A METHOD FOR REDUCING DIMENSIONALITY IN LARGE DESIGN PROBLEMS WITH COMPUTATIONALLY EXPENSIVE ANALYSES

A Thesis
Presented to
The Academic Faculty
by
Steven H. Berguin

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology
May 2015

Copyright © 2015 by Steven H. Berguin
A METHOD FOR REDUCING DIMENSIONALITY IN LARGE DESIGN PROBLEMS WITH COMPUTATIONALLY EXPENSIVE ANALYSES

Approved by:

Professor Dimitri Mavris, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Professor Graeme Kennedy
School of Aerospace Engineering
Georgia Institute of Technology

Professor Stephen Ruffin
School of Aerospace Engineering
Georgia Institute of Technology

Professor Karim Lounici
School of Mathematics
Georgia Institute of Technology

Mr. Andrew Hahn
Aeronautics Systems Analysis Branch
NASA Langley Research Center

Date Approved: 5 January 2015
To my mother, my sister and my wonderful Elena.

This degree is as much yours as it is mine.
ACKNOWLEDGEMENTS

It has been a few years since the first day of graduate school and, as I mark the end of a journey, I would like to recognize those who made a difference along the way.

I would first like to thank my advisor, Prof. Dimitri Mavris, for his guidance, support and mentorship. I will never be grateful enough for the opportunities he made possible and the freedom of research he encouraged. My journey at ASDL taught me to think outside the box, harness thoughts and formulate ideas into original research. It has been a privilege to be part of ASDL.

I would also like to thank Mr. Andrew Hahn from NASA Langley Research Center. He was my technical advisor during my fellowship with NASA and I couldn’t have asked for a better one. His knowledge and passion for aircraft design were energizing and his guidance was essential in developing a thorough understanding of the challenges associated with designing airplanes with over-wing mounted engines.

I would further like to thank the entire team from the Aeronautics Systems Analysis Branch (ASAB) at NASA Langley for providing me with a friendly platform to share my early ideas. Their feedback was an invaluable milestone on the path to finding a topic and their passion for aeronautics is inspiring. In particular, I would like to recognize Dr. Natalia Alexandrov for being so generous with her time and providing me with feedback on my early thesis ideas.

Finally, last but not least, I would like to thank all the members of my Ph.D. committee for the taking the time to review this research and endorse it. Your feedback has gone a long way in improving the final product and, in particular, I would like to thank Dr. Lounici for exposing me to enabling mathematical concepts, which turned out to be the corner stones of this thesis.
# TABLE OF CONTENTS

DEDICATION ................................................................................................................ iii

ACKNOWLEDGEMENTS ................................................................................................. iv

LIST OF TABLES ........................................................................................................ viii

LIST OF FIGURES .......................................................................................................... ix

NOMENCLATURE ........................................................................................................ xii

SUMMARY ................................................................................................................... xv

I INTRODUCTION ......................................................................................................... 1

1.1 Context .................................................................................................................... 2

1.2 Scope ....................................................................................................................... 4

1.3 Problem Statement ................................................................................................. 5

II FORMULATION ......................................................................................................... 8

2.1 Current Practices in Propulsion-Airframe Integration ........................................... 8

2.1.1 Conventional Concepts ..................................................................................... 8

2.1.2 Unconventional Concepts ................................................................................. 11

2.1.3 Observations & Research Questions .................................................................... 13

2.2 The Consequence of Defaulting Shape Variables .................................................. 14

2.2.1 Historical OWN Displacement Study ............................................................. 14

2.2.2 Experiment 1: Numerical OWN Displacement Study ..................................... 16

2.3 Synthesis: The Need for a New Approach .............................................................. 37

III METHODOLOGY ...................................................................................................... 40

3.1 Development Process and Main Hypothesis ........................................................ 40

3.1.1 Taxonomy of Dimensionality Reduction Methods ........................................ 41

3.1.2 Principal Component Analysis ......................................................................... 44

3.1.3 Method Functional Decomposition ................................................................... 48

3.2 Method: Latent Space Gradient Transformation .................................................. 50
3.2.1 Optimization Problem Setup ........................................ 50
3.2.2 Step 1: Scale and Center Design Space .................. 51
3.2.3 Step 2: Collect Gradient Samples .......................... 51
3.2.4 Step 3: Apply Principal Component Analysis ........... 52
3.2.5 Step 4: Reduce Dimensionality ........................... 54
3.2.6 Step 5: Establish Forward and Backward Mappings .... 55
3.2.7 Step 6: Check Model Accuracy ............................ 57
3.2.8 Step 7: Design Space Exploration ......................... 58
3.2.9 Step 8: Gradient-Based Optimization ...................... 59
3.3 Concepts & Limitations ........................................... 59
  3.3.1 Effect of Correlation ......................................... 60
  3.3.2 Effect of Non-Zero Mean Gradient ........................ 64
  3.3.3 Effect of Centering the Design Space ...................... 65
  3.3.4 Synthesis .................................................. 67

