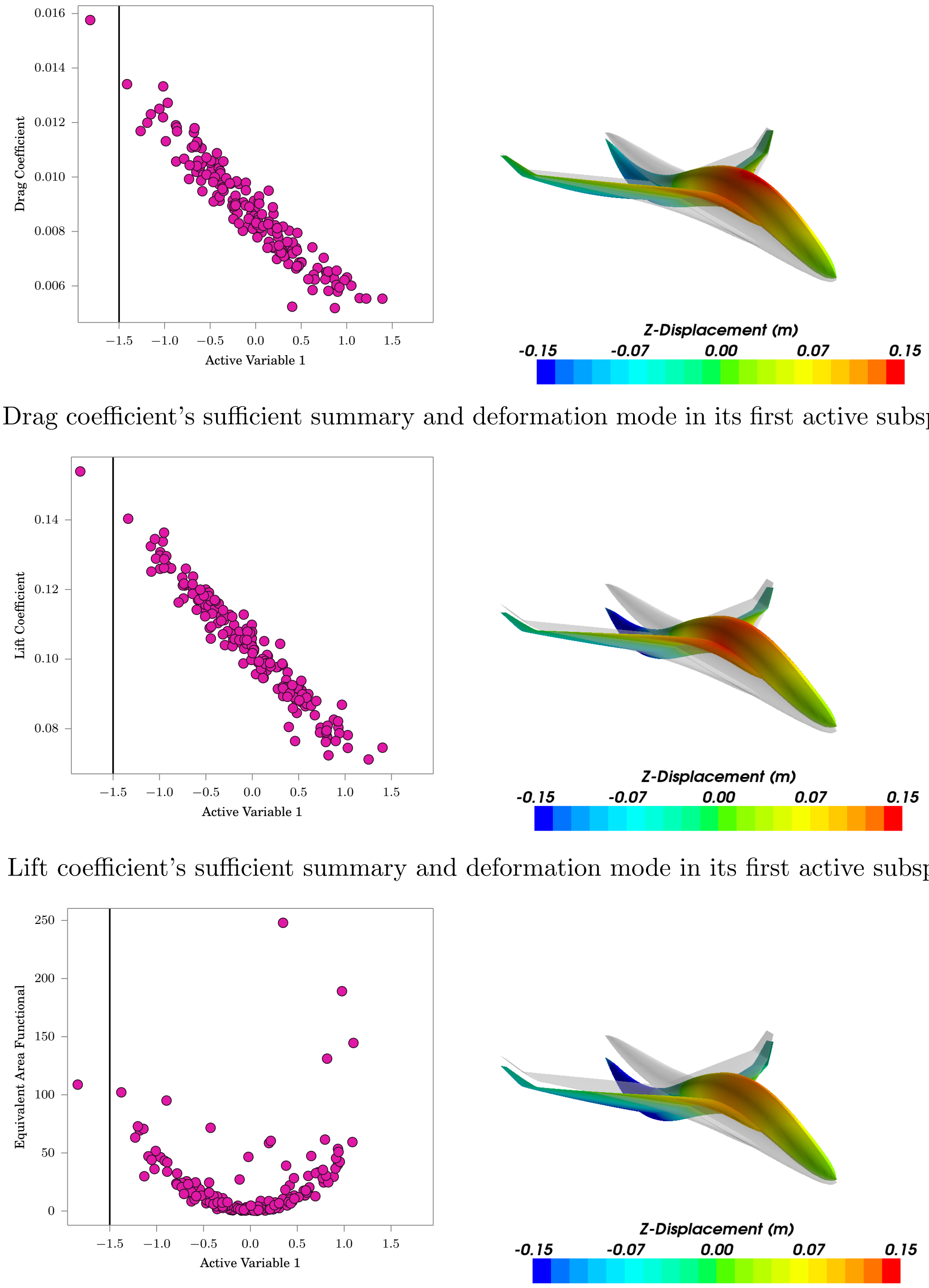
The coherence of these results means that the first order behavior of all three quantities of interest can be described by nearly the same variable, and the corresponding deformation mode that accompanies it. This would at first seem like an exciting result in the context of surrogate modeling, but under the context of constrained optimization it creates an over-constrained problem. For example, in setting a minimum lift, a minimum drag is prescribed. Or in setting a target equivalent area of nearly zero, it constricts a narrow range of shapes to explore, hardly enough room to make a significant improvement on drag. So the next level of active subspace modes are needed to approach an optimization problem here.

Several physically interesting higher order modes are presented in Figure 4.13.

First, the lift coefficient mode on the fourth active subspace dimension is shown in Figure 4.13a. This is a dihedral mode, with a combination of wing tip downwash.

In another example in Figure 4.13b, the equivalent area mode shows that a combination of fuselage expansion and wing wash-in increases equivalent area in the third active subspace dimension.

These types of deformations display characteristics of several engineering-like parameterizations commonly phenomenon constructed in supersonic aircraft design. This provides an important component of verification for the usefulness of active subspaces in design.

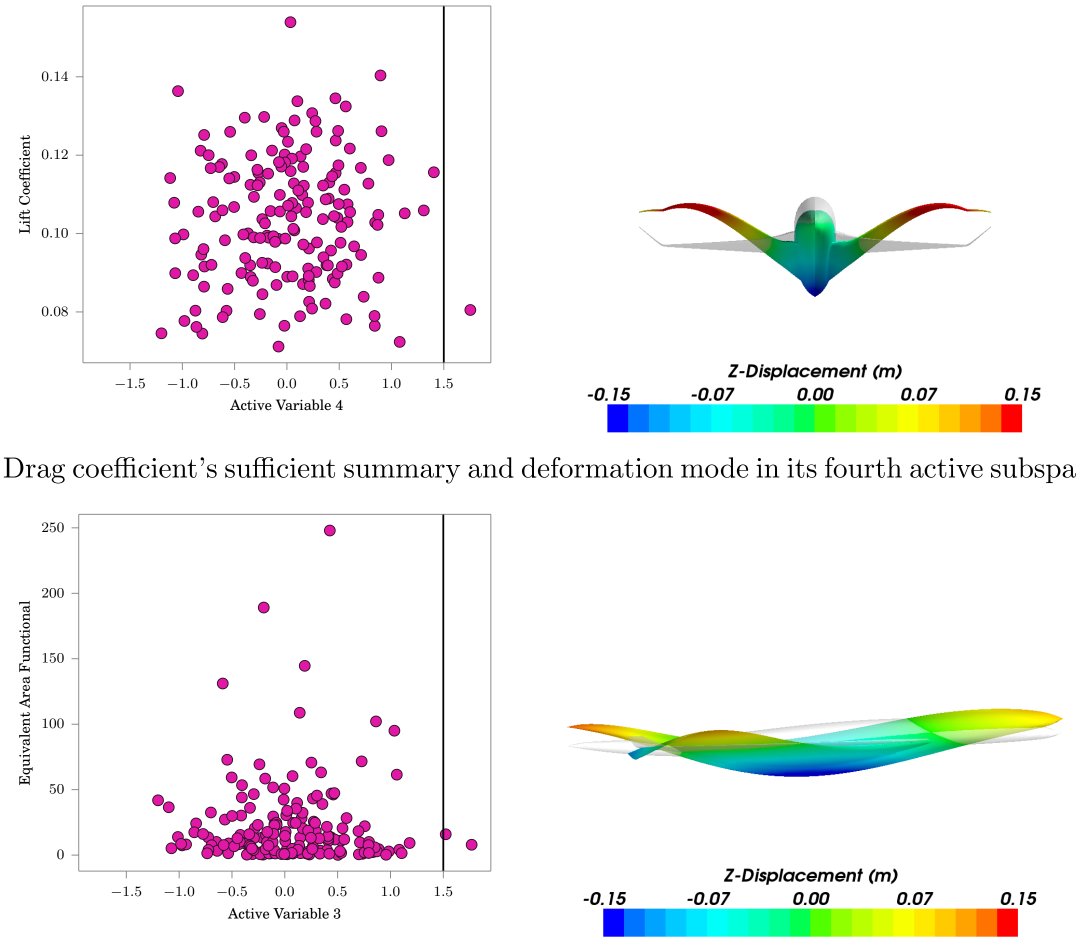


(a) Drag coefficient's sufficient summary and deformation mode in its first active subspace.

(b) Lift coefficient's sufficient summary and deformation mode in its first active subspace.

(c) Equivalent area functional's sufficient summary and deformation mode in its first active subspace.

Figure 4.12: First deformation modes for the Langley supersonic business jet discovered by an active subspace analysis.



(a) Drag coefficient's sufficient summary and deformation mode in its fourth active subspace.

(b) Lift coefficient's sufficient summary and deformation mode in its third active subspace.

Figure 4.13: Physically relevant deformation modes for the Langley supersonic business jet discovered by an active subspace analysis. The vertical line in the sufficient summary plots indicate the location in the active space that corresponds with the deformation.

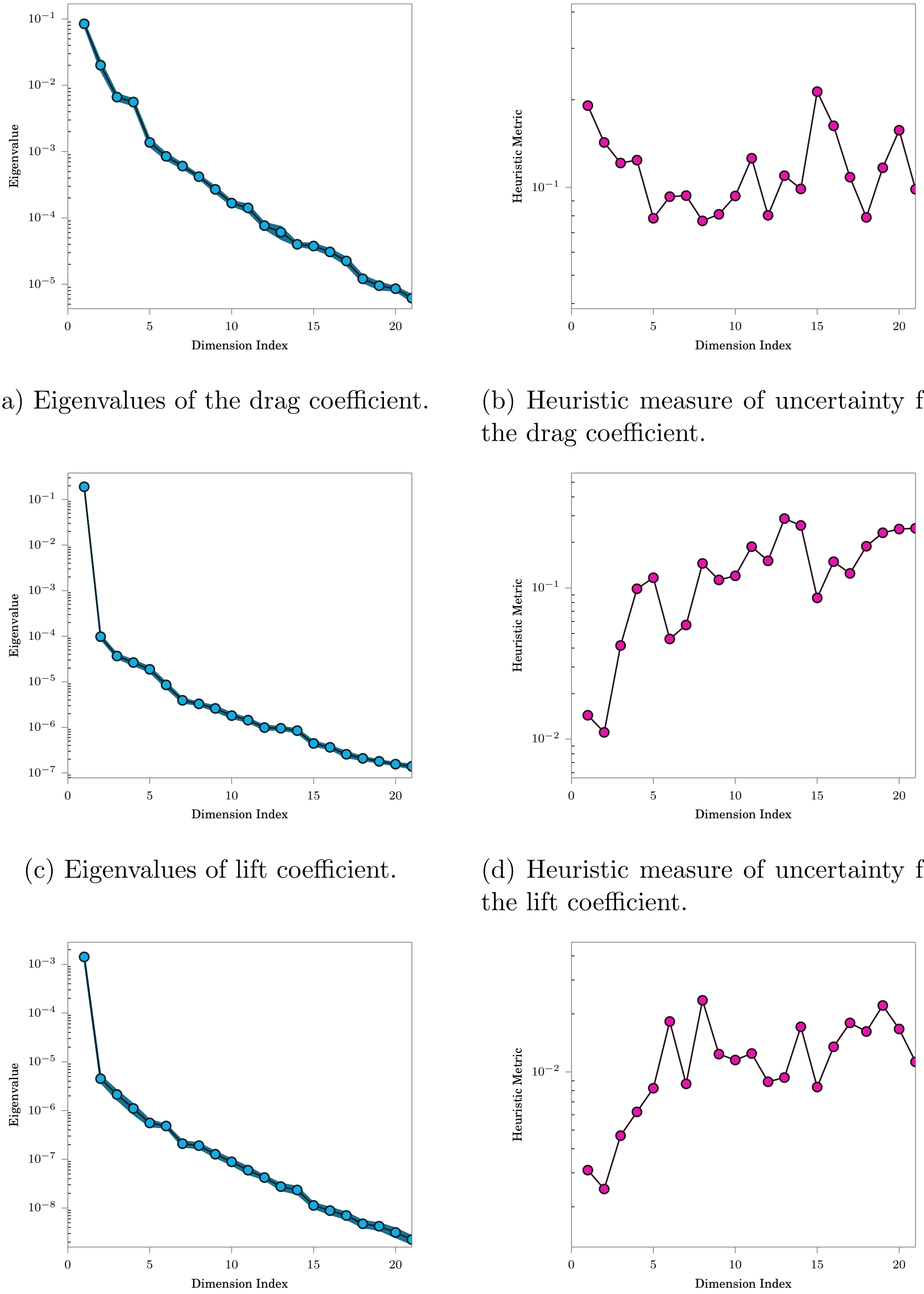
4.3.3 N+2 Supersonic Passenger Jet

This model is the most complex problem encountered in this dissertation. In terms of physics it is busy. Wings, tails, engine nacelles, and engine ingestion and exhaust are present. Samples of this design require 120 minutes on 96 cores. There were 315 sampled designs, 305 of which were retained after filtering for results that converged more than 1×10^{-6}. As described in Section 2.5.3, there are 105 variables here spread across the wing, tail, fuselage, and aft deck. Unlike the Langley business jet, parts of the design are frozen, such as the engine location and shape. So one large box over the whole design is not possible. As will be shown, despite this complexity, there are still well collapsed sufficient summary plots and interesting deformation modes to be discovered.

4.3.3.1 Active Subspace Properties

Following the active subspace analysis method, the gradients of the 305 samples are processed to investigate their eigenvalue spectra. The results are shown in Figure 4.14 below. Again there are large gaps after the first eigenvalue for lift and equivalent area. Lift sees a secondary gap after the fifth eigenvalue, so this could be worth looking at for second order behavior. The decay for drag is not as aggressive, though relative to the rest of the spectrum, the first mode can be considered very important, followed closely by the second and third modes. It would also be reasonable to group in the fourth mode with a second-order analysis.

Plotted are the sufficient summary plots for the first active subspaces of the N+2's quantities of interest in Figure 4.15. As seen in many of the other aerodynamic problems, there are strongly collapsed manifolds for lift and equivalent area. Again, the lift coefficient emits a linear trend, and the equivalent area emits a quadratic trend. The drag sufficient summary does not collapse in one dimension, as expected from the eigenvalue plot. Looking at the surrogate model's cross section in this space will reveal that there is a moderately quadratic trend present.



(a) Eigenvalues of the drag coefficient.

(b) Heuristic measure of uncertainty for the drag coefficient.

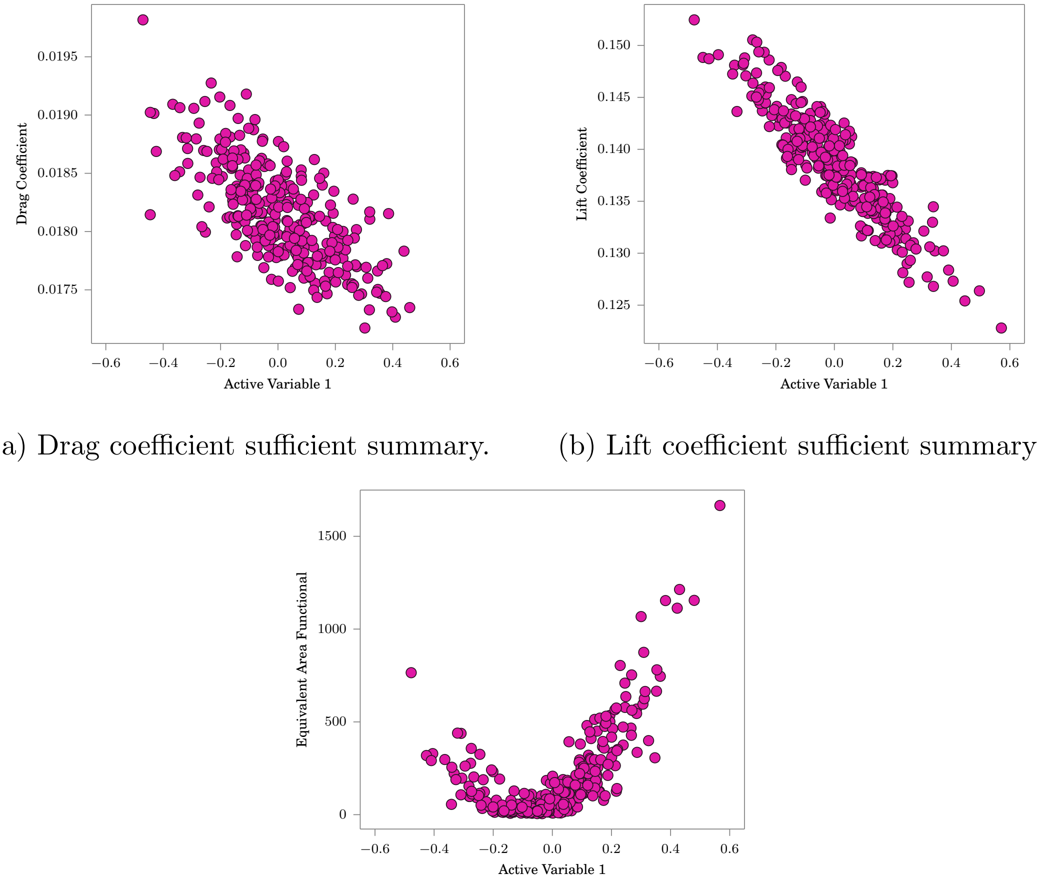
(c) Eigenvalues of lift coefficient.

(d) Heuristic measure of uncertainty for the lift coefficient.

(e) Eigenvalues of the equivalent area functional.

(f) Heuristic measure of uncertainty for the equivalent area functional..

Figure 4.14: Eigenvalue spectra for the N+2 supersonic passenger jet.



(a) Drag coefficient sufficient summary.

(b) Lift coefficient sufficient summary.

(c) Equivalent area functional sufficient summary.

Figure 4.15: Sufficient summary plots for the first active subspace of the N+2 supersonic passenger jet.

4.3.3.2 Physical Insights

The deformation modes for this geometry in the first active subspace of drag, lift and equivalent area are shown in Figure 4.16. Since the vehicle is parameterized with boxes that control only selected components - the main wing, tail, aft deck, and fuselage - the deformations found by the active subspace analysis only are permitted around these surfaces. This is an important consideration because it shows the parameter dependence of the active subspace analysis.

In the deformation modes, the plots are colored by displacement in the z-direction. The orange and red colors indicate that part of the wing is rising, and the blue colors indicate that part of the wing is lowering. The baseline geometry is plotted as well in transparent gray.

One of the first behaviors that can be recognized is on the main wing. In the case of lift, shown in Figure 4.16b, as the active subspace shows a large scale increase in lift coefficient, the leading edge of the wing is rising, and the trailing edge of the wing is falling. This indicates that the wing is twisting to increase incidence with on-coming flow. Wing twist is an important engineering-like shape change that is recognized in aerospace design. Here, the active subspace method is able to detect it automatically.

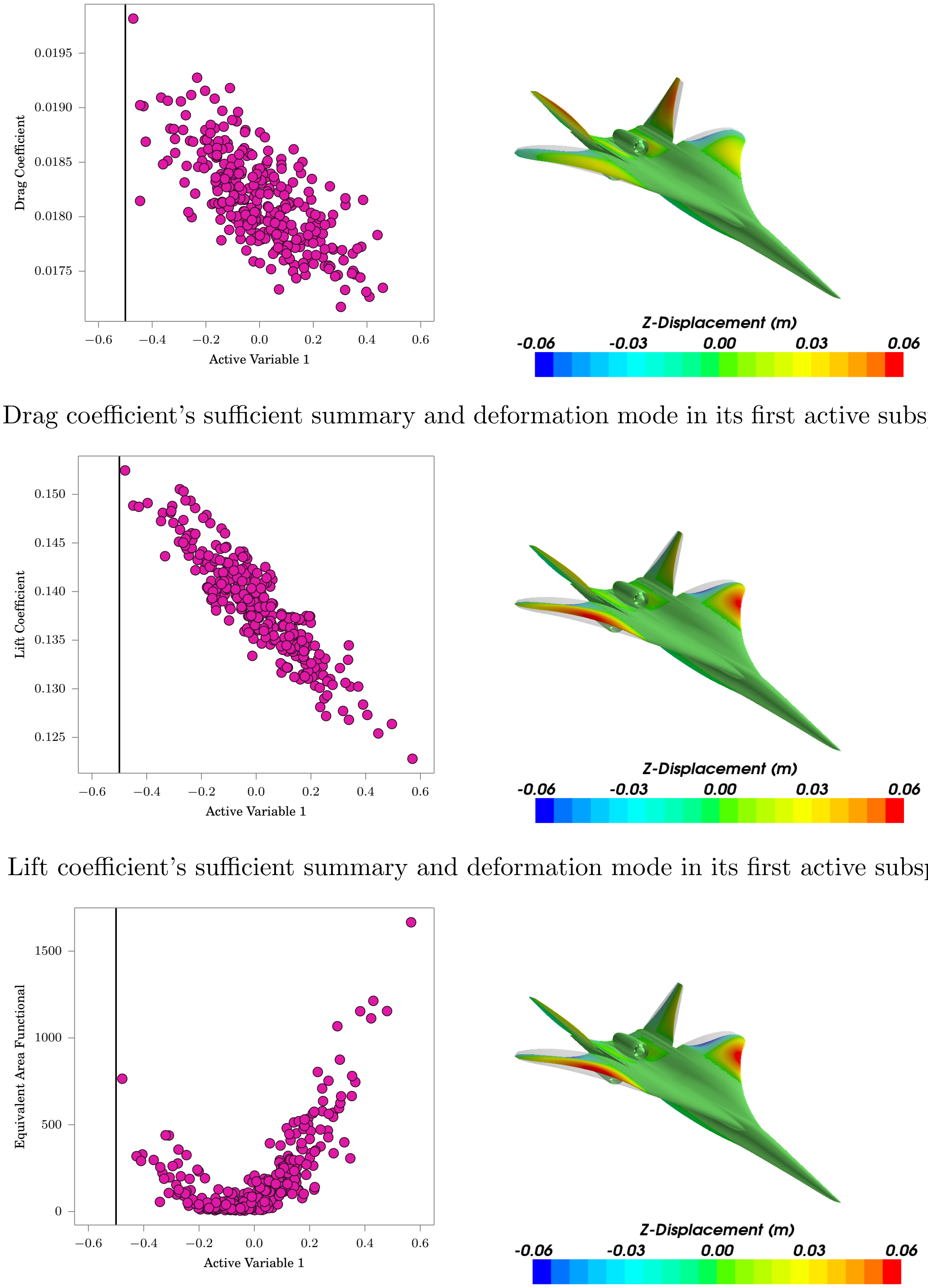
Twist is also present in the tail. It can be observed by seeing the leading edge of the tail brought inboard compared to the baseline surface. The other surfaces (aft deck and fuselage) do not have as much deformation in this mode.