IV STEP-BY-STEP CANONICAL EXAMPLE ............................ 68
  4.1 Optimization Problem Setup .................................. 68
  4.2 Step 1: Scale and Center Design Space .................. 70
  4.3 Step 2: Sample Gradient ...................................... 71
  4.4 Step 3: Apply Principal Component Analysis ........... 72
  4.5 Step 4: Reduce Dimensionality ........................... 74
  4.6 Step 5–6: Establish Mapping & Check Model Accuracy .... 75
  4.7 Step 7–8: Design Space Exploration & Optimization .... 76

V RESULTS ............................................................... 81
  5.1 Experiment 2: Staggered Airfoil Problem .................. 81
    5.1.1 Problem Setup ........................................... 82
    5.1.2 Dimensionality Reduction: LSGT vs. Screening Methods . 85
    5.1.3 Optimization: Comparison With and Without LSGT .... 95
  5.2 Experiment 3: OWN Problem .................................. 97
LIST OF TABLES

1  Definitions and ranges of design variables (same as Forrester et al. [14])  69
2  Comparison of predicted and actual minimum  . . . . . . . . . . . . .  79
3  Staggered airfoil design space  . . . . . . . . . . . . . . . . . . . . .  83
4  DR with PCA  . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  102
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Example of unconventional configurations (courtesy NASA).</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Embraer Engine-Airframe Methodology [65] (reproduced)</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>Wave drag vs. over-wing nacelle position by Fujino et al. [17] (reproduced)</td>
<td>11</td>
</tr>
<tr>
<td>4</td>
<td>Example design space exploration by Hooker et al. [27]</td>
<td>13</td>
</tr>
<tr>
<td>5</td>
<td>Effect of contouring nacelle shape by Henderson [24] (reproduced)</td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td>Illustration of numerical experiment</td>
<td>16</td>
</tr>
<tr>
<td>7</td>
<td>Baseline geometry</td>
<td>17</td>
</tr>
<tr>
<td>8</td>
<td>OWN design variables</td>
<td>18</td>
</tr>
<tr>
<td>9</td>
<td>Modeling &amp; Simulation environment</td>
<td>20</td>
</tr>
<tr>
<td>10</td>
<td>Euler grid resolution (fuselage present but not shown)</td>
<td>22</td>
</tr>
<tr>
<td>11</td>
<td>Drag as a function of nacelle location at $C_L = 0.525$ and $M = 0.785$</td>
<td>24</td>
</tr>
<tr>
<td>12</td>
<td>Optimization convergence history</td>
<td>25</td>
</tr>
<tr>
<td>13</td>
<td>Euler solution at $x/c = -0.85$, $M = 0.785$, $C_L = 0.525$</td>
<td>26</td>
</tr>
<tr>
<td>14</td>
<td>Euler solution at $x/c = -0.54$, $M = 0.785$, $C_L = 0.525$</td>
<td>27</td>
</tr>
<tr>
<td>15</td>
<td>Euler solution at $x/c = -0.23$, $M = 0.785$, $C_L = 0.525$</td>
<td>28</td>
</tr>
<tr>
<td>16</td>
<td>Euler solution at $x/c = 0.08$, $M = 0.785$, $C_L = 0.525$</td>
<td>29</td>
</tr>
<tr>
<td>17</td>
<td>Euler solution at $x/c = 0.38$, $M = 0.785$, $C_L = 0.525$</td>
<td>30</td>
</tr>
<tr>
<td>18</td>
<td>Euler solution at $x/c = 0.69$, $M = 0.785$, $C_L = 0.525$</td>
<td>31</td>
</tr>
<tr>
<td>19</td>
<td>Euler solution at $x/c = 0.85$, $M = 0.785$, $C_L = 0.525$</td>
<td>32</td>
</tr>
<tr>
<td>20</td>
<td>Euler solution at $x/c = 1.00$, $M = 0.785$, $C_L = 0.525$</td>
<td>33</td>
</tr>
<tr>
<td>21</td>
<td>Comparison of Euler vs. RANS OWN displacement study</td>
<td>35</td>
</tr>
<tr>
<td>22</td>
<td>Notional illustration of beneficial thrust created by suction peak [53]</td>
<td>37</td>
</tr>
</tbody>
</table>
23 Taxonomy of unsupervised DR by Lee and Verleysen [38] (modified).
Acronyms: PCA, principal component analysis [30]; LDR, linear dimensionality reduction; NLDR, nonlinear dimensionality reduction;
Methods: NLM, nonlinear mapping [74]; CCA, curvilinear component analysis [13]; Isomap [80]; GNLM, geodesic NLM [45]; CDA, curvilinear distance analysis [44]; KPCA, Kernel PCA [76]; SDE=MVU, Semidefinite embedding=Maximum variance unfolding [88]; SOM, self-organizing maps [34]; GTM, generative topographic mapping [11]; LLE, locally linear embedding [72]; LE, Laplacian eigenmaps [5]; Isotop [43]; AA NN, auto-associative neural network [25]; MES, main effect screening [57]; EES, elementary effect screening [55].