This result is contrasted with the results of the Langley business jet. In the case of the Langley example, a parameterization was built around the entire vehicle, enabling the discovery of fuselage twisting and cambering modes, in addition to wing twisting. While the N+2 passenger jet in this example does not include such an encompassing parameterization, it is still possible to discover similar elements of the lift mode from the Langley example, within the context of the wing component alone.

Similar results for wing twist are found for the equivalent area functional, shown in Figure 4.16c. Since equivalent area takes important contributions from volume and lift, it is reasonable to see a wing twisting mode similar to the lift deformation mode.

In the case of the drag deformation mode shown in Figure 4.16a, the deformation mode is comprised of two engineering-like shape changes. The first is a twisting mode, actuated by lowering the trailing edge. The second is a cambering mode, where the front quarter chord of the wing is raised slightly, while the leading edge is left unmoved. The combination of these behaviors reinforces the applicability of the active subspace analysis. The eigenvalue spectrum of drag shows a much more rich set of important modes for this objective. Furthermore, as discussed in the analysis of the biparabolic example, there are many shape parameterizations that are expected to be important for a lifting surface in supersonic flow.

A major result of this experiment says that despite the different components of the geometry being segregated into various sub-parameterizations with individual FFD boxes, the active subspace method was still able to identify intuitive design-oriented results. This demonstrates the usefulness of the method in complex geometry with complex parameterizations.



(a) Drag coefficient's sufficient summary and deformation mode in its first active subspace.

(b) Lift coefficient's sufficient summary and deformation mode in its first active subspace.

(c) Equivalent area functional's sufficient summary and deformation mode in its first active subspace.

Figure 4.16: First deformation modes for the N+2 supersonic passenger jet.

4.4 Summary

This chapter presented a method for reducing the dimensionality of high-dimensional aerospace problems built upon the active subspace method. This involved identifying and retaining a subset eigenvectors from an eigenvalue analysis of the estimated covariance matrix based on samples of the problem's gradients. Several heuristics were suggested to enable the selection of a reasonable number of active subspace dimensions. The training error of surrogate models built in active subspaces of varying dimension were investigated.

A core result of this chapter is the identification of the existence of active subspaces in aerospace design problems. In addition, several coherent features were present. Across the two- and three-dimensional supersonic design problems presented in this chapter, all three were shown to have collapsed linear behaviors for lift in one dimension, described with an angle of incidence mode on the lifting surfaces. Across all three problems, equivalent area was well described in one dimension and had a predominantly quadratic behavior. Across all three problems, the behavior of drag coefficient required several active subspace dimensions, anywhere between five and ten. In the case of the three dimensional problems, both showed a relationship between drag and angle of incidence on the lifting surfaces.

The coherence of these results supports the viability of the active subspace method in aerospace design problems. Despite building very high parameterizations of the problem, there exist reduced spaces in which a surrogate model could be applied for optimization. Applying these models for optimization is the subject of the next chapter.

Chapter 5

Optimization

This chapter will combine the two themes of surrogate modeling and active subspaces to enable efficient global optimization of high-dimensional high-fidelity flow simulations.

5.1 Background

A short review of optimization approaches and an closer look of existing work around efficient global optimization will be presented.

Local optimization is a category of optimizers that start with an initial candidate solution an the iteratively tries nearby solutions under chosen heuristics until a favorable result is reached. Local optimization typically asks for gradients from the analysis to accelerate the search heuristic, and can be very efficient in large dimension. This comes at the expense of not being able to search the global design space, making them liable to get stuck in a sub-optimal basin.

An important category of local optimizers use line searches to propagate this search. One of the most well known line search methods are Sequential Quadratic Programming (SQP) methods [104, 105]. The concept of sequential approximation appears in other methods such as Sequential Linear Programming [106], and the CONvex LINearization method [107, 108]. Another category of known as trust region methods, where reduced order surrogate models are applied in a region that is expanded or contracated according to the accuracy of the fit [109, 110]. Gradient-free approaches for local optimization are available when gradients are not appropriate or the response is non-smooth [111].

Global optimization differs from local optimization in that it seeks a result that is optimum in the entire input space of the problem, typically within some pre-defined global bounds on the input parameters. Thus they address the parasitic case that local optimization encounters when initialized in the wrong location. In order for an algorithm of this category to converge, a theorem by Torn and Zikinskas [112] suggests that the search must be "dense", or that it must essentially search every point in the domain. In practical implementation, the result of this theorem says that global optimizers must have a feature that encourage it to explore unattended parts of the design space.

Major categories of global optimization include genetic algorithms [113], direct search methods [114], and particle swarm algorithms [115]. They are robust to all manner of behavior in the design space, but require dense samples of analyses, making them inefficient or intractable when applied to high fidelity design problems.

Because surrogate models are inexpensive to evaluate, they can be used to identify estimates of the optimum, for example by using the above global optimization methods. The coupling of surrogate models and optimization is called Surrogate Based Optimization (SBO).

One of the powerful approaches in SBO is to use probabilistic models to construct Infill Sampling Criterion (ISC) that enable efficient search heuristics. Jones provides an excellent review of the various surrogate modeling techniques in the context of SBO, providing numerical examples to identify their different strengths and weaknesses [51]. Several criterion have been explored in literature to efficiently refine surrogates around potential optima. One such approach is to choose input locations that maximize the probability of improvement [116, 117]. Another is to choose locations that maximize the expected improvement [118]. The later was especially well exposed by Jones in the context of optimization [119]. These infill sampling criterion have been shown globally convergent because they have a feature that forces them to exercise parts of the domain that have been unexplored [112, 51, 120]. Constraints are easily expressed as a chain of statistically independent probabilities, described in literature [121, 122] and later in this dissertation.

The above methods are part of a category of efficient global optimization known as two-stage methods. First the response is sampled and fitted with a surrogate, and second the response is interrogated with the infill sampling criterion. Two stage methods suffer in the case that the initial sample is sparse or provides a highly misleading view of the function. In this case, the progression of samples converges extremely slowly or not at all. There is on going work on a category of 'one-stage' methods that avoid these pitfalls [51].

5.2 Method Formulation

5.2.1 Surrogate Based Optimization

As part of this dissertation, a surrogate based optimization approach was developed that follows the basic process flow shown in Figure 5.1. A summary of this process is described here, and detail on modifications to the traditional process are given in the following sections.

The process starts by choosing an initial set of design configurations with Latin Hypercube Sampling, which are evaluated using the SU2 simulation suite. This involves deforming the baseline mesh to the desired configuration, performing the direct solution to solve for the objective and constraint function values, and then performing multiple adjoint solutions to solve for the objective and constraint gradients. The data is given to the response surface modeler, which must tune a set of hyperparameters for both the objective and constraint surfaces. This is the first of two internal optimization problems required by the SBO. Two sets of convergence monitors are then checked. The first is based on the expected improvement, which monitors the efficient global optimization of the response surface surface. The second is based on the estimated optimum, which attempts to refine the response surface's final estimate of the optimum. Depending on which criteria is converged, adaptive refinement is then performed by choosing a sampling criteria function and solving for its minimum using a combined genetic algorithm and gradient based optimizer. This is the second internal optimization problem. Once both convergence monitors reach a specified tolerance, the process ends as converged.

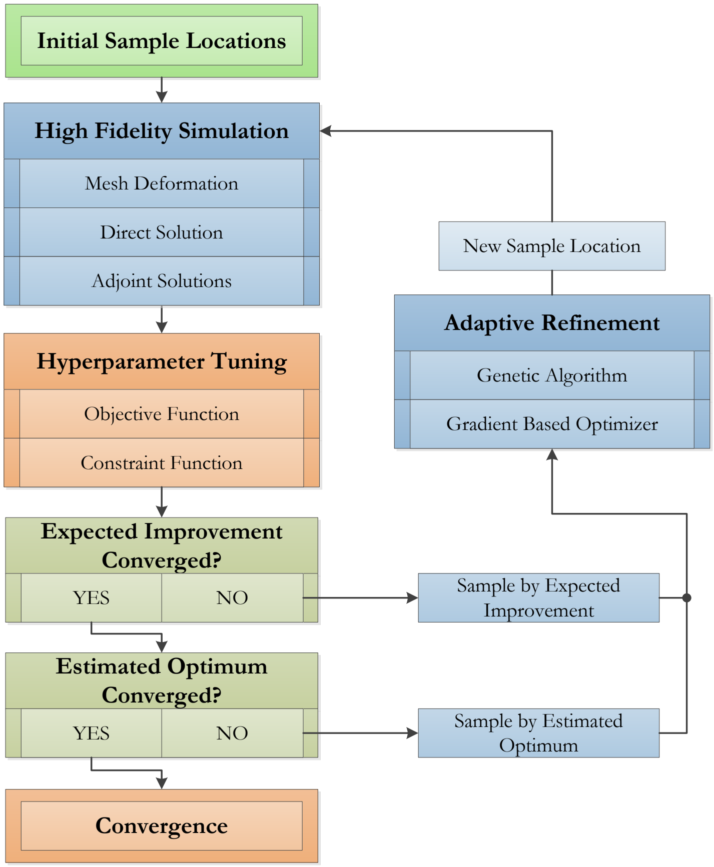


Figure 5.1: Surrogate based optimization process.

This approach to surrogate based optimization involves three sampling stages - first an initial sample, second an efficient global refinement, and last a refinement of the estimated minimum. These stages are designed to efficiently dissect the design space and attempts to yield the best estimate of the optimum only in the last sample. The main motivation of this construction is to build in a sense of convergence that is similar to that of gradient-based optimization.

First, the design space is sampled with a set of well-spread initial design points chosen with Latin Hypercube (LHC) sampling restricted to the hypercube,

$$lb_i < x_i < ub_i \ , \ i = 1, ..., D. \tag{5.1}$$

The initial $\mathbf{x} = \mathbf{0}$ design is included with this sample as the first point. Given enough computational resources, this first phase is nicely parallelizable as each design point can be run independently. The number of samples to take should depend on the expected complexity of the design space. For the problems explored in this study, two samples per dimension has worked well.

Second, the surrogate model is refined using an Infill Sampling Criteria (ISC) based upon Expected Improvement (EI). This sampling criteria is well known for its ability to leverage both estimated function value and uncertainty to balance design space exploration (targeting global accuracy) and exploitation (targeting local optimality)[52, 119]. Expected improvement is commonly expressed as:

$$\begin{aligned} \mathrm{E}\left[I(\mathbf{x})\right] &= \mathrm{E}\left[\,\max(f_{\min} - F, 0)\,\right] \\ &= (f_{\min} - g(\mathbf{x})^*)\Phi\left(\frac{f_{\min} - g^*(\mathbf{x})}{s^*(\mathbf{x})}\right) + s^*(\mathbf{x})\phi\left(\frac{f_{\min} - g^*(\mathbf{x})}{s^*(\mathbf{x})}\right). \end{aligned} \tag{5.2}$$

In the above, $\Phi(\cdot)$ and $\phi(\cdot)$ are the standard normal distribution and density functions, s^* is the estimated variance of the response surface, and $f_{\min}$ is the minimum of the current training point sample.

If additional information in the form of constraints is available in the problem, it can be added to the infill sampling criterion by searching in locations of high probability of feasibility. Following the method described by Shimoyama[123], this is done by building a second surrogate model and evaluating the expected improvement, conditioned by the probability of feasibility,

$$\mathrm{P}[f_c(\mathbf{x}) < 0] = \phi\left(\frac{g_c^*(\mathbf{x})}{s_c^*(\mathbf{x})}\right), \tag{5.3}$$

where the constraint is formulated as $f_c(x) < 0$, and c^* is the estimated constraint value with estimated variance, s_c^*, each at location in the design space $\mathbf{x}$.

In working experience, the expected improvement infill sampling criteria often added training data near the box boundaries and corners of the design space. This is inefficient because at least half of a training point's region of influence is wasted modeling the response surface outside the boundaries. To encourage sampling inside the design space, a penalty function is applied,

$$\mathrm{B}(\mathbf{x}) = 1 - \exp\left(-\frac{1}{2}\min\left(\frac{x_i - lb_i}{a_i^2}, \frac{ub_i - x_i}{a_i^2}, i = 1, ..., d\right)\right), \tag{5.4}$$

which is a Gaussian function of the distance from the nearest hypercube boundary. In this study, the length-scale a_i is set to $0.1 \cdot (ub_i - lb_i)$, or 1% of hypercube dimension i's length. The above penalty function is greatly simplified in the non-dimensional space constructed in Section 3.2.

Combining the expected improvement, constraint penalty, and boundary penalty results in the first infill sampling criterion:

$$\begin{gathered} ISC_1(\mathbf{x}) = \mathrm{E}\left[I(\mathbf{x})\right] \cdot \mathrm{P}[f_c(\mathbf{x}) < 0] \cdot \mathrm{B}(\mathbf{x}) \\ \mathbf{x}_{new} = \mathrm{argmax}\left(ISC_1(\mathbf{x})\right), \end{gathered} \tag{5.5}$$

The response surface is refined by sampling $f(\mathbf{x}_{new})$ and $f_c(\mathbf{x}_{new})$ in this way until the criterion converges to a small value,

$$R_1(i) = ISC_1(\mathbf{x}_i) < T_1, \tag{5.6}$$

where T_1 is a specified tolerance.

While indeed efficient, working experience with GPR-based SBO has suggested that after expected improvement has converged to some preset value, it becomes difficult to discover the sharp and narrow wells that develop in its surface. For this reason, it can be more efficient to switch to an infill sampling criteria based on pure exploitation.

Thus the final phase of this method's design space sampling is based on refining around the estimated deterministic optimum of the surrogate model:

$$\begin{aligned} ISC_2(\mathbf{x}) &= g^*(\mathbf{x}), \\ \mathbf{x}_{new} &= \operatorname{argmin}\left(ISC_2(\mathbf{x})\right), \;\; s.t. \;\; g_c^*(x) < 0. \end{aligned} \tag{5.7}$$

This procedure continues until the measured improvement of the design sample converges to a small value

$$R_2(i) = \max\left(\, f_{best} - f_i \,,\, R_2(i-1) \,\right) < T_2, \tag{5.8}$$

or until the sampling stalls after not finding an improvement for a predetermined number of iterations. In the above equation, T_2 is another specified tolerance, and f_{best} is the minimum of the training data that came before f_i. The approach uses similar monitors on the change in location of the best point, and the norm of the gradient of the best point, but often the function value of best point converges the fastest.

Both the second and third sampling stages use a two-part optimization of the surrogate model. First a genetic algorithm is used to find a reasonable estimate of the global minimum. This estimate is fed as the initial guess to a gradient based optimizer, which returns a better estimate of the global minimum. This process requires several thousand function evaluations of the response surface, but is far less expensive than evaluating an additional high-fidelity design point.

5.2.2 Optimization in Active Subspaces

Optimizing in active subspaces requires special considerations for translating data between the full dimensional parameterization and the reduced active parameterization. As will be shown, this can be managed with a toolset of subspace regularization techniques.

The definition of $\mathbf{y} = \boldsymbol{U}^T\mathbf{x}$ is a surjective map $\mathbf{y}(\mathbf{x})$, since $\boldsymbol{U}$ is a tall rectangular matrix by construction. This map defines a set of full-space coordinates $\mathbf{x}$ which map to a smaller dimension of active-subspace coordinates $\mathbf{y}$. To make use of this for optimization, an inverse map is needed, and thus requires the definition of a process for regularization.

If coordinates in both the active an inactive space are known, then an inverse map can be used to find the corresponding point in the full space.

$$\mathbf{x} = \boldsymbol{U}\mathbf{y} + \boldsymbol{V}\mathbf{z}. \tag{5.9}$$

Which can be shown according to,

$$\begin{aligned} \mathbf{x} &= \boldsymbol{I}\mathbf{x} \\ &= \boldsymbol{W}\boldsymbol{W}^\top\mathbf{x} \\ &= \boldsymbol{U}\boldsymbol{U}^\top\mathbf{x} + \boldsymbol{V}\boldsymbol{V}^\top\mathbf{x} \\ &= \boldsymbol{U}\mathbf{y} + \boldsymbol{V}\mathbf{z}. \end{aligned} \tag{5.10}$$

However, without any considerations the inverse map actually defines an infinite set of points in the full space. Assumptions must be expressed about the full space in order to *regularize* on a single point, which will be sent back to the flow solver. The next section will describe several formulations for regularization that are appropriate for an optimization context.

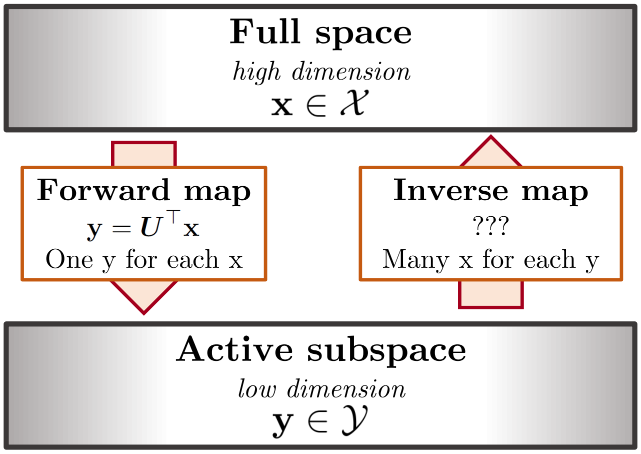


Figure 5.2: The inverse mapping problem.

A Simple Inverse Map

A simple approach to implement the inverse map is to take advantage of the orthogonal basis $\boldsymbol{U}$ and inverting the linear equation for the forward map:

$$\begin{aligned} \text{given} \quad & \mathbf{y} = \mathbf{y}_{select} \\ & \mathbf{y} = \boldsymbol{U}^T \mathbf{x} \\ \text{yield} \quad & \mathbf{x} = \boldsymbol{U}\mathbf{y} + \mathbf{x}_0 \end{aligned} \tag{5.11}$$

where an active subspace coordinate of interest $\mathbf{y}_{select}$ has been selected, and the goal is to find a full space coordinate $\mathbf{x}$ which maps to the subspace coordinate.

This can be made more robust by centering the sample data's features $\mathbf{x}$ on the origin. It should provide a nearly valid inverse map, if the active subspace describes nearly all functional behavior, since any other location in the projection space defined by $\mathbf{y} = \boldsymbol{U}^T\mathbf{x}$ should have approximately the same function value. However, this approach may yield full-space design vectors that are outside of the problem's bounding hypercube. It also does not allow enough flexibility when combining multiple objectives, for example in a constrained optimization problem, where it may be desired to allow movement in the null-space to meet a constraint.

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 $\mathbf{y}_{select}$.

$$\begin{aligned} \text{given} \quad & \mathbf{y} = \mathbf{y}_{select} \\ \text{find} \quad & \mathbf{x} \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{X} \\ & \mathbf{y} = \boldsymbol{U}^T \mathbf{x} \end{aligned} \tag{5.12}$$

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

$$\begin{aligned} \text{given} \quad & \mathbf{y} = \mathbf{y}_{select} \\ \underset{\mathbf{x}}{\text{minimize}} \quad & \mathbf{0}^\top \mathbf{x} \ \ \text{(a dummy function)} \\ \text{subject to} \quad & lb_i < \mathbf{x}_i < ub_i, \ i \in \{0, ..., m\} \\ & \mathbf{y} = \boldsymbol{U}^T \mathbf{x} \\ \text{yield} \quad & \mathbf{x} \end{aligned} \tag{5.13}$$

To use typical linear programming toolboxes, a linear objective must be defined. In the absence of this, a constant zero dummy objective is sufficient. 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 **x**. Depending on the initial conditions of the linear program and the trajectory the program takes, it 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 **x** outside of the parameterization of **y** have negligible effect on the objective function of interest.