24 Unsupervised learning example ........................................ 42
25 Notional illustration of the concept behind PCA ..................... 47
26 Effect of $\xi_r = 0$ (green lines) vs. Eqs. 42–43 (red lines) .......... 58
27 Projection along first principal component, $a_1, \beta_{12} = 1.5$ ........ 61
28 Zero correlation .............................................................. 63
29 Effect of nonlinear correlation on PCA .................................. 64
30 Projection along 1st principal component, $\beta_1 = 0.9, \beta_2 = -\beta_1, \beta_{12} = 1.5$ 65
31 Effect of choosing $x_0$ ...................................................... 66
32 $\frac{\partial f}{\partial \Lambda}$ vs. $\Lambda$ holding all other variables fixed at baseline .... 69
33 Convergence history of $\|S_N - \Sigma\|_{\infty}$ (see Eq. 19) ............ 72
34 Scatter plot matrix for $\nabla_z f$ .............................................. 73
35 Visualization of the matrix $A$ (symbols: $Sw=S_w, Wfw=W_{Fw}, A=A,
L=\Lambda, q=q, h=\lambda, tc=t/c, Nz=N_z, Wdg=W_{dg}, Wp=W_p$) ....... 74
36 Pareto plots for mean and standard deviation of $\partial f / \partial \xi_i, \forall i = 1, \ldots, 10$ 75
37 Model validation error for $f(\xi_q)$ ...................................... 76
38 Scatter plot matrix of sampling plan .................................... 77
39 Model validation error for $\hat{f}(\xi_q)$ .................................... 77
40 Sensitivity plot of wing weight, holding all other variables fixed to baseline ................................................................. 78
41 OWN vs. staggered airfoil comparison ................................ 82
42 Overview of modeling and simulation environment .................. 84
43 Example of CST shapes over selected design space ................. 85
Staggered airfoil grid sensitivity study at $M = 0.8$ and $C_L = 0.525$...

Convergence history of $\|S_N - \Sigma\|_\infty$ ........................................... 87

Dimensionality reduction with PCA (truncated beyond 19 PC) ...... 88

Coefficient values for first and second principal components .... 89

Model quality of $f(\xi_q)$ as a function of $q$ (100 validation points) ... 90

Main effects plot of $f(z) = 200C_d(z) + 650(C_l(z) - 0.525)^2$ .... 91

EES results ................................................................. 94

Dimensionality reduction with PCA (truncated beyond 19 variables) . 96

Comparison of optimization with and without LSGT ................. 98

Comparison of optimization with and without LSGT .................. 99

Convergence history of covariance matrix estimator ............... 100

Dimensionality reduction with PCA .................................. 101

Model quality of $f(\xi_q)$ as a function of $q$ (100 validation points) ... 102

Dimensionality reduction without PCA .............................. 103

Boeing YC-14 .................................................................. 117

Fokker VFW-614 .................................................................. 118

NAL ASKA .......................................................................... 120

NASA QSRA .......................................................................... 121

Honda Business Jet ............................................................ 122

Russian OWN concepts ...................................................... 123

Screening using elementary effects (reprinted [14]) ............... 130

Screening using scaled partial derivatives (symbols: $Sw = S_{w}$, $Wfw = W_{fw}$, $A = A$, $L = \Lambda$, $q = q$, $h = \lambda$, $tc = t/c$, $Nz = N_z$, $Wdg = W_{dg}$, $Wp = W_{p}$) 131

Screening using unscaled partial derivatives (symbols: $Sw = S_{w}$, $Wfw = W_{fw}$, $A = A$, $L = \Lambda$, $q = q$, $h = \lambda$, $tc = t/c$, $Nz = N_z$, $Wdg = W_{dg}$, $Wp = W_{p}$) 132
Nomenclature

\( A \) Matrix containing ordered eigenvectors of \( S_N \)

\( A_q \) Partition of \( A \) containing \( q \leq p \) columns

\( A_r \) Partition of \( A \) containing \( r = p - q \) columns

\( A_r \) CST coefficient for airfoil shape

\( A_s \) Partition of \( A \) containing \( s = q - t \) columns

\( A_t \) Partition of \( A \) containing \( t \leq q \) columns

\( a_j \) \( j^{th} \) column of \( A \)