Optimization with Non-Linear Constraints

This is needed for constrained optimization, where two or more functionals, and thus two or more active subspaces, are needed. It is advantageous to allow the multiple 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 two or more functionals through the full space $\mathcal{X}$ by constructing the following optimization:

$$\begin{array}{rl} \text{given} & f_a(\mathbf{x}) \approx g_a^*(\boldsymbol{U}_a^T\mathbf{x}),\ \boldsymbol{U}_a \in \mathbb{R}_{m\times n_a} \\ & f_b(\mathbf{x}) \approx g_b^*(\boldsymbol{U}_b^T\mathbf{x}),\ \boldsymbol{U}_b \in \mathbb{R}_{m\times n_b} \\ \underset{\mathbf{x}}{\text{minimize}} & g_a^*(\boldsymbol{U}_a^T\mathbf{x}) \\ \text{subject to} & lb_i < \mathbf{x}_i < ub_i,\ i \in \{0,...,m\} \\ & g_b^*(\boldsymbol{U}_b\mathbf{x}) \leq c \end{array} \tag{5.14}$$

In the above formulation, the bases $\boldsymbol{U}_a$ and $\boldsymbol{U}_b$ represent the subspaces for two separate functionals, $f_a(\mathbf{x})$, and $f_b(\mathbf{x})$, which are approximated with surrogate models in the active subspaces $g_a^*(\mathbf{y}_a)$ and $g^*(\mathbf{y}_b)$. There exists a common link between f_a and f_b via their embedding in the full space $\mathcal{X}$. However, the surrogate models are interrogated in their respective subspaces. The optimization links these two spaces while running a constrained optimization problem. In this case it is not possible to apply a linear program because of the potentially non-linear constraint g_b^*.

Here two quantities of interest were considered, but the formulation easily generalizes to more constraints.

Constraining the Inactive Space

When working with active subspaces in the way presented thus far, there remains a basis of undetermined parameters that the optimizer can exploit to move the objective while maintaining feasible constraints. As a heuristic, the dissertation contributes a constraint on this inactive space in order to minimize the chance that the problem will walk to the corners of the design space. In general, the optimizations constructed in this work have large enough design bounds where a constrained optimum is not expected in the bounds or corners of the design space. In fact it is likely that if a design is found in a corner, parameters have been skewed so much that physically impossible geometry results (such as inverted airfoils). It is desirable to avoid that.

One possibility here is to add a constraint on the z-space for each quantity of interest.

$$\begin{aligned} \text{given} \quad & f_a(\mathbf{x}) \approx g_a^*(\boldsymbol{U}_a^T\mathbf{x}),\ \boldsymbol{U}_a \in \mathbb{R}_{m\times n_a} \\ & f_b(\mathbf{x}) \approx g_b^*(\boldsymbol{U}_b^T\mathbf{x}),\ \boldsymbol{U}_b \in \mathbb{R}_{m\times n_b} \\ \underset{\mathbf{x}}{\text{minimize}} \quad & g_a^*(\boldsymbol{U}_a^T\mathbf{x}) \\ \text{subject to} \quad & lb_i < \mathbf{x}_i < ub_i,\ i \in \{0, ..., m\} \\ & g_b^*(\boldsymbol{U}_b\mathbf{x}) \leq c \\ & ||\boldsymbol{V}_a^T\mathbf{x}||_2 \leq z_a \\ & ||\boldsymbol{V}_b^T\mathbf{x}||_2 \leq z_b \end{aligned} \tag{5.15}$$

where V_a and V_b are the eigenvector bases for the inactive spaces of the quantities of interest f_a and f_b respectively. The magnitude of the displacement of the design parameterization into the inactive space is constrained for each quantity of interest with the values of z_a and z_b. These parameters must be explored for each problem to find a useful configuration that results in optimal results. The formulation is also easily extended to account for additional constraints, $(f_c, f_d, ...)$ for example.

Optimization with High Dimensions

When the full space is of large dimension, it is advantageous to construct the optimization in the active subspace to reduce dimensionality of the problem. In this case the variables of the optimizer become the active subspace vectors. This optimization is shown in Equation 5.16 below.

$$\begin{aligned} \text{given} \quad & f_a(\mathbf{x}) \approx g_a^*(\mathbf{y}_a),\ \mathbf{y}_a \in \mathbb{R}_{n_a} \\ & f_b(\mathbf{x}) \approx g_b^*(\mathbf{y}_b),\ \mathbf{y}_b \in \mathbb{R}_{n_b} \\ \underset{\mathbf{y}_a,\mathbf{y}_b}{\text{minimize}} \quad & g_a^*(\mathbf{y}_a) \\ \text{subject to} \quad & g_b^*(\mathbf{y}_b) \leq c_b \\ & c(\mathbf{y}_a, \mathbf{y}_b) \leq 0 \end{aligned} \tag{5.16}$$

In the case of an objective and constraints, with separate active subspaces, a consistency constraint $c(\mathbf{y}_a, \mathbf{y}_b)$ is needed to ensure that a point can be found in the full space that projects into both active subspaces in their respective locations $\mathbf{y}_a$ and $\mathbf{y}_b$, as well as respect the global design bounds. To satisfy these conditions an inner optimization is required to perform a regularization, shown in Equation 5.17.

$$\begin{array}{rl} \text{given} & \mathbf{y}_a \in \mathbb{R}_{n_a} \\ & \mathbf{y}_b \in \mathbb{R}_{n_b} \\ \underset{\mathbf{x}}{\text{minimize}} & \mathbf{x}^\top \mathbf{x} \\ \text{subject to} & lb_i < x_i < ub_i,\ i \in \{0, ..., m\} \\ & \boldsymbol{U}_a^\top \mathbf{x} = \mathbf{y}_a \\ & \boldsymbol{U}_b^\top \mathbf{x} = \mathbf{y}_b \\ \text{yield} & \mathbf{x}^* \end{array} \tag{5.17}$$

In this case the point in the fullspace is further regularized by trying to minimize the norm of the full space vector $\mathbf{x}$. Note that given the active space equality constraints, this has the effect of reducing the distance that the selected point $\mathbf{x}$ walks out in to the inactive spaces. The above optimization problem can be easily solved using a quadratic program.

To use this algorithm for a constrained optimization in the active subspaces as in Equation 5.16. The constraint violations are packed up into a norm in order to provide a single scalar value to constrain. The alternative is to include each constraint of Equation 5.17 in the optimization of Equation 5.16, based on the regularized result $\mathbf{x}^*$. However this would penalize an otherwise low-dimensional problem with a larger number of constraints. Thus consistency constraint $c(\mathbf{y}_a, \mathbf{y}_b)$ is presented as such.

$$c(\mathbf{y}_a, \mathbf{y}_b) = \left\| \begin{array}{c} \min\{x_i^* - ub_i, 0_i\} \\ \min\{lb_i - x_i^*, 0_i\} \\ \boldsymbol{W}_{a,1}^\top \mathbf{x}^* - \mathbf{y}_a \\ \boldsymbol{W}_{b,1}^\top \mathbf{x}^* - \mathbf{y}_b \end{array} \right\|_2, \tag{5.18}$$

where the L_2 norm is used. There are four sub-blocks. The first and second sub-blocks are composed of column vectors of the global box bound constraint violations, or zero for each axis that is inside its bounds. The second two sub-blocks are the errors of the selected point $\mathbf{x}^*$ projected and compared to the given active spaces variables.

5.3 Numerical Experiments

The methods presented through this thesis are now exercised together in numerical optimization experiments. The first set of experiments will provide a baseline approach to global optimization using gradient based optimizations with random start locations. Then attempts will be made to apply gradient-enhanced Gaussian process regression with adaptive refinement as an alternative for global optimization. As this approach will struggle with high-dimensional problems, the last examples pivot to use active subspaces with surrogate based optimization in order to discover regions of the design space which may be appropriate for local refinement.

5.3.1 Random Start Gradient Based Optimization

As described in the introduction, gradient based optimization is a standard approach for CFD based design problems. It is efficient in high dimension with gradients. As a local optimizer though it can miss global optima. Furthermore, when gradient information is inaccurate, this can introduce noise that can create ghost local minima.

Gradient based optimization can be extended into global optimization by randomly starting from samples in the design space. This gives the optimizer an opportunity to explore different minima. This is what has been done for the present experiment.

Eleven start positions were drawn using Latin hypercube sampling. The first sample was the baseline design. A sequential least squares quadratic programming (scipy's SLSQP) algorithm was used to drive the gradient based optimization. No more than 150 evaluations were permitted. The convergence tolerance was set to 1×10^{-6} on the objective and constraints as scaled for the optimizer.

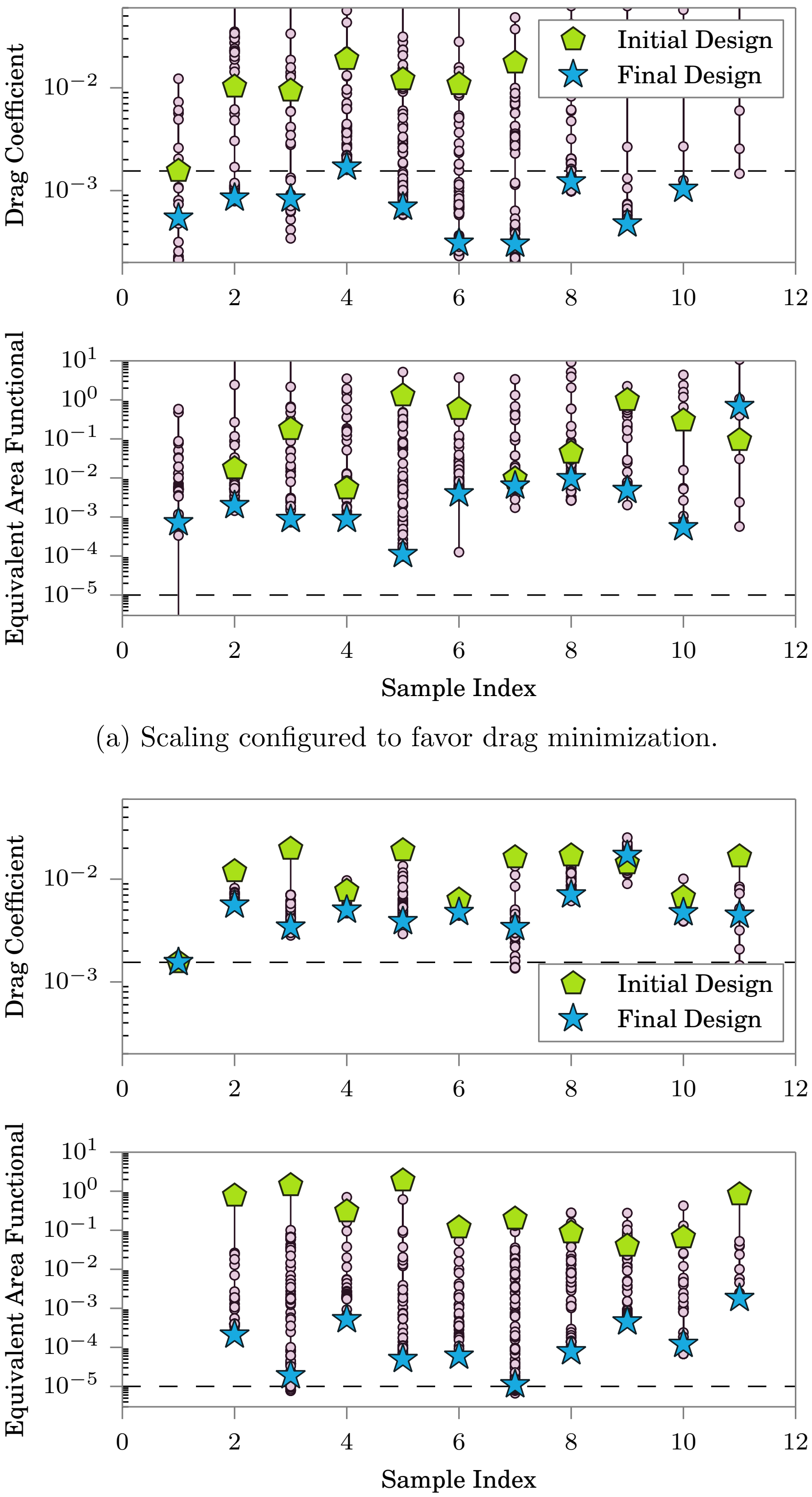
This optimizer needs the inputs and objectives to be scaled to condition the problem to balance the requirements of the constraints with the goal of minimizing an objective. Two scaling configurations were used, resulting in two different sets of results.

The results are plotted in Figure 5.3. In the first set shown in Figure 5.3a, the problem was configured to emphasize minimizing the objective. This allowed significant reductions in drag to be discovered, but at the cost of allowing changes to the equivalent are on the order of 1×10^{-4}. An example of the final geometry based on sample 5 is provided in Figure 5.4.

In the second set of random GBO samples shown in Figure 5.3b, the scaling was configured to emphasize the constraint. In this case many samples were able to meet a target equivalent area violation of 1×10^{-5}, but were unable to reduce the drag past the baseline. An example of the final geometry based on sample 7 is provided in Figure 5.4.

This is an issue, because the problem as setup, with physics in the loop and with adjoint gradients, is not immediately movable by a standard optimization package.

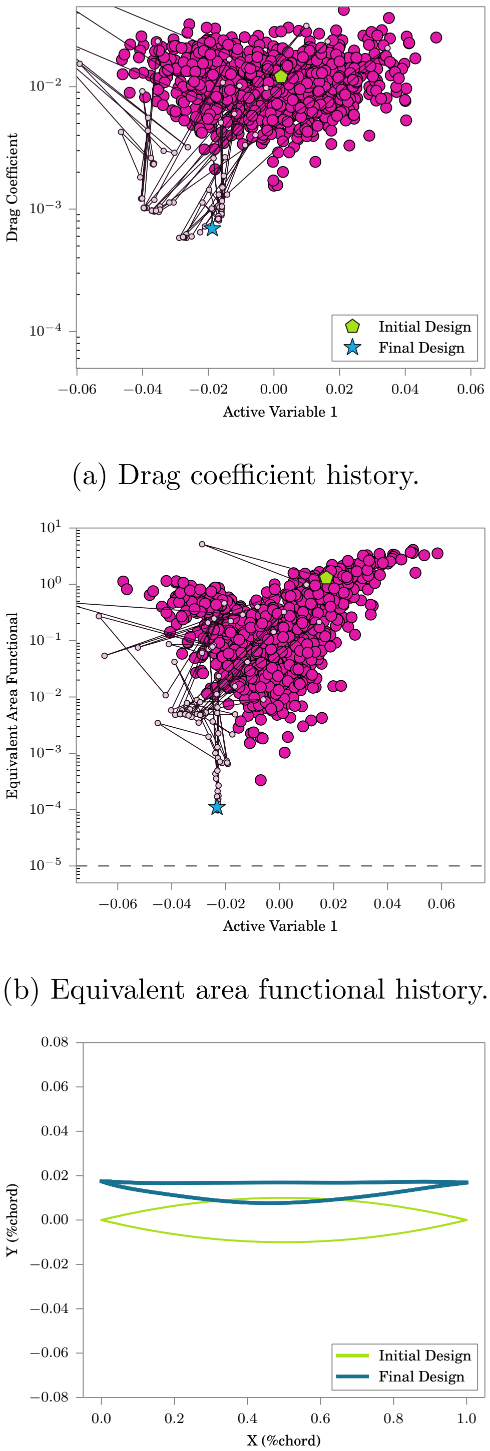
Another surprising result is that all the different samples of GBO terminated in different locations in the design space, with different realizations of an "optimized" geometry. This suggests that multiple local minima are present, at least as the result of numerical noise in the problem. Such is an issue that will be addressed with surrogate based optimization with active subspaces.



(a) Scaling configured to favor drag minimization.

(b) Scaling configured to favor enforcing the equivalent area constraint.

Figure 5.3: Summary of random-start gradient based optimizations.

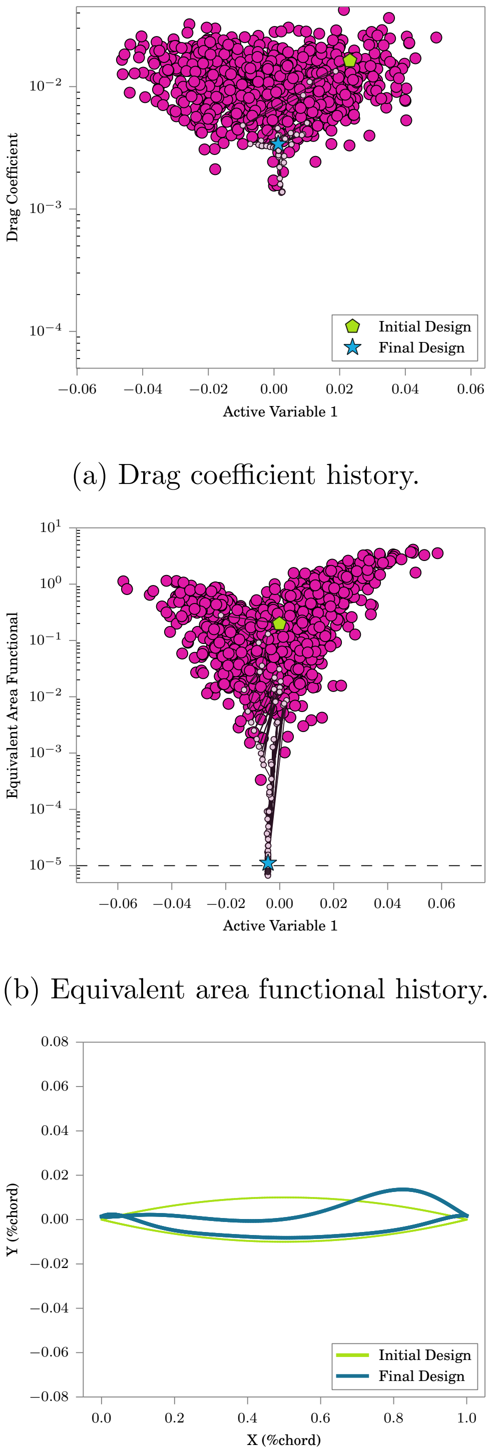


(a) Drag coefficient history.

(b) Equivalent area functional history.

(c) Converged airfoil design.

Figure 5.4: Gradient based optimization corresponding to sample 5 in Figure 5.3a.



(a) Drag coefficient history.

(b) Equivalent area functional history.

(c) Converged airfoil design.

Figure 5.5: Gradient based optimization corresponding to sample 7 in Figure 5.3b.

5.3.2 Surrogate Based Optimization

The results presented in this section will use the SBO method described in Section 5.2.1.

5.3.2.1 Rosenbrock Function - Minimization

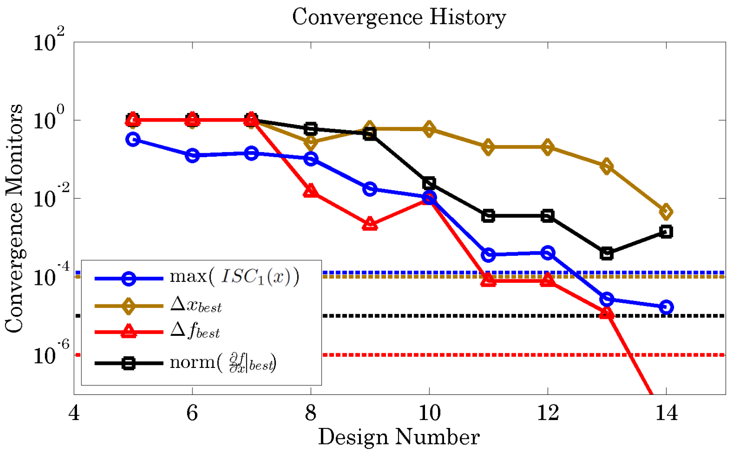
This example will minimize the Rosenbrock function without a constraint constraint. The general problem is summarized here:

$$\begin{array}{ll} \underset{\mathbf{x}}{\text{minimize}} & f_o(\mathbf{x}) \\ \text{subject to} & -2.0 < \mathbf{x}_i < 2.0,\ i \in \{1, 2\}. \end{array} \tag{5.19}$$

In the above, the objective f_o is the Rosenbrock function in two dimensions introduced in Chapter 2.

Using the adaptive refinement approach to surrogate based optimization, using two infill sampling criterion, the first example is optimized without a constraint. A plot of the convergence monitors is shown in Figure 5.6. This figure shows the traces for four convergence parameters. The first in blue circles is the maximum modified expected improvement (R_1 as described earlier). The second in green diamonds is the norm of the change in location of the best training point. The third in red triangles is the magnitude of change in value of the best training point. The fourth in black squares is the norm of the gradient of the best training point. The tolerances for convergence are plotted as the horizontal dashed line with corresponding color.

The convergence of these sensors explains the strategy of the SBO method proposed in this dissertation. The Δx_{best} and Δf_{best} values track how much the best sampled point is changing. The expected improvement infill sampling criterion is active for the first twelve samples, until it is predicting a relative improvement of 1×10^{-4}. In the iteration in which the SBO method predicts a low expected improvement, the infill sampling criterion changes to choose the best predicted point. Two additional iterations converge the surrogate model's estimated value of the best sampled objective. When this value becomes small it means that the surrogate model is not changing with additional samples around its predicted minimum. A stronger



$$x_{min} = [1.001556, 1.002614]\ ,\ f_{min} = 0.000027$$

Figure 5.6: Convergence of Unconstrained 2D Rosenbrock Problem

measure of convergence would be to wait until the change in sampled location Δx_{best} becomes nearly zero, and a stronger measure still in this unconstrained case would be to wait until the gradient of the best sample converges to zero. However, because of the noise model needed in this example to maintain numerical stability of the GPR problem, once the objective value is converged, this optimization method would run more samples by choosing nearly random perturbations around the optimum. A more efficient use of samples from this point would be to apply a gradient based algorithm, or to restart the surrogate based algorithm on a smaller design bound.

The surrogate-based optimization was compared to a gradient based optimization with the traditional starting point at $\mathbf{x} = [-1\,,\,1]$. A modified newton method (Matlab's fmincon, trust-region-reflective method) was used and a comparison of the solution history is shown in Figure 5.7. Three important comparisons can be drawn from this study. First, for this problem it took 68 fewer function evaluations of the test function to converge. The gradient based solution took so long because the initial point was placed such that the optimizer had to travel through the Rosenbrock function's long and flat banana-shaped valley, admittedly a contrived but typical case. However, this highlights the reduction in expensive function evaluations that can come by performing a thorough optimization on an accurate surrogate model.

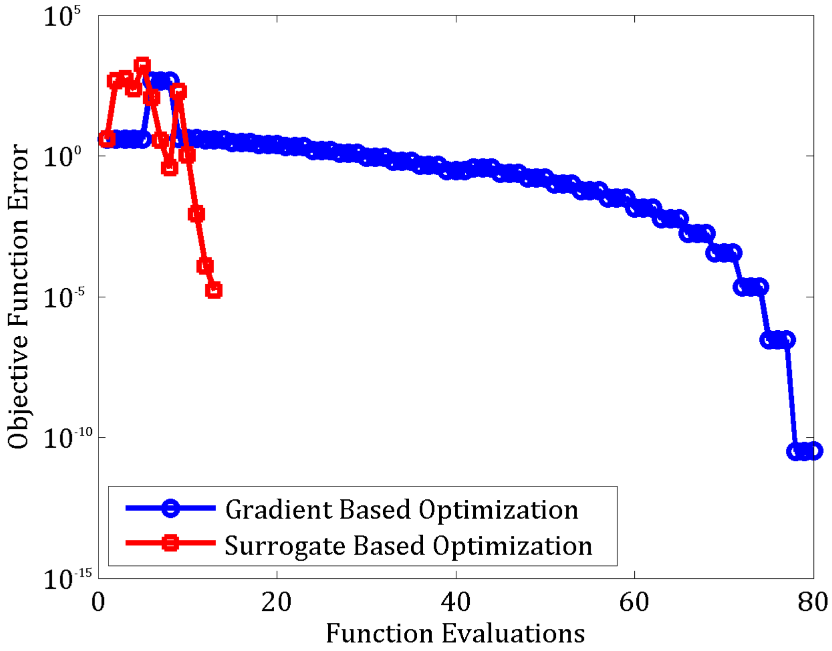


Figure 5.7: Comparison of Unconstrained GBO and SBO

Second, the SBO was only able converge the error of the function minimum to 1×10^{-5}, while the GBO was able to converge to 1×10^{-10}. This is because the surrogate model's accuracy is approximately limited by the value of the noise hyperparameters, which are still needed for this problem to maintain numerical stability. Finally, it is important to note that the GBO is fast in physical time, taking a fraction of a second to complete. Because the SBO is performing two secondary optimization problems (hyperparameter tuning and surrogate minimization) at each external iteration, it took approximately 400 seconds to converge this problem on a Intel quad-core PC with 8GB of ram. This is the trade-off made to build and scour the response surface model.

5.3.2.2 Rosenbrock Function - Minimization with Constraint on the Rastrigin Function

The combination of the Rosenbrock and Rastrigin functions (and their analytical gradients) exercise SBO's ability to handle complex and multimodal functions with constraints.

$$\begin{array}{ll} \underset{\mathbf{x}}{\text{minimize}} & f_o(\mathbf{x}) \\ \text{subject to} & -2.0 < \mathbf{x}_i < 2.0, \; i \in \{1,2\} \\ & f_c(\mathbf{x}) \leq 0. \end{array} \tag{5.20}$$

In the above, the objective f_o is the Rosenbrock function in two dimensions, and f_c is the modified Rastrigin function in two dimensions, both of which were introduced in Chapter 2.

The surfaces plotted in Figure 5.9 show the information used by the surrogate based optimizer midway through the solution of the constrained problem. Despite being only partially constructed, the constraint surface in Figure 5.9b is able to provide enough information to identify probable regions of feasibility in Figure 5.9d. Figure 5.9c shows the expected improvement surface. Notice the that the boundary penalty of Equation 5.4 conditions this surface to avoid sampling data in corners of the design space. Solving for the maximum of the conditioned expected improvement identifies a suitable location to run an additional sample.

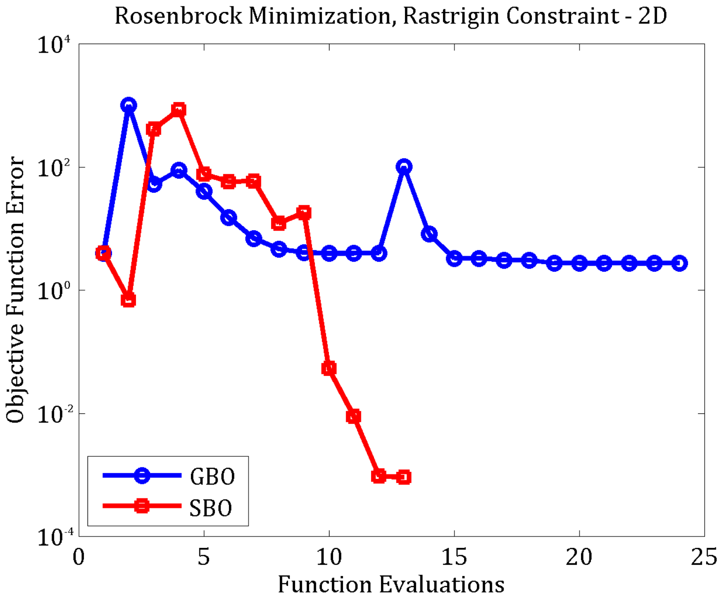
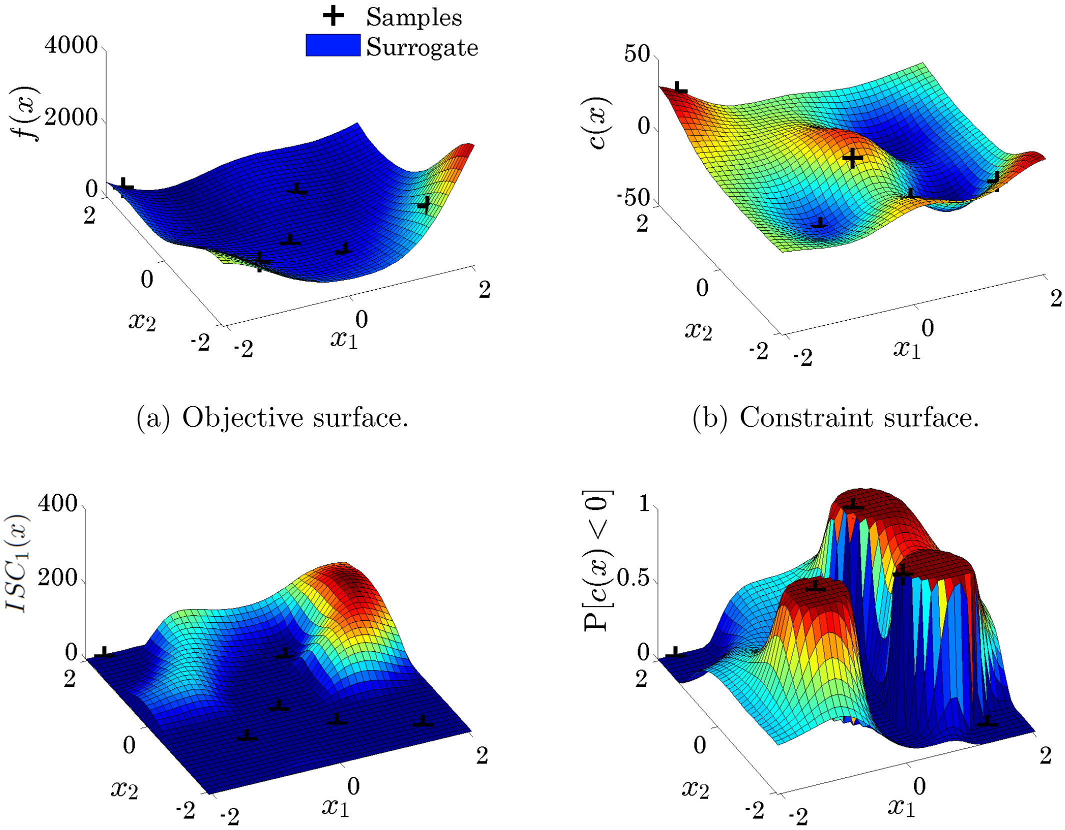


Figure 5.8: Comparison of Constrained GBO and SBO

This solution is again compared to the gradient-based solution in Figure 5.8, which this time starts in a feasible region that does not contain the global optimum. Because the SBO is exploring the entire design space, it is able to find the global optimum.



(a) Objective surface.

(b) Constraint surface.

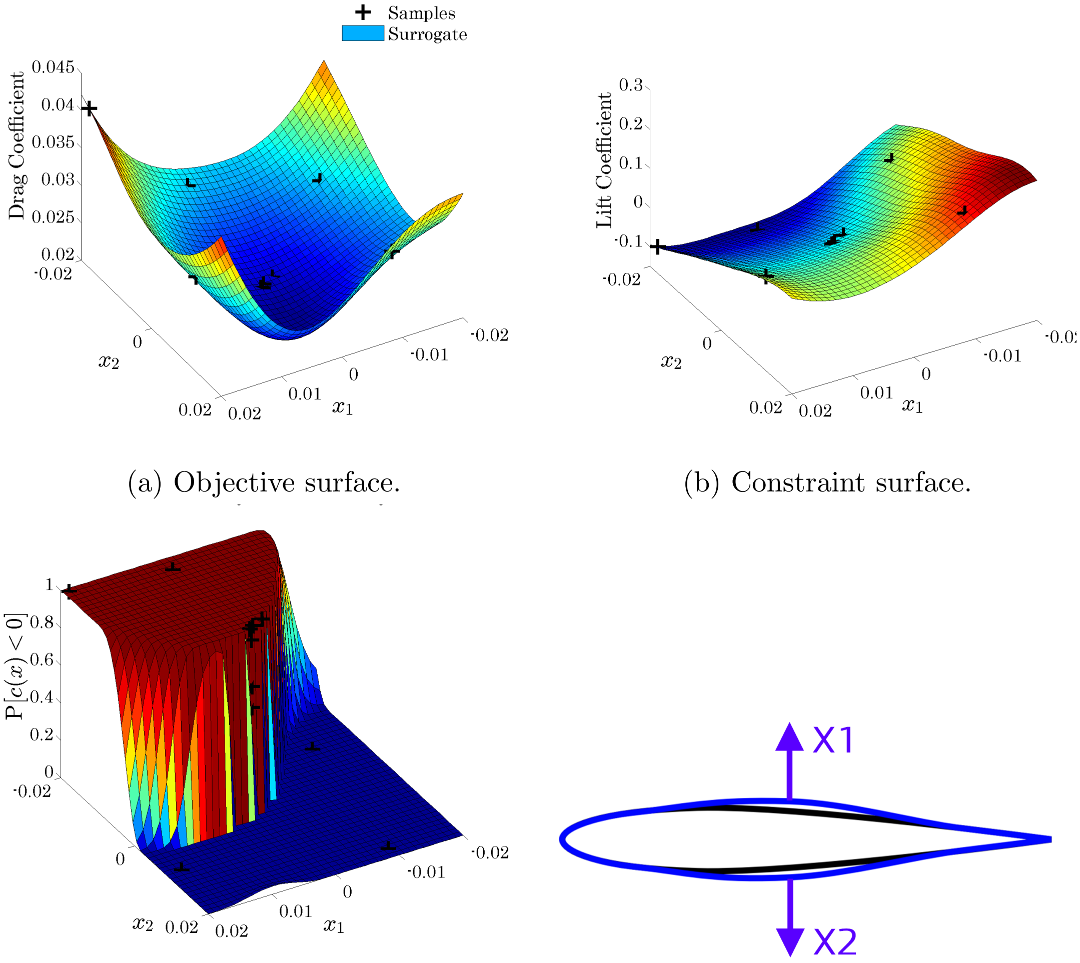
(c) Expected improvement.

(d) Probability of feasibility.

Figure 5.9: Example learning surfaces made with 6 samples for the optimization problem using Rosenbrock and Rastrigin test functions.

5.3.2.3 NACA 0012 Airfoil - Drag Minimization with Lift Constraint

In this test case, the drag, lift and their gradients were estimated on a two-dimensional NACA0012 airfoil, and an optimization problem was constructed using SBO according to the setup described in Chapter 2.



(a) Objective surface.

(b) Constraint surface.

(c) Probablility of feasibility.

(d) Problem parameterization.

Figure 5.10: Example learning surfaces for the NACA0012 constrained problem at the conclusion of the optimization.

Two Hicks-Henne bump functions were used for design variables, one each on the upper and lower surfaces. In this design problem, the optimum is located on the constraint boundary, so the surrogate model for the constraint will be very important to refine with reasonable accuracy near the constrained minimum.

The learning surfaces at the after converging the design are presented in Figure 5.10. The expected improvement surface is not shown as it has converged to nearly zero. The probability of constraint feasibility is shown in Figure 5.10c. In the process of optimizing under constraints, the SBO method here has been able to explore several designs on the constraint boundary. The drag coefficient surface in Figure 5.10a shows that these designs are indeed minimizing drag according to the surrogate model. Because this was a two dimensional design problem however, the optimization was only able to find two counts of drag reduction.

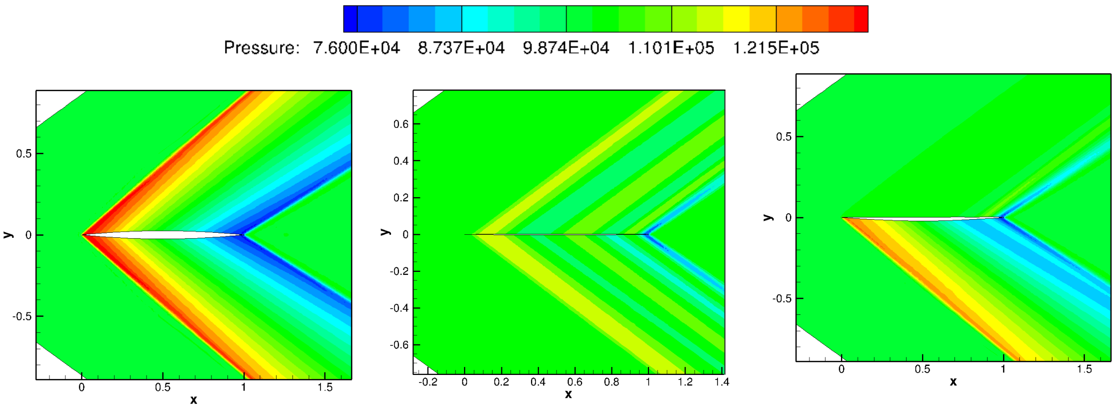
The collection of these surfaces shows a few important accomplishments. First that the SBO procedure suggested in this dissertation can interface with CFD in the loop with gradients to optimize a constrained design problem using GPR surrogate models. Second, it shows that the probability of constraint feasibility was useful for this design problem when applying SBO. And finally, the optimization was able to identify an improvement, as meager as it was, with thirteen samples of the design problem.

5.3.2.4 Supersonic Biparabolic Airfoil - Drag Minimization with Equivalent Area Constraint

This is a test case that demonstrates the need for active subspaces for surrogate based optimization. As will be shown, a subspace of the design variables has minimal effect on the equivalent area functional, which will be difficult for the surrogate modeling approach of this dissertation to manage.

In this test case, drag was reduced on the biparabolic airfoil with and without a constraint on the equivalent area, using the design setup shown in Chapter 2. In this particular experiment, 20 Hicks-Henne bump functions were used across the upper an lower surface of the airfoil. In this parameterization, there is no freedom to change the angle of attack of the airfoil.

The expected result for the unconstrained problem is to reduce the airfoil to a thin plate, which removes the thickness and lift contribution to wave drag, but changes the equivalent area significantly. In the constrained problem, the equivalent area is sensitive only to the lower surface, which must remain unchanged. However drag reduction is available by flattening the upper surface. A comparison of the resulting flow solutions obtained in this problem are presented in Figure 5.11.



(a) Baseline design.

(b) Unconstrained SBO.

(c) Equivalent area constrained SBO.

Figure 5.11: Biparabolic airfoil pressure contours.

Figure 5.12 compares the equivalent area signatures at the nearfield for the three cases. The constrained problem was able to partially enforce the equivalent area constraint to maintain a similar pressure distribution underneath the the airfoil, while flattening the upper airfoil surface, as shown in Figure 5.11c. The unconstrained optimization successfully identified that a minimum is available by collapsing the thickness of the airfoil.

For the unconstrained case, a total of 25 design iterations were performed. For the constrained case, a total of 34 design iterations were performed. As Figure 5.14 shows for the unconstrained case, a majority of these samples can be evaluated in parallel. In this case there are 20 initial design samples.

The convergence plot for this problem in Figure 5.13 shows that the two infill sampling criteria are performing as designed. First, the plot of the convergence monitors below show that the value of the expected improvement converged quickly by iteration 24. If this criterion alone was used to refine the response surface, the optimization would have stalled very early with a large constraint violation. However, the second sampling criterion was able to start refining the estimate of the constrained

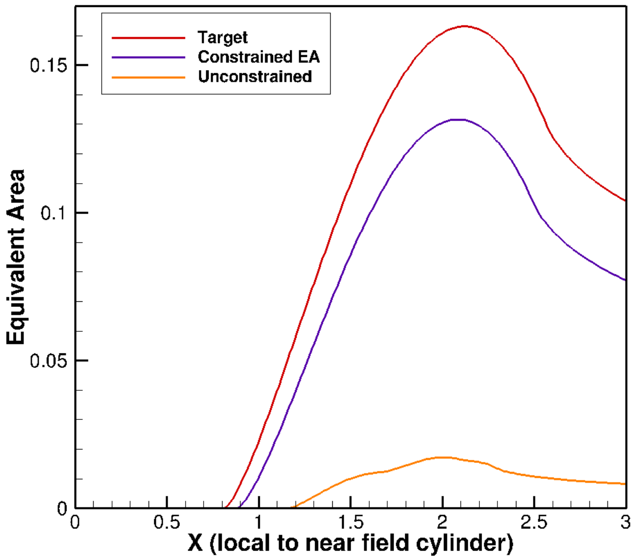


Figure 5.12: Equivalent area distributions for constrained and unconstrained drag SBO.

optimum, and was able to reduce the constraint violation to a pre-set threshold of five percent.

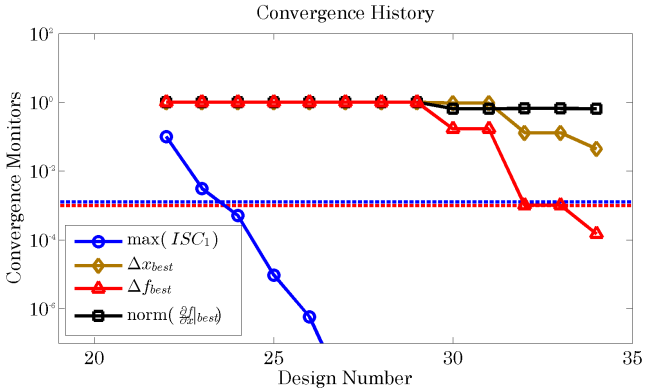


Figure 5.13: Convergence history: biparabolic airfoil.