\( C_D \) Drag coefficient

\( C_d \) Section drag coefficient

\( C_L \) Lift coefficient

\( C_l \) Section lift coefficient

\( f \) Objective function

\( J(s) \) Subset of \( s \) indices

\( N \) Number of samples

\( p \) Total number of design variables

\( q \) Reduced number of design variables

\( r \) Number of principal components NOT retained in basis

\( s \) Number of latent variables responsible for linear effects

\( t \) Number of latent variables responsible for nonlinear effects

\( S_N \) Sample covariance matrix of \( \nabla_z f \)

\( W \) Scaling matrix

\( x \) Vector of design variables

\( x_0 \) Reference point

\( x_L \) Lower bound for \( x \)

\( x_U \) Upper bound for \( x \)
\( y \) Vector of partial derivatives
\( y_i \) \( i \)th sample of \( y \)
\( \bar{y} \) Mean of multivariate \( y \) distribution
\( z \) Vector of normalized design variables
\( z_i \) \( i \)th sample of \( z \)
\( z_L \) Lower bound for \( z \)
\( z_U \) Upper bound for \( z \)

**Greek**
\( \beta_i \) \( i \)th regression coefficient
\( \epsilon \) Prediction error
\( \epsilon_i \) \( i \)th sample of prediction error
\( \lambda_j \) \( j \)th eigenvalue of \( S_N \)
\( \mu \) Expected value (i.e. mean)
\( \xi \) Vector of latent variables
\( \sigma \) Standard deviation
\( \Sigma \) Covariance matrix
\( \nu \) Vector of partial derivatives in latent space
\( \phi_i \) \( i \)th basis function
\( \Psi \) Ratio of variation explained relative to total variation
\( \Omega \) Domain of feasible latent space

**Math Symbols**
\( D \) Domain of feasible design space
\( \nabla_x f \) Gradient of \( f \) with respect to \( x \)
\( \nabla_z f \) Gradient of \( f \) with respect to \( z \)
\( \nabla_\xi f \) Gradient of \( f \) with respect to \( \xi \)

**Acronyms**
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFD</td>
<td>Computational Fluid Dynamics</td>
</tr>
<tr>
<td>DR</td>
<td>Dimensionality Reduction</td>
</tr>
<tr>
<td>ERA</td>
<td>Environmentally Responsible Aviation</td>
</tr>
<tr>
<td>LE</td>
<td>Leading Edge</td>
</tr>
<tr>
<td>NASA</td>
<td>National Aeronautics and Space Administration</td>
</tr>
<tr>
<td>OML</td>
<td>Outer Mold Line</td>
</tr>
<tr>
<td>OWN</td>
<td>Over-Wing Nacelle</td>
</tr>
<tr>
<td>RSE</td>
<td>Response Surface Equation</td>
</tr>
<tr>
<td>TE</td>
<td>Trailing Edge</td>
</tr>
<tr>
<td>UHB</td>
<td>Ultra-High Bypass</td>
</tr>
<tr>
<td>UWN</td>
<td>Under-Wing Nacelle</td>
</tr>
</tbody>
</table>
SUMMARY

Motivated by environmental concerns, NASA is developing innovative ways to reduce fuel burn, noise and emissions for future civil transports, one of which involves integrating Ultra-High Bypass (UHB) ratio engines over the wing. Such concepts offer the promise of improved fuel efficiency by increasing the bypass ratio and, most of all, perceived noise reduction by taking advantage of the wing to shield ground observers from the noise of the engines. Historically, Over-Wing Nacelles (OWN) concepts have been notorious for generating large amounts of drag at high speeds compared to conventional Under-Wing Nacelle (UWN) concepts, but recent progress suggests that this reputation is not well deserved. However, achieving aerodynamic efficiency for such unconventional airplanes is a challenging endeavor, plagued by the “curse of dimensionality” and exacerbated by the need for expensive CFD function evaluations. For example, the design space required for aerodynamic shape optimization typically reaches up to hundreds of design variables. The objective of this thesis is to overcome these challenges in order to enable the use of gradient-free methods for large, aerodynamic design problem, using OWN integration as a representative application for this class of problems.

Strides in modern computational fluid dynamics and leaps in high-power computing have led to unprecedented capabilities for handling large aerodynamic problem. In particular, the emergence of adjoint design methods has been a break-through in the field of aerodynamic shape optimization. It enables expensive, high-dimensional optimization problems to be tackled efficiently using gradient-based methods in CFD; a task that was previously inconceivable. However, adjoint design methods are intended for gradient-based optimization; the “curse of dimensionality” is still very
much alive when it comes to design space exploration, where gradient-free methods cannot be avoided. Design space exploration is a complementary step that is useful to characterize trends, understand what drives the response and, most of all, identify regions where different local minima are likely to exist. Yet, without sufficient design freedom, there is a risk that otherwise good concepts might be discarded on the basis that a satisfactory solution could not be found because the design space was not explored to the full extent necessary.