The surrogate-based optimization approach struggles with this case because it violates the assumption of isotropic variation in the scaled design space. The equivalent area is not sensitive to the design variables above the airfoil, and as a result

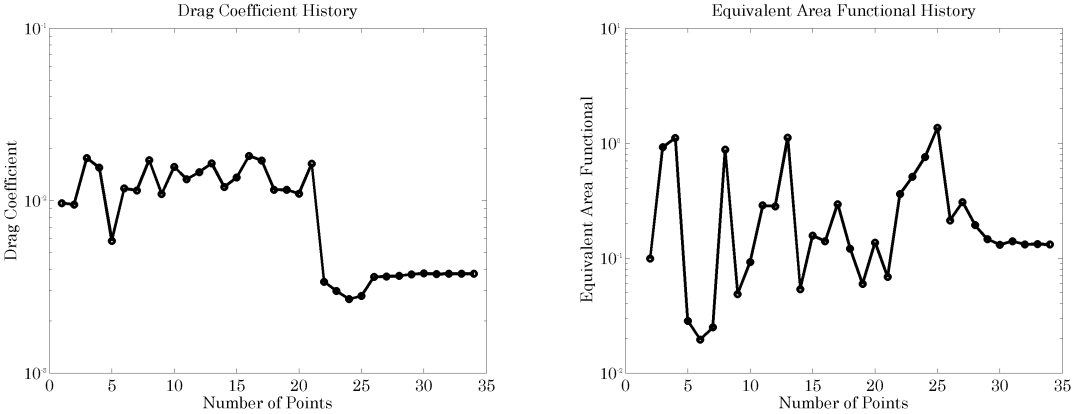


Figure 5.14: Objective and constraint history: biparabolic airfoil.

the true constraint surface will be exactly flat in the corresponding dimensions. The surrogate model, however, produces small variations in the constraint along these dimensions which requires the optimizer to explore many more designs to identify the true constraint boundary. This could of course be accounted for by tuning one length scale hyperparameter per dimension, at increased computational cost. A more clever approach might be to acknowledge *a-priori* the lack of sensitivity in these design variables and pre-scale their length scales by a large number. This would encourage the surrogate to more accurately model the low variation in these dimensions.

Such is the purpose of active subspaces.

5.3.2.5 N+2 Supersonic Passenger Jet - Unconstrained Drag Minimization

Surrogate based optimization was performed on the N+2 supersonic passenger jet using the nine free-form deformation control points placed on upper wing surface, as described in Chapter 2. The drag-only configuration of the N+2 is used here. This problem introduces the ability of the SBO approach to address complicated design problems.

A comparison with a gradient-based optimization of the problem is shown in Figure 5.15. The surrogate based optimizer was able to discover a 4.6% reduction in drag in 20 fewer iterations than a gradient based optimizer. Again, because the first 21 design points are independent of each other, they can be simulated simultaneously given enough computing resources, which further reduces the wall-clock time.

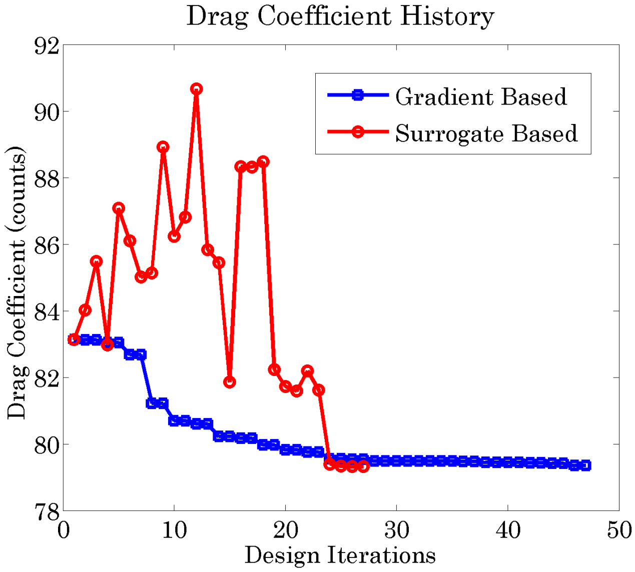


Figure 5.15: Comparison of unconstrained drag GBO and SBO

A plot of the pressure coefficient on the upper part of the vehicle is presented in Figure 5.17. A comparison of the geometry change between the baseline and final design in Figure 5.16 shows that the drag reduction was accomplished by reducing the thickness of the wing. This is an intuitively expected result, especially in the context of the results from the biparabolic test problem in the previous experiment.

It is enabled by the omission of the lift constraint.

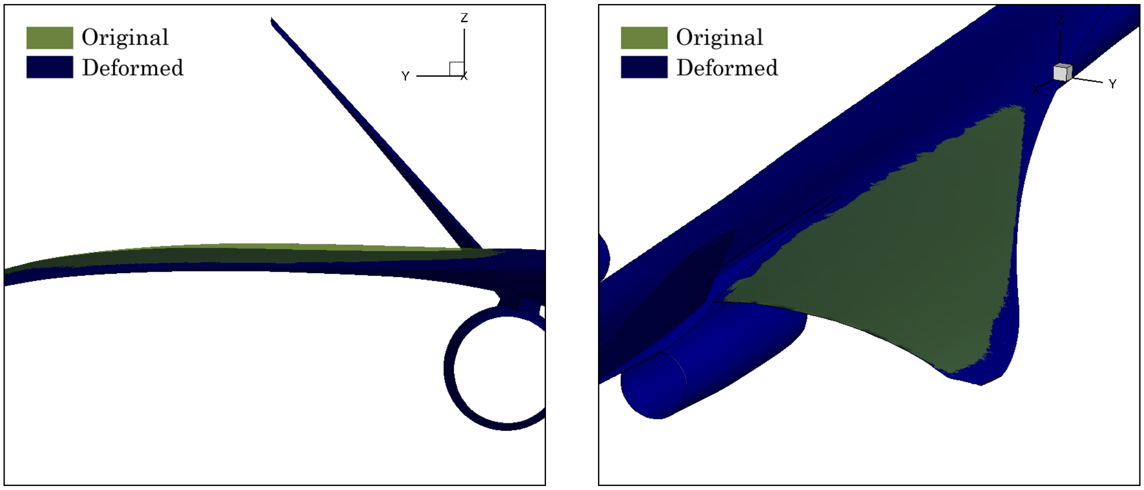


Figure 5.16: Original and deformed N+2 surfaces, unconstrained drag minimization.

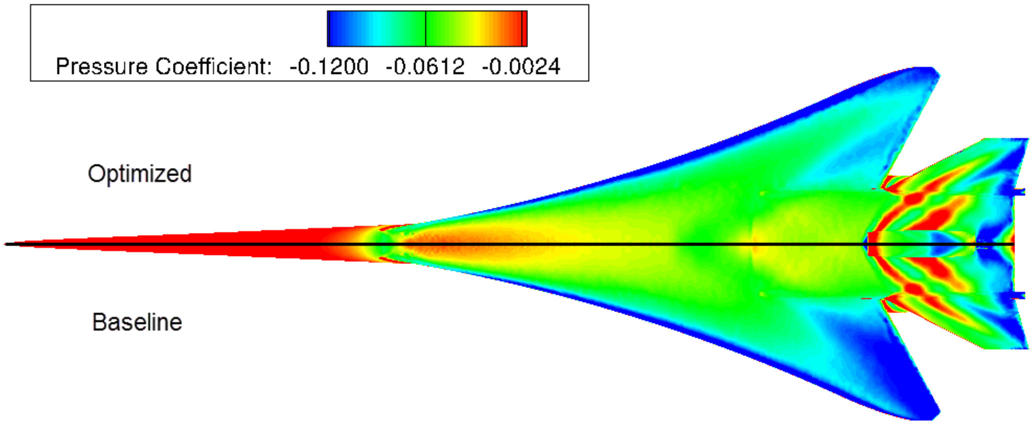


Figure 5.17: Contours of pressure coefficient for the baseline and optimized design after surrogate based optimization with GPR.

5.3.3 Active Subspaces for Surrogate Based Optimization

Here surrogate models are built in an active subspace to estimate optimal aerodynamic shapes. They will first be used to demonstrate an unconstrained drag minimization problem. Then they will be used on a lift-constrained drag minimization problem, which will be compared with the random-start gradient based optimization results of Section 5.3.1.

5.3.3.1 Analytical Test Case

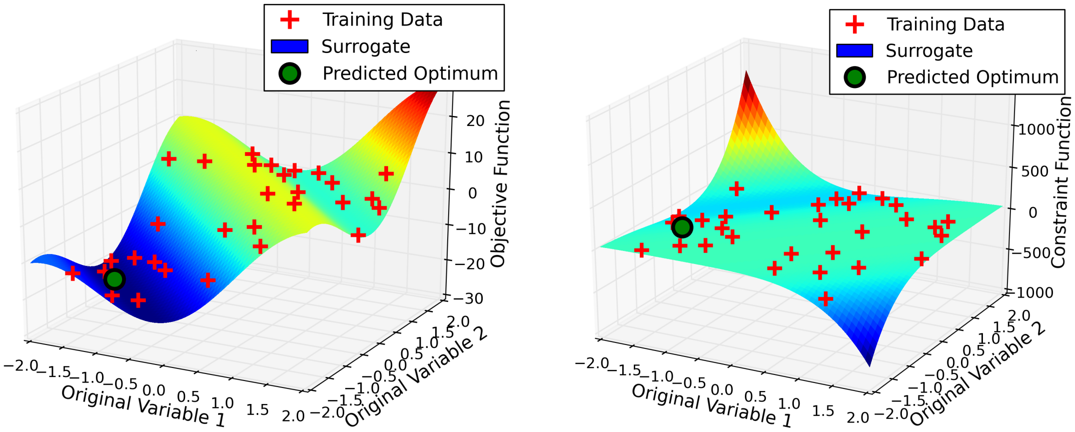
This section uses the rotated polynomial test functions described in Section 2.2.

Thirty training sample locations were chosen in $\mathcal{X}$. The objective and constraint functions and gradients were evaluated at these locations. These data were evaluated for their active subspace bases. One active domain was clearly identified by the eigenvalue decay plot - in both cases the second dimension's power was machine zero. The data were mapped forward into their respective domains, and two separate Gaussian Process regression surrogate models were trained on these mapped data.

The resulting functions are an objective surrogate model in the objective active subspace $g_o^*(y_o)$, and a constraint surrogate model in the constraint active subspace $g_c^*(y_c)$. These active subspace surrogate models can be used to solve the original domain's optimization problem with this surrogate optimization problem presented in Equation 5.14.

The toy problem is multimodal, but because surrogate problem evaluations are fast, it is possible to use a sample-based optimization approach. In this case, Covariance Matrix Adaptation [39] is applied. Had the original objective and constraint functions been expensive to evaluate, as in CFD simulations for example, this optimization strategy would be intractable.

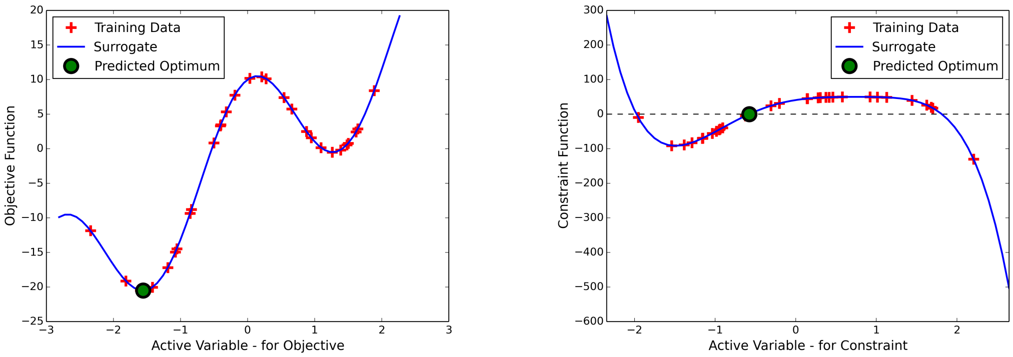
Plots of the surrogate models and the result of the optimization are presented in Figures 5.18 and 5.19. It can be seen that the optimizer found the region of the constrained global minimum for this problem, which is defined by the intersection of the two lines of the objective's minimum and constraint's second root. This is not the exact optimum, as the surrogate models are not refined around this location, which can be addressed with adaptive refinement, or by using the predicted optimum as a starting point for a local optimizer like gradient based methods.



(a) Objective surrogate model in the original domain.

(b) Constraint surrogate model in the original domain.

Figure 5.18: Surrogate models in the original domain.



(a) Objective surrogate model in the active domain.

(b) Constraint surrogate model in the active domain.

Figure 5.19: Surrogate models in the active domain.

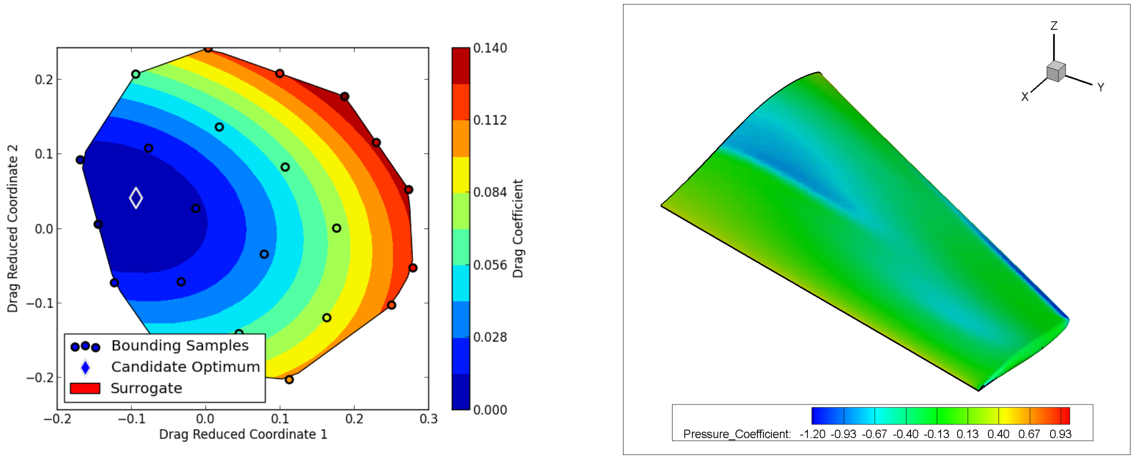
5.3.3.2 ONERA M6 - Unconstrained Drag Minimization

In this example, the optimization problem of the ONERA M6 parameterized with 50 FFD control points is explored with active subspaces and surrogate based optimization to minimize drag without a lift constraint. Because there is no need to communicate with a separate subspace for lift, it is possible to directly optimize on the surrogate model in the drag subspace. In the process of applying the active subspace analysis, five active subspace dimensions were identified as needed by inspecting the eigenvalues to properly capture the drag behavior. However, in order to visualize the problem the surrogate and optimization will be built in only two dimensions. In this case, the inverse maps of Section 5.2.2 is modified to solve the following optimization problem:

$$\begin{aligned} \underset{\mathbf{y}}{\text{minimize}} \quad & g^*(\mathbf{y}), \quad \mathbf{y} \in \mathbb{R}^2 \\ \text{subject to} \quad & \mathbf{y} \in \mathcal{Y} \textit{ (recall Equation 5.13).} \end{aligned} \tag{5.21}$$

Here, $g^*(\mathbf{y})$ is a surrogate model constructed from the initial 300 training samples projected into the active subspace. 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 $\mathcal{X}$, as illustrated in Equation 4.9. This requires a secondary optimization problem at each evaluation of $\mathbf{y}$. This problem solves the linear program described in Equation 5.13.

A plot of the surrogate model visualized in two dimensions using the first two active subspace eigenvectors is shown in Figure 5.20a. The bounding hull of this plot is constrained by the full-space bounds of $\mathcal{X}$. The shape approximates what is essentially the silhouette of a 50-dimensional hypercube projected into two dimensions. The verticies of the hull are found by coarsening the convex hull of the points represented by all the hypercube corners projected into this space. The surrogate indicated by the color contours is generated by fitting the data in two dimensions. The candidate optimum indicated by the diamond marker is found by performing an optimization on the two-dimensional surrogate model. Then the point is mapped back to the full space using the regularization strategy, in order to identify a full-space design to evaluate.



(a) Surrogate model in the 2-D active subspace for drag without lift constraint.

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

Figure 5.20: Results for an unconstrained drag minimization in with active subspaces.

Because of the minimal computational expense, it is possible use a genetic algorithm on the surrogate model. Applied here again is a python implementation of Covariance Matrix Adaptation (CMA)[39]. This demonstrates the ability to conduct *high-dimensional global optimization* using surrogate models with active subspaces. The full-space problem has a total of 50 dimensions, which would have been intractable to regress even with gradient information.

The optimization problem identified a candidate design, which is evaluated with CFD to verify the design's drag. The resulting plot of pressure coefficient in Figure 5.20b show that the shock line on the upper surface of the wing was reduced near the leading edge. This can be compared with the baseline contour plot provided in Figure 2.10a. 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.

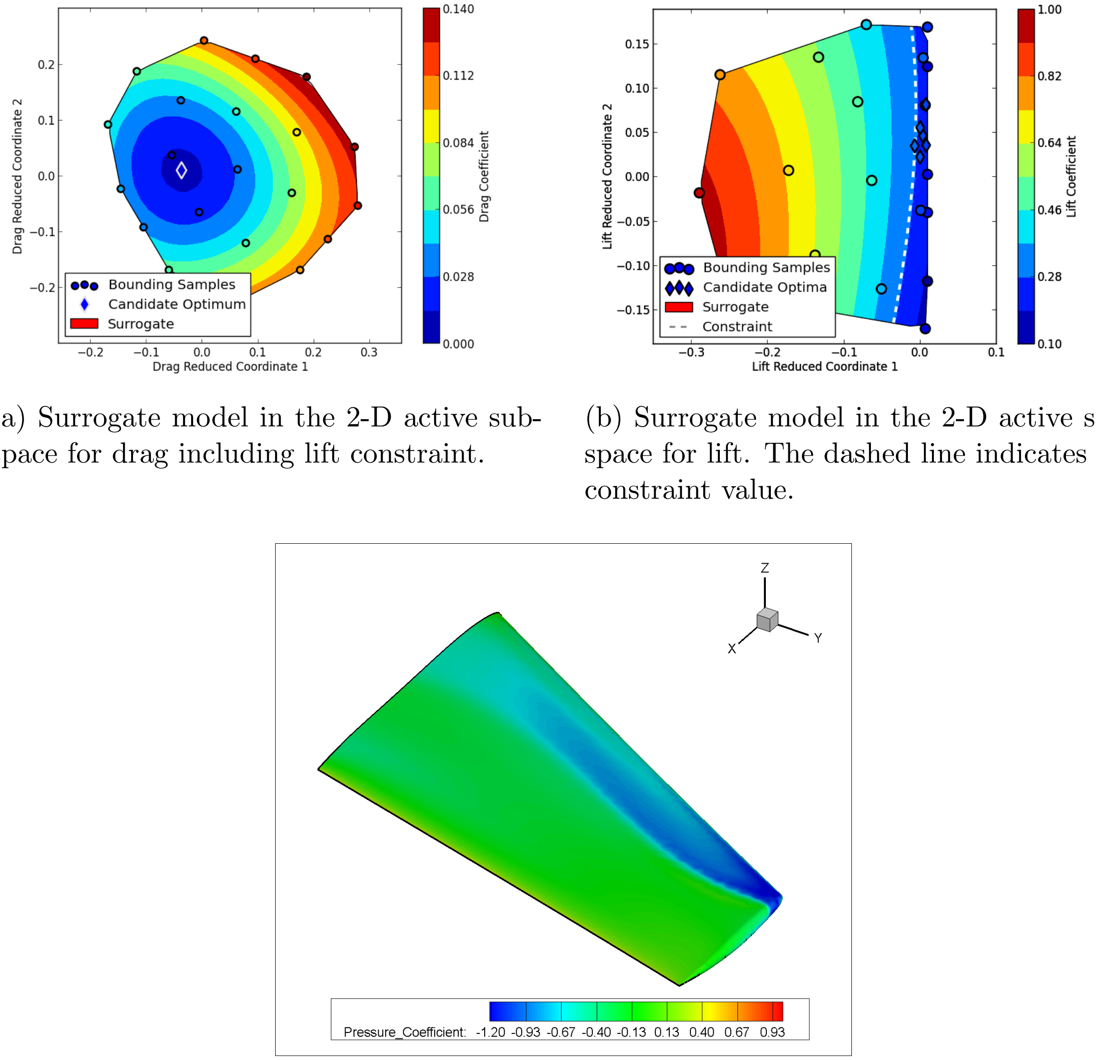
5.3.3.3 ONERA M6 - Drag Minimization with Lift Constraint

In this example, the full optimization problem is evaluated, minimizing drag while enforcing a lift constraint. There is now a requirement to communicate between subspaces for drag and lift. As identified during the eigenvalue analysis, five active subspace dimensions are sufficient to properly capture the drag behavior. However, in this case the predictive model is constructed in two dimensions. As a result, an inverse map is needed that connects three different active subspaces.

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

$$\begin{array}{cl} \text{given} & \mathbf{y}_{a1} \in \mathbb{R}^2 \\ \underset{\mathbf{y}_{a1}}{\text{minimize}} & C_D \sim g_{a1}(\mathbf{y}_{a1}) \\ \text{subject to} & \mathbf{y}_{a1} \in \mathcal{Y}_{a1} \\ \text{yield} & \mathbf{y}_{select}, \end{array} \tag{5.22}$$

where ($\sim$) identifies that the surrogate model g^*_{a1} is of lower dimension than suggested by an eigenvalue analysis. The surrogate model g^*_{a1}, here a GPR model, is optimized 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 5.21a.



(a) Surrogate model in the 2-D active subspace for drag including lift constraint.

(b) Surrogate model in the 2-D active subspace for lift. The dashed line indicates the constraint value.

(c) Surface contours of pressure coefficient for the sampled drag optimum.
$C_D = 0.0101$, (14.4% reduction)
$C_L = 0.2786$, (2.7% reduction)

Figure 5.21: Results for a lift-constrained drag minimization in active subspaces.

With the location of this optimum identified in the 2-D drag active subspace, an inverse map is used 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 5.21a with a GPR surrogate model, (a_2) a 5-D drag space with a quadratic surrogate model, and (b) the 1-D lift space with a linear surrogate model. The optimization problem which solves this map is shown in the equation below.

$$
\begin{aligned}
\text{given} \quad & \mathbf{y}_{a1} = \mathbf{y}_{select} \\
& C_D(\mathbf{x}) \approx g^*_{a2}(\mathbf{y}_{a2}) \\
& C_L(\mathbf{x}) \approx g^*_{b}(\mathbf{y}_{a}) \\
& \mathbf{y}_{a1} = \boldsymbol{U}_{a1}^{\top}\mathbf{x}, \;\; \boldsymbol{U}_{a1} \in \mathbb{R}^{m \times n_a} \\
& \mathbf{y}_{a2} = \boldsymbol{U}_{a2}^{\top}\mathbf{x}, \;\; \boldsymbol{U}_{a2} \in \mathbb{R}^{m \times n_a + \ell} \\
& \mathbf{y}_{b} = \boldsymbol{U}_{b}^{\top}\mathbf{x}, \quad \boldsymbol{U}_{b} \in \mathbb{R}^{m \times n_b} \\
\underset{\mathbf{x}}{\text{minimize}} \quad & g^*_{a2}(\boldsymbol{U}_{a2}^{\top}\mathbf{x}) \\
\text{subject to} \quad & lb_i < \mathbf{x}_i < ub_i, \; i \in \{0, ..., m\} \\
& \mathbf{y}_{a1} = \boldsymbol{U}_{a1}^{\top}\mathbf{x} \\
& g^*_{b}(\boldsymbol{U}_{b}^{\top}\mathbf{x}) \leq c \\
\text{yield} \quad & \mathbf{x}
\end{aligned}
\tag{5.23}
$$

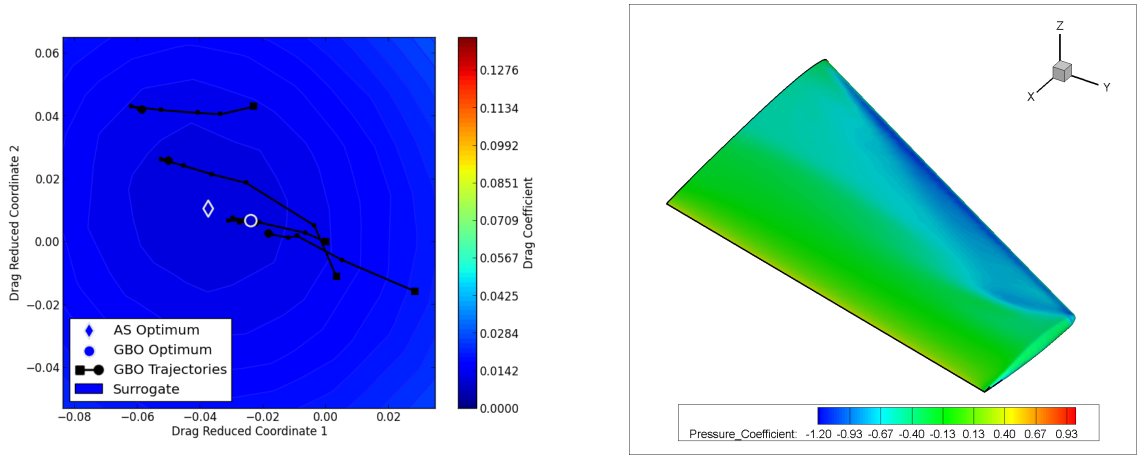
There are two active subspace bases $\boldsymbol{U}_{a1}$ and $\boldsymbol{U}_{a2}$ for drag, and one basis $\boldsymbol{U}_b$ for lift. It finds a location in the full space which maps to the chosen optimum point $\mathbf{y}_{a1}$ while minimizing the estimated drag g^*_{a2}, and constraining the estimated lift g^*_b.

With a potential design identified in the full space, the next step is to run an additional CFD solution. The resulting surface-plot of pressure coefficient is found in Figure 5.20b. 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. 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.

Comparison to Gradient Based Optimization

To compare the quality of the estimated minimum found by the active subspace method, four gradient based analyses were performed 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 5.22a

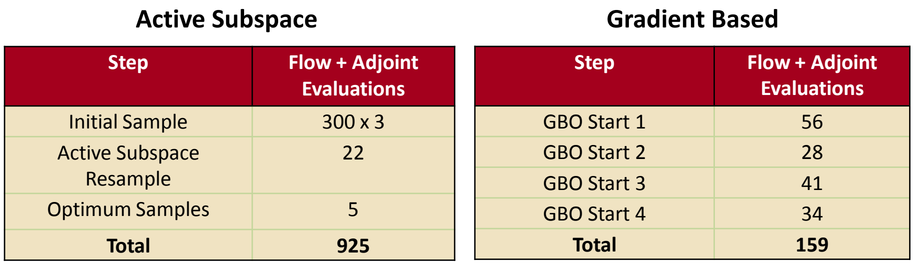


(a) Surrogate model in the 2-D active subspace for drag without lift constraint.

(b) Surface contours of pressure coefficient for the sampled drag optimum.
$C_D = 0.0089$, (24.6% reduction)
$C_L = 0.2868$, (0.1% increase)

Figure 5.22: Gradient based optimization trajectories compared to active subspaces.

As with the motivating example in Section 5.3.1, 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



Active Subspace

Step	Flow + Adjoint Evaluations
Initial Sample	300 x 3
Active Subspace Resample	22
Optimum Samples	5
Total	**925**

Gradient Based

Step	Flow + Adjoint Evaluations
GBO Start 1	56
GBO Start 2	28
GBO Start 3	41
GBO Start 4	34
Total	**159**

Table 5.1: Comparision of computational cost for Active Subspace- and Gradient-Based Optimization for the ONERA M6 problem.

plotted in Figure 5.22b. 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.

A comparison of cost for the two approaches are shown in Table 5.1. To be able to compare a global optimization approach, more GBO random starts 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.

5.3.3.4 Biparabolic Airfoil - Drag Minimization with Equivalent Area Constraint

This is a major exmple of the thesis and demonstrates the usefulness of active subspaces in surrogate based optimization. It uses the biparabolic test case to minimize drag with an equivalent area constraint.

The setup for this optimization is to first pick an active subspace dimensionality for each quantity of interest. After inspecting the eigenvalues and the testing errors of surrogates in various active subspace dimensions, a basis of 10 for both drag and equivalent area was chosen. Then applied is the optimization strategy introduced by Equation 5.15 to manage the mapping between different objective spaces. Below, Equation 5.24 applies this to the current design problem. The inactive space is weakly regularized using a constraint on the z-space distance for each quantity of interest. The optimization problem is run through a global optimizer, again CMA, and then confirmed the results with local refinement using SLSQP. The resulting design vector is sent to simulation to start a full space optimization problem with GBO.

$$
\begin{array}{rl}
\text{given} & C_D(\mathbf{x}) \approx g^*_{C_D}(\boldsymbol{U}^T_{C_D}\mathbf{x}),\ \boldsymbol{U}_{C_D} \in \mathcal{R}_{20\times 10} \\
 & \Delta A_e(\mathbf{x}) \approx g^*_{\Delta A_e}(\boldsymbol{U}^T_{\Delta A_e}\mathbf{x}),\ \boldsymbol{U}_{\Delta A_e} \in \mathcal{R}_{20\times 10} \\
\underset{\mathbf{x}}{\text{minimize}} & g^*_{C_D}(\boldsymbol{U}^T_{C_D}\mathbf{x}) \\
\text{subject to} & 0.03 < \mathbf{x}_i < 0.03,\ i \in \{1, ..., 20\} \\
 & g^*_{\Delta A_e}(\boldsymbol{U}_{\Delta A_e}\mathbf{x}) \leq c \\
 & ||\boldsymbol{V}^T_{C_D}\mathbf{x}||_2 \leq z_{C_D} \\
 & ||\boldsymbol{V}^T_{\Delta A_e}\mathbf{x}||_2 \leq z_{\Delta A_e}
\end{array}
\tag{5.24}
$$

There were 10 different configurations of the inactive space constraint to choose this full space GBO start point. The configuration of these inactive constraints are shown in Table 5.2. The result of the GBO trajectories is then shown in Figure 5.23.

These results are promising because some of the samples, which correlated to a intermediately relaxed z-space constraint around 0.1, were able to find a drag reduction while strongly enforcing the equivalent area constraint. As shown by the random start GBO experiment in Section 5.3.1, this would have seemed to be an unapproachable

Table 5.2: Configuration of z-space constraints.

Sample Index	$\mathbf{z}_{C_D}$	$\mathbf{z}_{\Delta A_e}$
1	0.001	0.001
2	0.010	0.001
3	0.100	0.001
4	1.000	0.001
5	0.001	0.010
6	0.001	0.100
7	0.001	1.000
8	0.010	0.010
9	0.100	0.100
10	1.000	1.000

result with GBO.

In the case of Sample 6 that constrained equivalent area and reduced drag the most, a summary of the flow and nearfield results is available in Figure 5.24. Note that because of the coarse geometric parameterization, the optimizer was not able to completely remove shocks on the upper part of the airfoil. More FFD control points are needed to provide enough fidelity to maintain the lower airfoil shape while modifying the upper shape, since control points affect both the upper and lower surfaces.

This is a major result because it demonstrates the usefulness of Surrogate Based Optimization with Active Subspaces. Even though the surrogate model is built in a reduced parameterization, it is enough information to discover a region that may contain a global minimum.

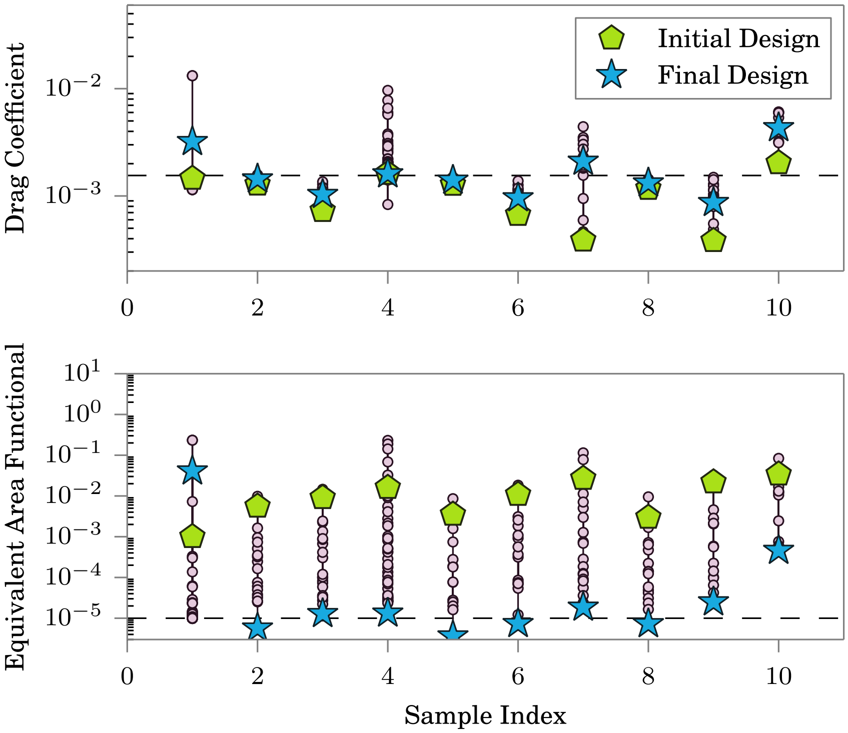
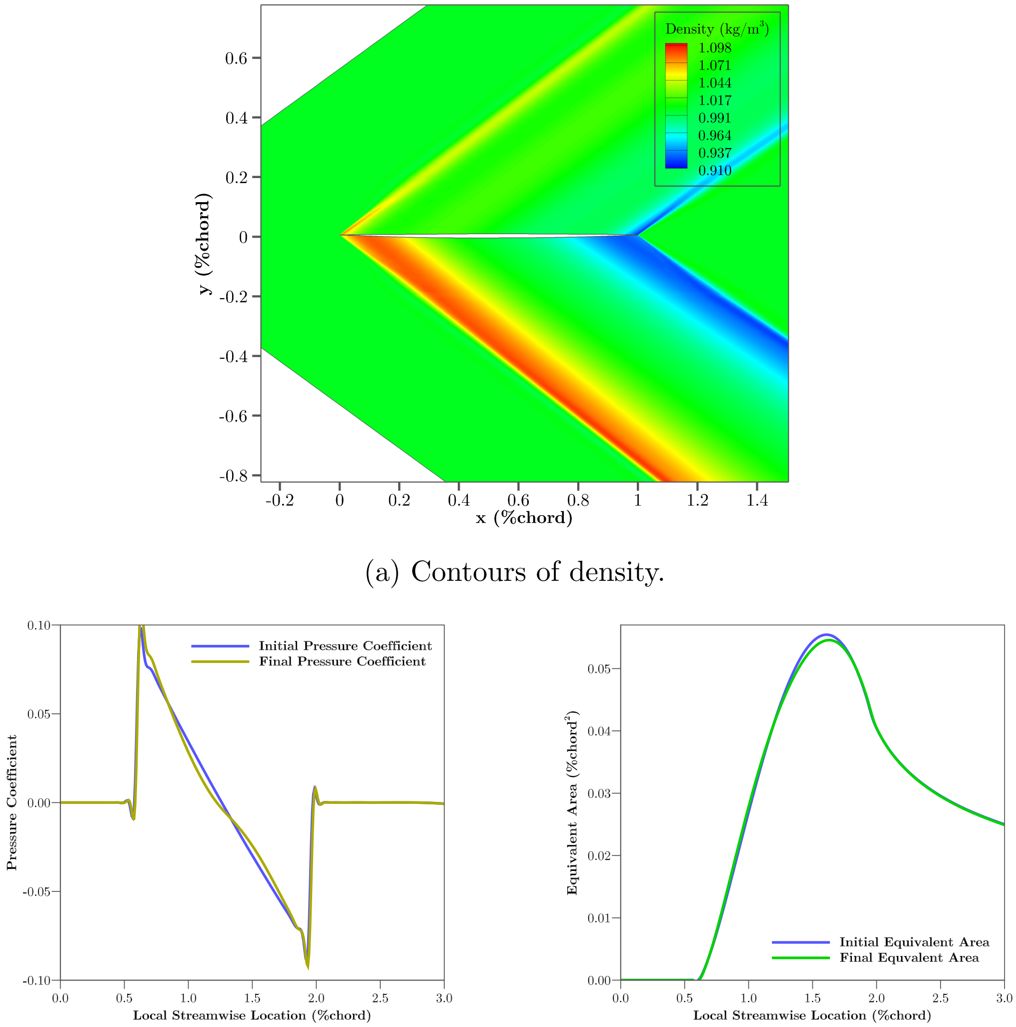


Figure 5.23: Summary of GBO optimizations started from locations predicted by surrogate based optimization in active subspaces, for various configurations of the inactive space constraint.



(a) Contours of density.

(b) Pressure coefficient at the nearfield marker.

(c) Equivalent area at the nearfield marker.

Figure 5.24: Results for a surrogate based optimization in active subspaces, used to start a gradient based optimization. This result corresponds to sample 6 of Figure 5.23.

5.3.3.5 Langley Supersonic Business Jet - Drag Minimization with Lift and Equivalent Area Constraints

In this capstone optimization example of the thesis, a surrogate based optimization problem is constructed in the active subspace to minimize drag with two constraints, while avoiding a non-linear optimization in the 198 dimensional full space. The problem is written below.

$$
\begin{array}{cl}
\text{given} & C_D(\mathbf{x}) \approx g^*_{C_D}(\boldsymbol{U}^T_{C_D}\mathbf{x}) = g^*_{C_D}(\mathbf{x}_{C_D}),\ \boldsymbol{U}_{C_D} \in \mathcal{R}_{198\times 7} \\
 & C_L(\mathbf{x}) \approx g^*_{C_L}(\boldsymbol{U}^T_{C_L}\mathbf{x}) = g^*_{C_L}(\mathbf{y}_{C_L}),\ \boldsymbol{U}_{C_L} \in \mathcal{R}_{198\times 5} \\
 & \Delta A_e(\mathbf{x}) \approx g^*_{\Delta A_e}(\boldsymbol{U}^T_{\Delta A_e}\mathbf{x}) = g^*_{\Delta A_e}(\mathbf{y}_{\Delta A_e}),\ \boldsymbol{U}_{\Delta A_e} \in \mathcal{R}_{198\times 5} \\
\underset{\mathbf{y}_{C_D},\mathbf{y}_{C_L},\mathbf{y}_{\Delta A_e}}{\text{minimize}} & g^*_{C_D}(\mathbf{y}_{C_D}) \\
\text{subject to} & -0.4 < y_{C_D,i} < 0.4,\ i \in \{1,...,7\} \\
 & -0.4 < y_{C_L,i} < 0.4,\ i \in \{1,...,5\} \\
 & -0.4 < y_{\Delta A_e,i} < 0.4,\ i \in \{1,...,5\} \\
 & g^*_{\Delta A_e}(\mathbf{y}_{\Delta A_e}) \leq 1 \times 10^{-2} \\
 & g^*_{C_L}(\mathbf{y}_{C_L}) \geq 0.10 \\
 & c(\mathbf{y}_{C_D}, \mathbf{y}_{C_L}, \mathbf{y}_{\Delta A_e}) \leq 1 \times 10^{-3}
\end{array}
\tag{5.25}
$$

In this problem there are now three active subspaces. The dimension of these are chosen according to the heuristics presented in Chapter 4. The behavior of the optimization is very sensitive to the dimension of these subspaces, as the spread of the samples has an large effect on the behavior of the surrogate. In practice several configurations of the active subspace dimensions were chosen. The configuration presented here represents the final result.

Another important parameter to adjust is the bounds of the active subspace variables. The bounds for this problem were chosen by inspecting a 2-D scatter plot of the samples projected into the active subspace of the first two eigenvectors. A threshold of 0.4 created a boundary that left a comfortable amount of data aroud the boundary. The main goal here is to avoid leaving a large distance between the bounding hull of the data projected in the active space, and the optimization boundary.

The consistency constraint was formulated in Equation 5.18. The result of this function is essentially a norm in the scaled activespace, and should be less than zero. However a small threshold of 1×10^{-3} is used in order to allow optimization programs to explore designs. Similarly a small threshold is used for the Equivalent Areas.

The resulting deformations of this optimization is visualized, before sending the design to CFD for a flow solution. This enables the user to test a few configurations on the optimization problem, like active subspace dimension, and omit clearly infeasible or unlikely optimized designs. Several designs can be generated and checked by eye. Then the designs are sent to a server that evaluates the various performance metrics.

The deformation for the best design of this study is presented in Figure 5.25. It is colored by displacement in Z-direction. Because the FFD paramterization is setup such that the control points only move up and down, the only component of displacement that is expected is the z-displacement, which is plotted via color contours. The baseline shape is also shown in transparent gray.