This research describes a novel approach that seeks to solve this problem by reducing dimensionality to a point where gradient-free methods can be used to solve large, computationally expensive design problems. It uses an innovative application of Principal Component Analysis (PCA) to reduce dimensionality, where the latter is applied to the gradient distribution of the objective function; something that had not been done before. This yields a linear transformation that maps a high-dimensional problem onto an equivalent low-dimensional subspace. None of the original variables are discarded; they are simply linearly combined into a new set of variables that are fewer in number. The method is tested on a range of analytical functions, a two-dimensional staggered airfoil test problem and a three-dimensional OWN integration problem. In all cases, the method performed as expected and was found to be cost effective, requiring only a relatively small number of samples to achieve large dimensionality reduction.
CHAPTER I

INTRODUCTION

The origins of this thesis take root in a GSRP fellowship sponsored by NASA Langley Research Center, where the objective was to assess the potential of Over-Wing-Nacelle (OWN) concepts for future generation civil transports, with focus on engine integration. Historically, OWN concepts are notorious for generating large amounts of drag at high speeds compared to conventional Under-Wing Nacelle (UWN) concepts. However, the success of the Honda Jet [17] in the business arena, a small business jet with over-wing engines, and a recent study by government and industry in the military arena [27] prove that this reputation is not well deserved. In fact, according to those authors, OWN concepts offer an aerodynamic advantage over their UWN counterparts. However, in order to achieve it, significant computational resources have to be expended. For example, Hooker et al. [27] report using over 2,500 Navier-Stokes solutions to explore over 790 engine positions [27] from which 25 configurations were down-selected for further optimization, resulting in even more CFD runs. Moreover, exploring such design problems is particularly challenging because the number of variables required to deform the Outer Mold Lines (OML) of the wing and the nacelles is usually on the order of hundreds. As a result, they are usually defaulted to some optimal baseline value during design space exploration, in favor of concentrating on other variables such as nacelle location, Mach number or engine type. The primary objective of this thesis is therefore to develop a method that removes the need to default important degrees of freedom during early exploratory design phases, using OWN engine integration a representative application. In the process, the secondary objective will be to assess the potential of OWN concepts as future generation civil
transports. This chapter introduces the reader to the context, scope and problem statement of this research.

1.1 Context

In response to rising environmental awareness, the National Aeronautics and Space Administration (NASA) endeavors to catalyze the development of the next generation of eco-friendly civil transports. Under the scope of NASA’s Environmentally Responsible Aviation (ERA) project, a large portfolio of innovative technologies is being considered. The goal is to reduce fuel burn by 50 percent relative to best in class, community noise by 42dB below stage 4, and NOx emissions by 75 percent at cruise by 2020 [77]. However, after more than half a century of evolution, some NASA experts believe that conventional tube and wing configurations could be reaching the limit of their full potential. In order to meet the ERA’s ambitious objectives, incremental improvements may no longer be sufficient; a leap is required. As a consequence, NASA is considering several unconventional airplane concepts among its portfolio of solutions. For example, Fig. 1 shows a joined-wing, a Hybrid-Wing Body (HWB), and an Over-Wing Nacelle (OWN) configuration currently under review. None of these concepts are new; however, they have never been applied to civil aviation before and are therefore unconventional. In the military arena for example, Northop-Grumman had already developed the B-2 bomber, which is also a flying wing concept like the HWB. Similarly, Boeing had successfully developed the YC-14, a Short Take-Off and Landing (STOL) tactical military airlift that used over-wing nacelles to generate powered lift through upper surface blowing [28]. Finally, the concept of a joined wing has been known since the early days of Prandtl [67].
Figure 1: Example of unconventional configurations (courtesy NASA).
1.2 Scope

Each one of these concepts promises to reduce fuel burn, noise and emissions through the novelty of its configuration, rather than solely relying on improved subsystem technology. For instance, a joined wing reduces induced drag by improving span efficiency at fixed span [36], while avoiding ground clearance issues when installing Ultra-High Bypass (UHB) turbofan engines as shown. An HWB reduces induced drag by taking advantage of greater structural depth at the wing root to increase span without weight penalty [49] and, furthermore, it also offers a convenient installation for UHB engines, which could be installed on top of the passenger cabin as shown. This engine location might also reduce perceived engine noise by using the airframe to shield a ground observer from the noise of the engine. In addition, the HWB also has a parasite drag advantage. The reason is that the wetted aspect ratio is higher \((b/S_{\text{wet}})\), meaning that for the same span, the wetted area is lower, which is a first order indicator of parasite drag. The key problem is that the non-circular pressurized cross section structural disadvantage may more than offset the wing structure advantage. Finally, the primary advantage of OWN is that it offers similar benefits without departing drastically from conventional concepts. In particular, the wing can be used to shield ground observers from the noise of the engine (but cabin noise would increase), it provides a convenient accommodation for UHB turbofans and, with significant effort, there is the possibility of improved aerodynamic efficiency [27, 17].

In addition to reducing airframe drag, fuel burn can also be improved by producing thrust more efficiently. For instance, notice that all concepts in Fig. 1 feature large turbofan engines with Ultra-High Bypass ratios (UHB). This class of engines improves fuel efficiency significantly; however, they are also harder to integrate onto the airframe [26]. When the engine is brought into close proximity of the airframe, the flow field of the former affects the flow field of the latter and vice versa. This is known as aerodynamic interference. Interference is exacerbated by the size of the
engine and can be either favorable or unfavorable. If it is unfavorable, then it is termed installation drag. Propulsion-Airframe Integration (PAI) is the process that minimizes installation penalties throughout the flight envelope. This is a difficult problem because, in order to ensure a feasible design, PAI must take into account not only installed drag, but also tradeoffs with TSFC, weight balance, structural design, noise and thrust requirements, in addition to performing satisfactorily at on- and off-design conditions [7]. PAI is therefore a Multi-Disciplinary Analysis and Optimization (MDAO) problem involving aeroacoustics, aerodynamics, propulsion and structures. However, unless a feasible disciplinary solution even exists for aerodynamics, then tackling the MDAO problem has no chance of success.

This thesis will therefore focus on aerodynamic shape optimization, which is challenging for two important reasons. First, aerodynamic function evaluations are expensive because, for unconventional concepts, the designer must rely on Computational Fluid Dynamics (CFD), which can take on the order of hours to run a single case. Secondly, the design space is high dimensional, with a number of design variables on the order of 100. The curse of dimensionality therefore prevents the design space from being fully explored and, as a consequence, there is a risk that otherwise good airplane concepts might be discarded on the basis that a satisfactory engine installation could not be found.

1.3 Problem Statement

Strides in modern computational fluid dynamics and leaps in high-power computing have led to unprecedented capabilities for handling large aerodynamic problem. In particular, the emergence of adjoint design methods [20, 29] has been a break-through in the field of aerodynamic shape optimization. It enables expensive optimization problems with hundreds of dimensions to be tackled efficiently using gradient-based methods in CFD; a task that was previously inconceivable. For example, Smith et
al. [79] successfully solved a civil transport nacelle integration problem with 146 design variables, Leung et al. [48] optimized the Onera M6 wing with 225 design variables, and Lyu and Martins [52] solved a Hybrid Wing Body (HWB) optimization problem with 273 design variables. However, adjoint design methods are intended for gradient-based optimization; the curse of dimensionality [6] is still very much alive when it comes to design space exploration, where gradient-free methods cannot be avoided.

Design space exploration is a complementary step that can be used to characterize trends within the context of multi-disciplinary trade studies, understand what drives the response or identify regions of interest. This can be accomplished in multiple ways but, in general, a large number of samples is usually required and the computational burden quickly becomes unaffordable beyond ten design variables, especially when function evaluations are expensive; see for example the work of Rodriguez [70]. It would therefore be useful if there were a way to enable the use of gradient-free methods for large CFD problems, which yields the main research question behind this thesis:

**Question 1.** Given that a large number of design variables is usually required for aerodynamic design, how is it possible to overcome the curse of dimensionality in order to facilitate the use of gradient-free methods for large CFD problems without defaulting important degrees of freedom?

To the author’s knowledge, no solution currently exists to solve this problem. For instance, in an extensive survey on modeling and optimization strategies to solve high-dimensional design problems with computationally expensive black-box functions, Shan and Wang [78] concluded that “model approximation techniques have been successfully applied to low dimensional, expensive black-box problems, [but that] further study is worthy and needed for high dimensional problems.” In particular, they argued for the need to develop a unique mapping that allows a high-dimensional
optimization problem to be re-formulated into an equivalent lower dimensional space, while preserving the optimum of the original function. Furthermore, they called for a deeper understanding of the high-dimensional space by enabling design space exploration and new ways to build high-dimensional surrogate models. The present work establishes such a mapping using a novel application of Principal Component Analysis (PCA), which facilitates exploration by reducing the design space to its “intrinsic dimensionality,” where the latter term refers to the minimum number of parameters needed to account for the observed properties of the response. The document is organized as follows: Chapter II formulates preliminary research questions and hypotheses based on observations from literature, Chapter III describes the mathematical approach created to solve the problem and, finally, Chapters IV–VI apply the method to a canonical problem, an applied 2D aerodynamic problem and a 3D engine integration, respectively, in order to test its performance. Finally, based on these results, Chapter VII concludes whether or not the approach was successful.
CHAPTER II

FORMULATION

The purpose of this chapter is to formulate the preliminary research questions and hypotheses that motivate this thesis. It starts with an assessment of current practices in Propulsion-Airframe Integration (PAI) and, progressively, works its way to the primary motivation for developing the method behind this thesis.