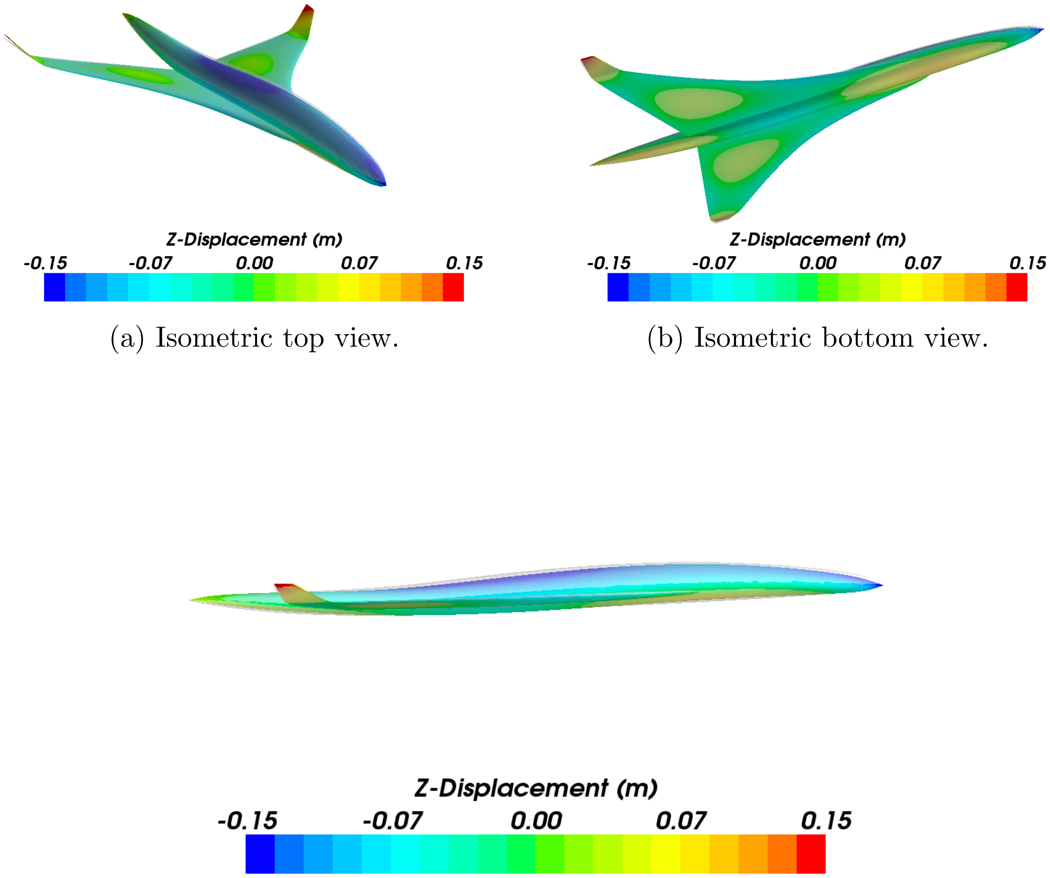
The result shows a few physically expected motions that encouraged the design to be retained for flow valuation. The major change is the fuselage, which is deflated to minimize the wave-component of drag. This is especially apparent in the right view in Figure 5.25c. A subtle trade is being made between twist and camber of the wing. In trying to maintain equivalent area, the bottom surface of the wing was largely unmoved, as shown in Figure 5.25b. Furthermore in Figure 5.25a, the light blue regions of the leading edge indicate that it has drooped the leading edge, while green regions at the mid-chord indicate only small movements. A large amount of the design problem is likely being actuated here, as the deformation modes showed that all three quantities of interest have modes including twist and camber with varying degrees of importance.

The flow solutions for this design are presented in Figure 5.26a. The optimized design reduced a strong pressure gradient on the upper surface of the wing. Additionally a plot of equivalent area distribution in the nearfield is given in Figure 5.26b. Two azimuthal locations are plotted, zero (on the symmetry plane), and sixty degrees. The nearfield equivalent area is not tightly respected, however it could be seen as an approximate results and one that could be further refined by employing gradient based optimization in the full space.

Table 5.3: Vehicle performance results for the optimization problem.

Metric	Baseline	Optimized
C_D	0.0082	0.0070
C_L	0.10	0.98
ΔA_e	0.0	3.9

The summary of performance data is presented in Table 5.3. Using surrogate models built in three and five dimensional active subspaces, the optimization algorithm was able to identify the potential to reduce drag by fourteen percent, while maintaining lift, and allowing a noticeable violation of the equivalent area functional. In doing so it has created a cabin that would not be amenable to carrying passengers. This is a constraint that could be added in future work.

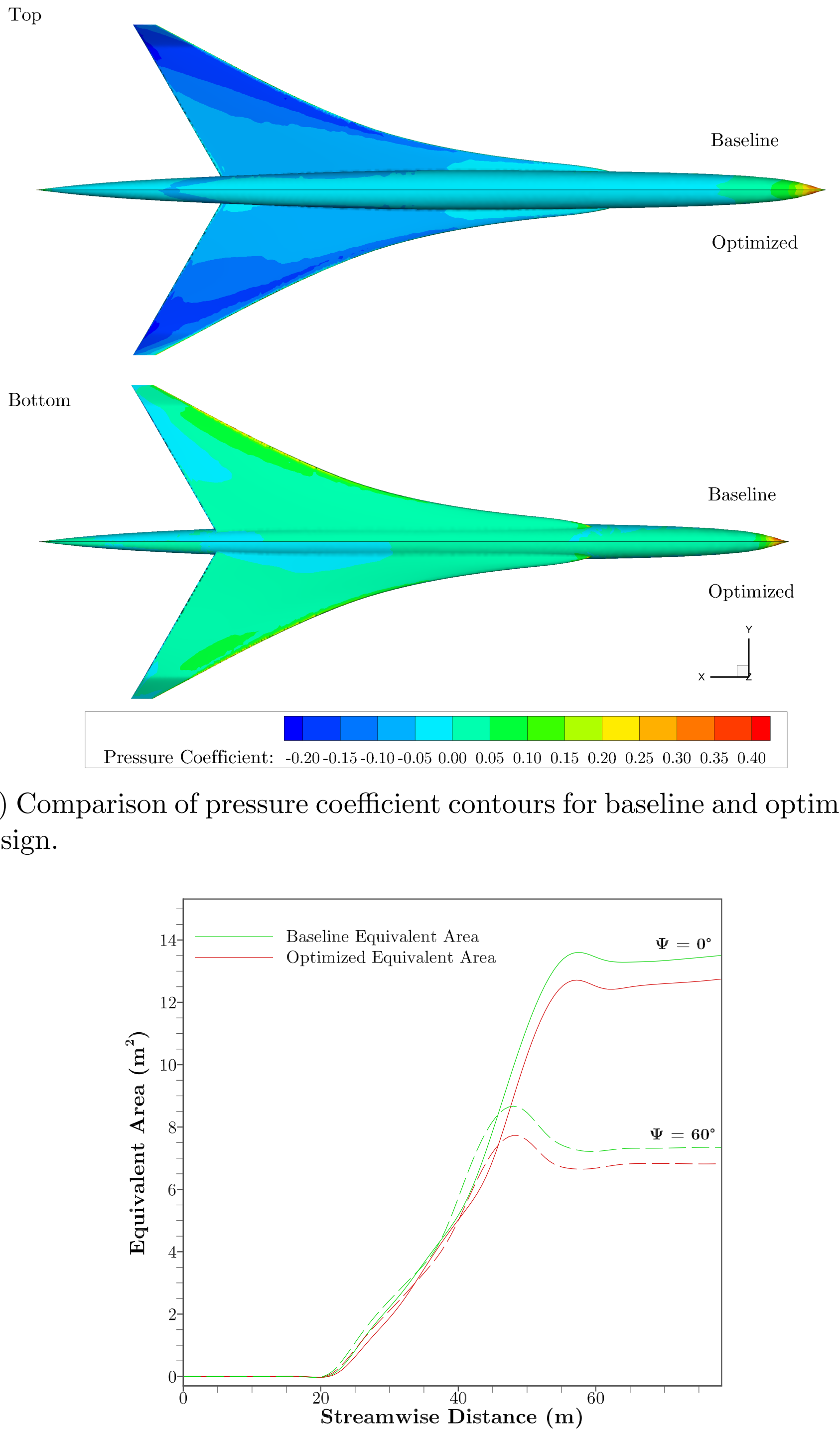


(a) Isometric top view.

(b) Isometric bottom view.

(c) Right view.

Figure 5.25: Deformation of SBO estimated optimum.



(a) Comparison of pressure coefficient contours for baseline and optimized design.

(b) Comparison of equivalent area distributions for baseline and optimized design.

Figure 5.26: Comparison of baseline and optimized designs.

5.4 Summary

This chapter presented two approaches to surrogate based optimization for aerospace design problems. The first approach used a surrogate model based on a gradient enhanced Gaussian process regression, using two infill sampling criteria for adaptive refinement of constrained optimization problems. This was applied to various analytic and CFD based problems to demonstrate the ability to accelerate the convergence of an optimization problem when design evaluations are the rate determining step. A contribution here is in the integration of several ideas related to efficient global optimization with Gaussian process regression to increase the robustness of the adaptive refinement procedure.

The first approach was found only suitable for low dimensional design problems, and only if the quantities of interest had similar amounts of variation along each input dimension. In the case of high dimension, or if a subspace of inputs had very little effect on the objective, the surrogate modeling procedure was shown to suffer.

The second approach addressed these issues by applying the active subspace method to surrogate based optimization. It had been shown in the previous chapter that active subspace surrogate models sufficiently approximate high dimensional design problems in a reduced linear subspace. When applied to aerospace design problems, it was found that surrogate models built in active subspaces could predict the locations near local minimum, and when coupled with a gradient based optimization, enabled the optimization of problems otherwise intractable via gradient based optimization alone.

Chapter 6

Conclusion

The primary aim of this dissertation was to develop approaches that can bring high-fidelity design optimization in to the preliminary design phase through the use of surrogate based optimization and dimensionality reduction. Its contributions take a step towards improving the ability of surrogate modeling to handle large numbers of input variables. In doing so it has also discovered several useful tools for understanding design behaviors. This chapter will conclude the dissertation by summarizing the approach, highlighting the contributions, and outlining areas for future work.

6.1 Summary

This dissertation visited several classes of design problems. They were described in terms of their setup, baseline solutions, and optimization formulations. There were three classes of examples – analytic functions, two dimensional flow problems, and three dimensional flow problems. The analytic examples contained standard test functions like the Rosenbrock and Rastrigin functions, as well as two functions constructed to demonstrate active subspaces. The two dimensional flow problems included the NACA 0012 and Biparabolic airfoils. The three dimensional flow problems included the ONERA M6 wing, the Langley supersonic business jet, and the N+2 supersonic passenger jet.

Several approaches were developed for surrogate modeling to enable a robust regression procedure in the face of sparse data and inaccurate gradients. To improve the hyperparameter learning process, a series of constraints were developed to encourage the problem to land in a region of the marginal likelihood surface that was physically representative of the data. The issue of gradient inaccuracies was then presented, including an analysis of where it impacts the surrogate modeling process. To characterize the gradient inaccuracies from adjoint methods used in this dissertation, a surrogate-based gradient estimation technique was applied by analytically solving 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 the estimation of a root-mean-squared error of the gradients across the entire design space. These results were used as motivation to build upon common noise model for GPR, modified to enable the independent treatment of objective and gradient noise terms, and constrained during the hyperparameter learning process to ensure that objective information was honored before gradient information.

Next, a method was presented for reducing the dimensionality of high-dimensional aerospace problems based upon the active subspace method. This involved identifying and retaining a subset of eigenvectors from an eigenvalue analysis of the estimate of the average outer product of the problem's gradients. Several heuristics were suggested to enable the selection of a reasonable number of active subspace dimensions. The training errors of surrogate models built in active subspaces of varying dimension were investigated. Across the two- and three-dimensional supersonic design problems presented in this dissertation, all three were shown to have collapsed linear behaviors for lift in one dimension, described with an angle of incidence mode on the lifting surfaces. Across all three problems, equivalent area was well described in one dimension and had a predominantly quadratic behavior. Across all three problems, the behavior of drag coefficient required several active subspace dimensions, anywhere between five and ten.

Finally, two approaches to surrogate based optimization were presented for aerospace design problems. The first approach used a surrogate model based on a gradient enhanced Gaussian process regression, using two infill sampling criteria for adaptive refinement of constrained optimization problems. This was applied to analytic and CFD-based problems to demonstrate the ability to accelerate the convergence of an optimization problem when design evaluations are the rate-determining step. The second approach extended surrogate based optimization in to high dimension, on the order of 200 variables, by using the active subspace method. When applied to aerospace design problems, it was found that surrogate models built in active subspaces could predict locations near local minimum, and when coupled with a gradient based optimization, enabled the optimization of problems otherwise intractable via gradient based optimization alone.

6.2 Contributions

This section will provide a concluding summary of the contributions made through this dissertation.

First, it has advanced the perspective that inaccurate gradients can and should be taken into account within gradient enhanced surrogate modeling. As explored by Forester [76], noise models are able to account for inaccuracies encountered in objective functions obtained from numerical simulations. Shown in Section 3.3.2.1, gradient bias errors naturally present in continuous adjoint formulations warrant the separate treatment of objective and gradient training data. To that end a traditional noise model based for GPR was modified to enable the independent treatment of objective and gradient noise terms, and constrained during the hyperparameter learning process to ensure that objective information was honored before gradient information. Complimenting this contribution is the integration of a series of hyperparameter constraints that encourage the learning problem to enable a robust regression procedure in the face of sparse data and inaccurate gradients. Development of these approaches was largely presented in Section 3.2.4, and verification of the approach was especially demonstrated in Section 3.3.2.2.

Second, this dissertation builds upon the a traditional refinement strategy based on the expected improvement function, to improve the sample efficiency of surrogate based optimization. One component of this contribution was modifying the original expected improvement formulation of Jones [119] with the introduction of an *a-priori* conditioning that avoids selecting new samples in the bounds and corners of the design space. The second component of this contribution was the development of a two-phase strategy that starts with a global refinement based on the modified expected improvement, and then a local refinement based on pure exploitation by finding the predicted minimum. A heuristic was developed for transitioning from the first phase to the second phase, and declaring convergence in the second phase. The approaches were developed in Section 5.2.1, and were exercised on both analytic and simulation based optimization problems in Section 5.3.2.

Third, in applying the active subspace method [99] this dissertation discovers the presence of active subspaces for supersonic design problems. These subspaces built from linear combinations of inputs reasonably modeled objective behavior of the lift coefficient, drag coefficient and equivalent area functional in supersonic design problems. An important result was connecting the behavior of active subspaces to classical supersonic theory in Section 4.3.1.2. Several coherent physical features were found across several design problems in both two and three dimensions, especially in Sections 4.3.1.1, 4.3.2.2, and 4.3.3.2. The coherence of these results suggests the viability of the active subspace method in aerospace design problems. Despite building very high parameterizations of the problem, there exist reduced spaces in which a surrogate model could be applied for optimization.

Fourth, an algorithm is proposed for optimization in high-dimension with surrogate models built in reduced dimension using active subspaces. Several inverse maps were developed heuristically in Section 5.2.2 that enable the linking of separate active subspaces for objectives and constraints, enabling surrogate based optimization in high dimension. Several design problems were explored, for example in Sections 5.3.3.4 and 5.3.3.5, and it was shown that surrogate based optimization in active subspaces could enable the optimization of problems otherwise intractable via gradient based optimization alone.

Finally, many of the algorithms of this dissertation have been deposited in open source. Interfaces for automatic evaluation of flow simulations were developed as part of the SU2 simulation suite [124]. A new package for Gaussian process regression called VyPy was created as part of this dissertation [86]. Lastly, development input was provided to a package called PAUL developed by Constantine for the active subspace method [103].

6.3 Future work

There are several areas available for future work and extension of this dissertation.

A formal mathematical investigation of the inverse mappings for optimization in active subspaces is an important area for future work. In this dissertation, the inverse maps were developed heuristically. A formal treatment of the problem could identify mappings that include more appropriate consideration of the effects of this approach to dimensionality reduction. For example, Constantine has suggested a resampling method based on a Markov chain Monte Carlo in the active subspace [99]. Developments in this area can identify new mappings and advance an understanding of the conditions under which they are viable.

Using active subspaces to pre-condition Gaussian process regression could be an exciting direction in which to extend this work. The full eigenvector and eigenvalue decomposition from the active subspace method can be used to rotate and scale the inputs of the GPR covariance matrix, which would allow regression to be applied on the full space including gradients, but with the reduced number of hyperparameters suggested in this dissertation. It would be a more elegant combination of the active subspace method and GPR and would further allow the use of adaptive refinement procedures. The possibility for this has been identified at by Constantine, who was able to estimate anisotropic hyperparameters based on the active subspace methods [55]. A potential difficulty that could be encountered in this endeavor is managing the size of the covariance matrix with large numbers of samples in high dimension with gradient information. In the case of the Langley business jet problem, which had 198 dimensions and 172 samples with gradients, the covariance matrix for would be of size 34228 × 34228. As suggested by Yamazaki, parallelization of the GPR equations can be beneficial [41]. Similarly here, consideration for the runtime of the regression techniques will be needed, and the application of large scale linear solvers could be explored.

This dissertation could be extended to model surrogates involving varying Mach number as an input, with a domain that extends from subsonic to supersonic regimes. This would introduce an important transition in the transonic region that will be difficult to model under stationary surrogate models. Methods for decomposing the design space via Treed Kriging as explored by Nelson [71], or for warping the design space via non-stationary covariance functions as explored by Paciorek [125], could extend active subspaces with surrogate based optimization into these more complex design problems.

Finally, surrogate models built in active subspaces could be applied to mission level aircraft design analyses, such as SUAVE [126], to enable mission-level optimization with high-fidelity analyses of next generation aircraft.

Appendix A

Appendix

This appendix documents fundamentals for supersonic thin airfoil theory, and the freeform deformation methodology.

A.1 Supersonic Thin Airfoil Theory

Assuming small deflections, which would result from a thin airfoil at small angles of attack.

$$\begin{aligned} \alpha &\approx 0 \\ \tau &<< c \end{aligned} \tag{A.1}$$

Linear supersonic potential theory can be used to find analytic performance results for thin airfoils. A major result of thin airfoil theory is that the pressure change, and thus the pressure coefficient, on the surface of the airfoil is dependent on the surface gradient. As shown by Anderson [127], the following relationships are attainable under these assumptions.

Given an airfoil with upper and lower surfaces, normalized on a unit chord,

$$\begin{gathered} y_u(x) \\ y_l(x) \\ x \in [0,1], \end{gathered} \tag{A.2}$$

then thin airfoil theory predicts that the upper and lower pressure coefficients C_{p_u} and C_{p_l} are modeled according to

$$C_{p_u} = +\frac{2}{\sqrt{M_\infty - 1}}\frac{\partial y_u}{,}\partial x \tag{A.3}$$

and

$$C_{p_l} = -\frac{2}{\sqrt{M_\infty - 1}}\frac{\partial y_l}{\partial x}. \tag{A.4}$$

Here, M_∞ is the freestream Mach number, and $\frac{\partial y_u}{,}\partial x$ and $\frac{\partial y_l}{\partial x}$ are the surface gradients of the upper and lower airfoil respectively. Several performance metrics can be found by integrating the pressure coefficients over the airfoil surface.

A.1.0.6 Lift Coefficient

The sectional lift coefficient c_l is found by integrating the pressure coefficient,

$$c_l = \int_0^1 C_{p_l}(x) - C_{p_u}(x)dx, \tag{A.5}$$

where the above takes the airfoil as normalized on a unit chord length. The integral simplifies to,

$$c_l = \frac{-2}{\sqrt{M_\infty - 1}}\left[y_l(1) - y_l(0) + y_u(1) - y_u(0)\right]. \tag{A.6}$$

Knowing that the airfoil upper and lower surfaces meet at the tips of the airfoil, and restricting to small angles of attack, there is the canonical result,

$$c_l = \frac{4}{\sqrt{M_\infty - 1}}\alpha \tag{A.7}$$

which is a statement that lift for supersonic thin airfoils at a constant Mach number is only dependent on angle of attack.

A.1.0.7 Drag Coefficient

The sectional drag coefficient can be found similarly, starting with

$$\begin{aligned} c_d &= \int_0^1 C_{p_u}(x)\frac{\partial y_u}{\partial x} - C_{p_l}(x)\frac{\partial y_l}{\partial x}dx \\ &= \frac{2}{\sqrt{M_\infty - 1}}\int_0^1 \left(\frac{\partial y_u}{\partial x}\right)^2 - \left(\frac{\partial y_l}{\partial x}\right)^2 dx \end{aligned} \tag{A.8}$$

The remaining surface derivatives mean that the drag coefficient is dependent on the shape of the airfoil. This dissertation explores a biparabolic airfoil. Here the baseline shape for this analytic example will be parameterized only by maximum thickness τ, and angle of attack α.

$$\begin{aligned} y_u &= -\tau(x^2 - x) - \alpha(x - \tfrac{1}{2}) \\ y_l &= +\tau(x^2 - x) - \alpha(x - \tfrac{1}{2}) \end{aligned} \tag{A.9}$$

$$\begin{aligned} \tfrac{\partial y_u}{\partial x} &= -2\tau(x-1) - \alpha \\ \tfrac{\partial y_l}{\partial x} &= +2\tau(x-1) - \alpha \end{aligned} \tag{A.10}$$

Incorporating this into the equation for drag coefficient, the integral becomes,

$$c_d = \frac{2}{\sqrt{M_\infty - 1}}\int_0^1 (-2\tau(x-1) - \alpha)^2 + (+2\tau(x-1) - \alpha)^2 dx \tag{A.11}$$

and after integrating yields,

$$c_d = \frac{4\left(\frac{4}{3}\tau^2 + \alpha^2\right)}{\sqrt{M_\infty - 1}} \tag{A.12}$$

which says that drag coefficient is dependent on the square of thickness and square of angle of attack.

A.1.0.8 Equivalent Area

Supersonic potential theory also provides a relationship for equivalent area, which is strongly tied to the change in boom loudness at the ground.

The equivalent area relationship for a two-dimensional problem is given as,

$$A_e(x) = \frac{4\sqrt{2\beta r}}{\gamma p_\infty M_\infty^2} \int_0^x (p(t) - p_\infty)(x-t)^{\frac{1}{2}} dt, \tag{A.13}$$

where $\beta = \sqrt{M_\infty^2 - 1}$ is the Prandtl-Glaurert factor. The integral is evaluated along a line of interest parallel to the freestream, at a distance r below the body [35].