2.1 Current Practices in Propulsion-Airframe Integration

This section explains how industry currently conducts engine integration, using representative examples found in literature.

2.1.1 Conventional Concepts

The framework used to develop the Embraer 170 is reproduced in Fig. 2, as a representative example of PAI for conventional airplane configurations. In a first step, multi-disciplinary considerations are addressed during the conceptual design phase. Oliveira et al. [65] explain that nacelle position affects force and moment balance, which influences handling characteristics, loads, structures, weight and performance. For example, nacelle span position alone alters both wing bending moment, which affects internal structure, and engine inoperative yaw moment, which sizes the vertical tail. Furthermore, installation drag affects thrust, which affects the size of the engine and, therefore, the amount of fuel that must be carried. In addition to such physical considerations, non-physical constraints such as safety (e.g. foreign object damage avoidance, turbine disk burst or ground clearance), operations (e.g. door clearance for loading/unloading) and maintenance issues (e.g. engine maintenance access) must also be accounted for [65, 7]. However, notice that aerodynamic shape
Figure 2: Embraer Engine-Airframe Methodology [65] (reproduced)
optimization using higher order analysis is not part of the conceptual design process. Instead, interference effects are initially addressed using empirical data, which allows the preliminary engine position to be taken away from historical zones of high interference [65]. As long as the airplane fits inside the empirical envelope, this approach provides a “good” geometrical starting point for the more elaborate process to follow.

In a second step, once this geometrical starting point is obtained, 3D Euler and Navier-Stokes methods are used for further refinement [65]. The nacelle is perturbed vertically and horizontally in order to find a satisfactory engine location that minimizes interference drag. Local contours are held fixed during this process; they are not re-optimized. Once this is done, nacelle orientation is modified to ensure that the inlet is aligned with the local flow and that the jet plume does not interfere with the flaps or horizontal tail [65]. At this point of the process, a feedback loop ensures that the new nacelle position and orientation still meet basic multidisciplinary constraints. If so, the design proceeds to local lofting of the pylon and the wing to account for the presence of the nacelle, where aerodynamic shape optimization is used to reduce wave drag and boundary layer separation caused by interference effects. The same process is then repeated for each alternative under consideration.

While this approach has led to world-class airplanes that are strong competitors in the regional jet market, notice that the process starts with prior knowledge. Finding a “good” geometrical starting point requires the existence of a historical database. Unfortunately, such a database does not exist for unconventional airplane concepts. Furthermore, changes to the design variables, including engine location, are confined to a local region in the vicinity of the starting geometry. Design space exploration is only achieved by repeating the process for a handful of alternative engine installations (e.g. rear-mounted or unconventional), as shown in Fig. 2. However, no guidance is provided for how to select “good” geometrical starting points for unconventional alternatives. Similar observations also hold for other airframe manufacturers, such as
Boeing. For example, Berry [7] describes the process used to integrate the engines onboard the Boeing 777. As for Embraer, the starting position of the nacelle was selected based on empirical considerations inspired from the 757 and 767 programs, followed by local refinement.

2.1.2 Unconventional Concepts

Since historical databases usually don’t exist for unconventional configurations, it is necessary to first explore the design space in order to identify “good” geometries worthy of further study. The Honda Business Jet (HBJ) features over-wing nacelles and, therefore, it is a salient example of how industry approached engine integration for an unconventional concept. The design and development of this airplane is summarized in four publications by Fujino et al. [16, 17, 18, 19] and a description of the airplane can be found in Appendix B. Since those authors could not rely on historical data, they opted to use design sweeps in order to explore candidate nacelle positions [17]. However, the design space was limited to three variables taken to be axial, span-wise
and vertical nacelle position; the shape of the Outer Mold Line (OML) did not vary. In fact, the wing was designed beforehand, independently of the nacelle, in order to promote natural laminar flow [19]. By judiciously using CFD to guide wind tunnel tests, those authors discovered that favorable aerodynamic interference was possible if the nacelle inlet was located near the 80% chord, as shown in Fig. 3.