In linear supersonic potential theory, pressure disturbances propagate along characteristics aligned with the Mach number, so the pressure distribution of the airfoil will be equivalent to the pressure distribution sampled on the lower surface of the airfoil, displaced downstream along a line with an angle of $\arcsin\left(\frac{1}{M_\infty}\right)$ against freestream.

Bringing in this information to the equivalent area relationship, the following integral is built,

$$\begin{aligned} A_e(x) &= \sqrt{2\beta r} \int_0^x C_{p_l}(t)(x-t)^{\frac{1}{2}} dt \\ &= \sqrt{2\beta r} \int_0^x (2\tau(2t-1) - \alpha)(x-t)^{\frac{1}{2}} dt, \end{aligned} \tag{A.14}$$

which for an airfoil of unit chord length integrates to:

$$A_e(x) = \sqrt{2\beta r} \frac{2}{15} x^{\frac{3}{2}} \left(2\tau(4x-5) - 5\alpha\right), x \in [0,1]. \tag{A.15}$$

A cost function is formulated based on the squared error of the equivalent area [31] with:

$$\Delta A_e = \frac{1}{c} \int_0^c \left(A_e(x) - A_t(x)\right)^2 dx \tag{A.16}$$

where the goal is to minimize the difference between the current equivalent area $A_e(x)$ and the target equivalent area $A_t(x)$.

Using the parameterization of the biparabolic airfoil given above, and setting an equivalent area based on an airfoil with a given thickness τ_t and angle of attack α_t, integrate for a unit chord length,

$$\Delta A_e = \sqrt{2\beta r} \int_0^1 \frac{2}{15} x^{\frac{3}{2}} \left(2\tau(4x-5) - 5\alpha\right) - \frac{2}{15} x^{\frac{3}{2}} \left(2\tau_t(4x-5) - 5\alpha_t\right) dx \tag{A.17}$$

to find a cost function that behaves as:

$$\Delta A_e = \frac{\sqrt{2\beta r}}{675}(44(\tau-\tau_t)^2 + 108(\tau-\tau_t)(\alpha-\alpha_t) + 75(\alpha-\alpha_t)^2), \tag{A.18}$$

which has a paraboloid relationship on τ and α.

A.1.1 Freeform Deformation

The actual deformation of the surface is carried out by a separate geometry parametrization. In this paper, a Free-Form Deformation (FFD) strategy [58] 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

$$X(u, v, w) = \sum_{i,j,k=0}^{l,m,n} P_{i,j,k} B_j^l(u) B_j^m(v) B_k^n(w), \tag{A.19}$$

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. A.19.

Bibliography

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

[2] “Protecting Yourself from Noise in Construction,” Safety Pocket Guide 3498-12N, Occupational Health and Safety Administration, 2011.

[3] Woolley, P., “A cost-benefit analysis of the Concorde project,” *Journal of Transport Economics and Policy*, 1972.

[4] Jones, L., “Lower bounds for sonic bangs,” *Journal of the Royal Aeronautical Society*, Vol. 65, 1961.

[5] McLean, F. E., “Some nonasymptotic effects on the sonic boom of large airplanes,” Tech. Rep. NASA TN D-2877, National Aeronautics and Space Administration, 1965.

[6] Seebass, R., “Sonic Boom Theory,” *Journal of Aircraft*, Vol. 6, 1969.

[7] George, A. and Seebass, R., “Sonic boom minimization including both front and rear shocks,” *AIAA Journal*, Vol. 9, No. 10, 1971.

[8] Whitham, G., “The flow pattern of a supersonic projectile,” *Communications on pure and applied mathematics*, Vol. 5, No. 3, 1952.

[9] Hayes, W. D., *Linearized supersonic flow*, Ph.D. thesis, California Institute of Technology, 1947.

[10] Seebass, R., "Minimum sonic boom shock strengths and overpressures," *Nature*, Vol. 221, 1969.

[11] Darden, C. M., "Minimization of sonic-boom parameters in real and isothermal atmospheres," Tech. Rep. NASA TN D-7842, National Aeronautics and Space Administration, 1975.

[12] Thomas, C., "Extrapolation of sonic boom pressure signatures by the waveform parameter method," Tech. Rep. NASA TN D-6832, National Aeronautics and Space Administration, 1972.

[13] Plotkin, K. J. and Grandi, F., "Computer models for sonic boom analysis: PCBoom4, CABoom, BooMap, CORBoom," *Wyle Report WR*, Vol. 211, 2002.

[14] Durston, D. A., "A preliminary evaluation of sonic boom extrapolation and loudness calculation methods," Tech. Rep. NASA CP 10133, National Aeronautics and Space Administration.

[15] Morgenstern, J. and by Institutions, F., "Optimum Signature Shaping for Low Sonic Boom," *30th Applied Aerodynamics Conference*, AIAA Paper 2012-3218, New Orleans, LA, June 2012.

[16] 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.

[17] Haas, A. and Kroo, I., "A multi-shock inverse design method for low boom supersonic aircraft," *48th AIAA Aerospace Sciences Meeting*, AIAA Paper 2010-843, Orlando, FL, January 2010.

[18] Morgenstern, J., Buonanno, M., Chai, S., and Marconi, F., "Overview of Sonic Boom Reduction Efforts on the Lockheed Martin N+ 2 Supersonic Validations Program," *32nd Applied Aerodynamics Conference*, AIAA Paper 2014-2138, Atlanta, GA, June 2014.

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

[20] 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.

[21] Morgenstern, J. M., Arslan, A., Lyman, V., and Vadyak, J., "F-5 shaped sonic boom demonstrators persistence of boom shaping reduction through turbulence," *43rd AIAA Aerospace Sciences Meeting*, AIAA Paper 2005-12, Reno, NV, January 2005.

[22] Shepherd, K. P. and Sullivan, B. M., "A loudness calculation procedure applied to shaped sonic booms," Tech. Rep. NASA-TP-3134, National Aeronautics and Space Administration, 1991.

[23] Johnson, D. and Robinson, D., "Procedure for calculating the loudness of sonic bangs," *Acta Acustica united with Acustica*, Vol. 21, No. 6, 1969.

[24] Needleman, K. E., Darden, C. M., and Mack, R. J., "A study of loudness as a metric for sonic boom acceptability," *48th AIAA Aerospace Sciences Meeting*, AIAA Paper 91-0496, Reno, NV, January 1991.

[25] Stevens, S., "Perceived level of noise by Mark VII and decibels (E)," *The Journal of the Acoustical Society of America*, Vol. 51, No. 2B, 1972.

[26] Wintzer, M. and Kroo, I., "Optimization and adjoint-based CFD for the conceptual design of low sonic boom aircraft," *50th AIAA Aerospace Sciences Meeting*, AIAA Paper 2012-0963, Nashville, TN, January 2012.

[27] Li, W. and Shields, E., "Generation of Parametric Equivalent-Area Targets for Design of Low-Boom Supersonic Concepts," *49th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, AIAA Paper 2011-462, Orlando, FL, January 2011.

[28] Ordaz, I. and Li, W., “Adaptive Aft Signature Shaping of a Low-Boom Supersonic Aircraft Using Off-Body Pressures,” *50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, AIAA Paper 2012-0020, Nashville, TN, January 2012.

[29] 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.

[30] Alonso, J. J., Kroo, I. M., and Jameson, A., “Advanced algorithms for design and optimization of quiet supersonic platforms,” January 2002.

[31] 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.

[32] Ordaz, I. and Li, W., “Advanced Usage of Vehicle Sketch Pad for CFD-Based Conceptual Design,” *51st AIAA Aerospace Sciences Meeting*, AIAA Paper 2013-0329, Grapevine ,TX, January 2013.

[33] Aftosmis, M. J., Nemec, M., and Cliff, S. E., “Adjoint-based low-boom design with Cart3D,” *29th AIAA Applied Aerodynamics Conference*, Honolulu, HI, June 2011.

[34] Kroo, I., Willcox, K., March, A., Haas, A., Rajnarayan, D., and Kays, C., “Multifidelity Analysis and Optimization for Supersonic Design,” Tech. Rep. NASA/CR-2010-216874, NF1676L-11896, National Aeronautics and Space Administration, 2010.

[35] Plotkin, K. J., “Review of sonic boom theory,” *12th AIAA Aeroacoustics Conference*, AIAA Paper 89-1105, San Antonio, TX, April 2010.

[36] Plotkin, K. J., "State of the art of sonic boom modeling," *The Journal of the Acoustical Society of America*, Vol. 111, No. 1, 2002.

[37] Kenway, G. and Martins, J., "Multi-point High-Fidelity Aerostructural Optimization of a Transport Aircraft Configuration," *Journal of Aircraft*, 2012.

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

[39] Igel, C., Suttorp, 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.

[40] 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*.

[41] 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.

[42] 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.

[43] Jones, E., Oliphant, T., Peterson, P., et al., "SciPy: Open source scientific tools for Python," `http://www.scipy.org/`, 2001.

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

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

[46] Nadarajah, S. K. and Jameson, A., "A Comparison of the Continuous and Discrete Adjoint Approach to Automatic Aerodynamic Optimization," *AIAA 38th Aerospace Sciences Meeting and Exhibit*, No. AIAA-2000-0667, Reno, NV., January 2000.

[47] 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.

[48] 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.

[49] 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.

[50] 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.

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

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

[53] Koehler, J. and Owen, A., "Computer experiments," *Handbook of statistics*, Vol. 13, No. 13, 1996.

[54] 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.

[55] Constantine, P. G., Dow, E., and Wang, Q., "Active subspace methods in theory and practice: Applications to kriging surfaces," *SIAM Journal on Scientific Computing*, Vol. 36, No. 4, 2014.

[56] Ng, A., "Principal component Analysis," *Stanford Univeristy CS229 Lecture Notes*, 2012.

[57] Ghisu, T., Parks, G., Jarrett, J., and Clarkson, P., "Accelerating Design Optimization Via Principal Components Analysis," *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Victoria, British Columbia, Canada, Sept 2008.

[58] Palacios, F., Alonso, J. J., Duraisamy, K., Colonno, M. R., Aranake, A. C., Campos, A., Copeland, S. R., Economon, 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*, Grapevine, TX, Jan 2013.

[59] Molga, M. and Smutnicki, C., "Test functions for optimization needs," 2005.

[60] Mack, R. J., "A supersonic business-jet concept designed for low sonic boom," Tech. Rep. NASA/TM-2003-212435, L-18307, NASA, 2003.

[61] Rasmussen, C. and Williams, C., *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, MA, 2006.

[62] Box, G. E. and Wilson, K., "On the experimental attainment of optimum conditions," *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 13, No. 1, 1951.

[63] Myers, R. H. and Montgomery, D. C., "Response Surface Methodology: Process and Product in Optimization Using Designed Experiments," 1995.

[64] Schoenberg, I. J. and Schoenberg, I. J., *Cardinal spline interpolation*, SIAM Press, 1973.

[65] Hajela, P. and Berke, L., "Neural networks in structural analysis and design: an overview," *Computing Systems in Engineering*, Vol. 3, No. 1, 1992.

[66] Matheron, G., "Principles of geostatistics," *Economic geology*, Vol. 58, No. 8, 1963.

[67] Cox, D. D. and John, S., "SDO: A statistical method for global optimization," *Multidisciplinary design optimization: state of the art*, 1997.

[68] Giunta, A. A., Balabanov, V., Kaufman, M., Burgee, S., Grossman, B., Haftka, R. T., Mason, W. H., and Watson, L. T., "Variable-complexity response surface design of an HSCT configuration," *Multidisciplinary Design Optimization*, 1997.

[69] Otto, J., Paraschivoiu, M., Yesilyurt, S., and Patera, A. T., "Bayesian-validated computer-simulation surrogates for optimization and design," *Proceedings of the ICASE/NASA Langley Workshop on Multidisciplinary Optimization*, 1997.

[70] Booker, A. J., Dennis Jr, J., Frank, P. D., Serafini, D. B., Torczon, V., and Trosset, M. W., "A rigorous framework for optimization of expensive functions by surrogates," *Structural optimization*, Vol. 17, No. 1, 1999.

[71] Nelson, A., *Treed Kriging Aerodynamic Database Modeling and Optimzation*, Ph.D. thesis, Stanford University, June 2008.

[72] Rajnarayan, D., Haas, A., and Kroo, I., "A multifidelity gradient-free optimization method and application to aerodynamic design," *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2008.

[73] Liu, W., *Development of gradient-enhanced kriging approximations for multidisciplinary design optimization*, Ph.D. thesis, University of Notre Dame, 2003.

[74] Giunta, A. A., Narducci, R., Burgee, S., Grossman, B., Mason, W. H., Watson, L. T., and Haftka, R. T., "Variable-complexity response surface aerodynamic design of an HSCT wing," AIAA Paper 95-1886, 1995.

[75] Glaz, B., Goel, T., Liu, L., Friedmann, P. P., and Haftka, R. T., "Application of a weighted average surrogate approach to helicopter rotor blade vibration reduction," *48th AIAA Stuctures, Structural Dynamics and Materials Conference*, AIAA Paper 2007-1898, Honolulu, HI, April 2007.

[76] J. Forrester, A. I., Keane, A. J., and Bressloff, N. W., "Design and analysis of Noisy computer experiments," *AIAA journal*, Vol. 44, No. 10, 2006.

[77] Giunta, A. A., Dudley, J. M., Narducci, R., Grossman, B., Haftka, R. T., Mason, W. H., and Watson, L. T., *Noisy aerodynamic response and smooth approximations in HSCT design*, AIAA Paper 94-4376, 1994.

[78] Madsen, J. I., Shyy, W., and Haftka, R. T., "Response surface techniques for diffuser shape optimization," *AIAA journal*, Vol. 38, No. 9, 2000.

[79] Booker, A., "Well-conditioned Kriging models for optimization of computer simulations," Tech. Rep. M&CT-TECH-00-002, The Boeing Company, 2000.

[80] Rajnarayan, D. G., *Trading risk and performance for engineering design optimization using multifidelity analyses*, Ph.D. thesis, Stanford University, 2009.

[81] Koch, P. N., Simpson, T. W., Allen, J. K., and Mistree, F., "Statistical approximations for multidisciplinary design optimization: the problem of size," *Journal of Aircraft*, Vol. 36, No. 1, 1999.

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

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

[84] 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.

[85] Moré, J. J. and Wild, S. M., "Estimating computational noise," *SIAM Journal on Scientific Computing*, Vol. 33, No. 3, 2011.

[86] Lukaczyk, T., "VyPy - An Optimization Toolbox," `github.com/aerialhedgehog/VyPy`.

[87] Neumann, M., Huang, S., Marthaler, D., and Kersting, K., "PyGPs API," `http://www-ai.cs.uni-dortmund.de/weblab/static/api_docs/pyGPs/`, 2014.

[88] Sobol, I., "Multidimensional quadrature formulas and Haar functions," Moscow, 1969.

[89] Myer, R. and Montgomery, D. C., "Response surface methodology: process and product optimization using designed experiment," *John Wiley and Sons, New York*, 2002.

[90] Morris, M. D., "Factorial sampling plans for preliminary computational experiments," *Technometrics*, Vol. 33, No. 2, 1991.

[91] Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., and Tarantola, S., *Global sensitivity analysis: the primer*, John Wiley & Sons, 2008.

[92] Jolliffe, I., *Principal component analysis*, Wiley Online Library, 2002.

[93] Bui-Thanh, T., Willcox, K., and Ghattas, O., "Model reduction for large-scale systems with high-dimensional parametric input space," *SIAM Journal on Scientific Computing*, Vol. 30, No. 6, 2008.

[94] LeGresley, P. A., *Application of proper orthogonal decomposition (POD) to design decomposition methods*, Ph.D. thesis, Stanford University, 2005.

[95] Rowley, C., "Model reduction for fluids, using balanced proper orthogonal decomposition," *International Journal of Bifurcation and Chaos*, Vol. 15, No. 03, 2005.

[96] Carlberg, K., Bou-Mosleh, C., and Farhat, C., "Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations," *International Journal for Numerical Methods in Engineering*, Vol. 86, No. 2, 2011.

[97] van der Maaten, L. J., Postma, E. O., and van den Herik, H. J., "Dimensionality reduction: A comparative review," *Journal of Machine Learning Research*, Vol. 10, No. 1-41, 2009.

[98] Russi, T. M., *Uncertainty Quantification with Experimental Data and Complex System Models*, Ph.D. thesis, UC Berkeley, 2010.

[99] Constantine, P. G., *Active Subspaces: Emerging Ideas for Dimension Reduction in Parameter Studies*, SIAM, 2015.

[100] Lukaczyk, T., Palacios, F., Alonso, J., and Constantine, P., "Active subspaces for shape optimization," January 2014.

[101] Berguin, S. H., Rancourt, D., and Mavris, D. N., "A Method for High-Dimensional Design Space Exploration of Expensive Functions with Access to Gradient Information," *15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA Paper 2014-2174, Atlanta, GA, June 2014.

[102] Berguin, S. H., , and Mavris, D. N., "Dimensionality Reduction Using Principal Component Analysis in Aerodynamic Design," *10th AIAA Science and Technology Conference*, AIAA Paper 2014-0112, National Harbor, Maryland, January 2014.

[103] Constantine, P., “PAUL - Python Active-subspaces Utility Library,” `github.com/paulcon/active_subspaces`.

[104] Gill, P. E., Murray, W., and Wright, M. H., *Practical optimization*, Vol. 5, Academic press London, 1981.

[105] Wright, S. J. and Nocedal, J., *Numerical optimization*, Vol. 2, Springer New York, 1999.

[106] Pedersen, P., *The integrated approach of FEM-SLP for solving problems of optimal design*, Department of Solid Mechanics, Technical University of Denmark, 1980.

[107] Fleury, C. and Braibant, V., “Structural optimization: a new dual method using mixed variables,” *International journal for numerical methods in engineering*, Vol. 23, No. 3, 1986.

[108] Fleury, C., “CONLIN: an efficient dual optimizer based on convex approximation concepts,” *Structural Optimization*, Vol. 1, No. 2, 1989.

[109] Giunta, A. A. and Eldred, M. S., “Implementation of a trust region model management strategy in the DAKOTA optimization toolkit,” *Proceedings of the 8th AIAA/USAF/NASA/ISSMO symposium on multidisciplinary analysis and optimization, Long Beach, CA*, 2000.

[110] Rodrigues, J., Renaud, J. E., and Watsen, L., “Convergence of trust region augmented Lagrangian methods using variable fidelity approximation data,” *Structural optimization*, Vol. 15, No. 3-4, 1998.

[111] Nelder, J. A. and Mead, R., “A simplex method for function minimization,” *The computer journal*, Vol. 7, No. 4, 1965.

[112] Torn, A. and Zilinskas, A., *Global optimization*, Springer-Verlag New York, Inc., 1989.

[113] Goldberg, D. E., *Genetic algorithms*, Pearson Education India, 2006.

[114] Jones, D. R., Perttunen, C. D., and Stuckman, B. E., "Lipschitzian optimization without the Lipschitz constant," *Journal of Optimization Theory and Applications*, Vol. 79, No. 1, 1993.

[115] Kennedy, J., "Particle swarm optimization," *Encyclopedia of Machine Learning*, Springer, 2010.

[116] Žilinskas, A., "A review of statistical models for global optimization," *Journal of Global Optimization*, Vol. 2, No. 2, 1992.

[117] Lophaven, S., Nielsen, H., and Sondergaard, J., "DACEa Matlab Kriging toolbox; version 2; informatics and mathematical modelling," *Technical University of Denmark*, 2002.

[118] Sasena, M. J., Papalambros, P., and Goovaerts, P., "Exploration of metamodeling sampling criteria for constrained global optimization," *Engineering optimization*, Vol. 34, No. 3, 2002.

[119] Jones, D. R., Schonlau, M., and Welch, W. J., "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, Vol. 13, 1998.

[120] Locatelli, M., "Bayesian algorithms for one-dimensional global optimization," *Journal of Global Optimization*, Vol. 10, No. 1, 1997.

[121] Booker, A. J., "Examples of surrogate modeling of computer simulations," *ISSMO/NASA/AIAA First Internet Conference on Approximations and Fast Reanalysis in Engineering Optimization*, 1998.

[122] Schonlau, M., Welch, W. J., and Jones, D. R., "Global versus local search in constrained optimization of computer models," *IMS*, 1998.

[123] Shimoyama, K., Sato, K., Jeong, S., and Obayashi, S., "Comparison of the Criteria for Updating Kriging Response Surface Models in Multi-Objective Optimization," *IEEE World Congress on Computational Intelligence*, 2012.

[124] Palacios, F., Economon, T., et al., "SU2 - The Open-Source CFD Code," `github.com/su2code/SU2`.

[125] Paciorek, C., *Nonstationary Gaussian Processes for Regression and Spatial Modelling*, Ph.D. thesis, Carnegie Mellon University, May 2003.

[126] Stanford Aerospace Design Lab, "SUAVE: An Aerospace Vehicle Design Environment," `github.com/suavecode/SUAVE`.

[127] Anderson Jr, J. D., *Fundamentals of aerodynamics*, McGraw-Hill, 4th ed., 2007.