Similarly, in a recent OWN study for an air mobility mission conducted by Lockheed-Martin (LM) and the Air Force Research Labs (AFRL), Hooker et al. [27] explored over 790 engine positions [27] from which 25 configurations were down-selected for further aerodynamic shape optimization. However, as for the HBJ, the shape OML was held fixed during design space exploration. Only three continuous and two discrete variables were considered: horizontal nacelle position, vertical nacelle position, Mach number, engine architecture (e.g. mixed flow turbofan vs separate flow turbofan at different bypass ratios) and airframe type (e.g. high wing vs. low wing). In order to explore this design space, those authors took advantage of design of experiment methods, as illustrated in Fig. 4 for a single engine-airframe combination. However, they emphasize that the criteria for down-selecting “good” starting geometries was not based on optimality (i.e. the best nacelle positions found). Rather, it was based on the belief that a given configuration might improve sufficiently enough to meet requirements upon further aerodynamic shape optimization. The human criterion for making such decisions is thus “maximization of expected improvement.” Human experts are effective at this task because of their ability to find associations among scarce amounts of information and extrapolating it using prior knowledge (i.e. experience and education), in order to determine the likelihood of improvement. Overall, those authors report using over 2,500 Navier-Stokes solutions to carry out this effort, which resulted in a 5% improvement in efficiency over conventional installations under identical conditions. Once again, the success of this approach suggests that exploring the design space need not consider the OML, provided there is an expert in the loop.
An advantage of CFD is that the drag on individual aircraft components can easily be calculated, making it possible to study the interference drag in detail by breaking it into separate aircraft and nacelle components. For this effort, three primary interference drag components were used to evaluate the performance characteristics of the various engine positions studied. 

ΔC_D_Aircraft (or airframe interference drag) was calculated by considering only the forces acting on the aircraft portion of the complete model minus the clean wing. Similarly, the ΔC_D_Nacelle (or nacelle interference drag) was calculated by considering the forces acting on the nacelle portion of the complete model minus the isolated nacelle. The total interference drag (ΔC_D_All) is equal to ΔC_D_Aircraft plus ΔC_D_Nacelle. These drag components are illustrated in Figure 12.

Figure 4: Example design space exploration by Hooker et al. [27] to assess “expected improvement.”

2.1.3 Observations & Research Questions

At the outcome of this section, several observations can be made about current practices in PAI. First, aerodynamic shape optimization is not part of conceptual design, even though this is where design freedom is greatest. Second, in the absence of historical data, finding a “good” starting geometry worthy of further study requires design space exploration, which is usually achieved by defaulting the OML to some predetermined baseline shape in order to focus on nacelle displacement. Third, candidate engine placements worth studying further are selected on the basis of “expected improvement,” assessed using expert opinion. However, reliance on human expertise introduces subjectivity and uncertainty in the process. Furthermore, in the context of Multi-Disciplinary Optimization (MDO), it would be preferable to account for all design variables at once, as early on as possible, without defaulting important degrees of freedom or relying on historical data. New solutions might then be found, which may not have been obvious otherwise. On the basis of these observations, the
following research question must logically be raised:

**Question 2.** What is the consequence of defaulting OML shape variables during engine placement selection?

### 2.2 The Consequence of Defaulting Shape Variables

In attempt to answer Research Question 2, this section compares the effect of displacing the nacelle with and without re-shaping the OML using two examples. The first is taken from literature, based on OWN wind-tunnel tests conducted in the 1980’s, and the second is generated numerically using Euler analysis.

#### 2.2.1 Historical OWN Displacement Study

In 1983 Henderson et al. [24] conducted a series of experiments in the NASA Langley 16-foot transonic wind tunnel, in order to investigate propulsion installation characteristics for several commercial transport configurations near Mach 0.8, including an over-wing nacelle configuration. They found that contouring the nacelle reduced interference drag by a staggering 50 percent, as shown in Fig. 5. Said otherwise, the expected improvement due to re-shaping the OML for this vehicle was 50%, which is not a small number. This was not enough to outperform the under-wing baseline, but it demonstrates the importance of accounting for OML shape variables during engine placement selection. The results of their study therefore suggest that installation drag is a strong function of OML shape and that displacing the nacelle without reshaping the OML could have up to 50% uncertainty. In the context of MDO, such large uncertainty would propagate to other disciplines and possibly cause the vehicle to be over-designed by masking better regions of the design space. There would thus be a strong advantage for MDO if shape variables could be accounted for early on.
Figure 5: Effect of contouring nacelle shape by Henderson [24] (reproduced)
2.2.2 Experiment 1: Numerical OWN Displacement Study

The study in the previous subsection only pertains to a specific airplane configuration at a fixed engine location. For the sake of generality, it would thus be preferable to repeat it for different engine locations using a different configuration. Since such a study could not be found in literature, it was decided to conduct it numerically.

Description of Numerical Experiment

The objective of the experiment is to compare installation drag penalties before and after optimization at several over-wing nacelle locations, as it translates from leading edge to trailing edge as illustrated in Fig. 6. The baseline geometry is shown in Fig. 7 and was provided by NASA based on the work by Smith et al. [79]. It represents a notional, single-aisle civil transport designed for $C_L = 0.525$ at Mach 0.785. The nacelles are flow-through and sized to accommodate a UHB engine with a bypass ratio of 15. The pylons were removed in order to provide direct comparison with the nacelle displacement study to follow, which does not consider pylons. This is common practice when studying a large number of engine locations; see for example the work of Reubush [69], Fujino et al. Fujino-Kawamura or Hooker et al. [27]. Pylons are hard to design and, according to NASA experts, their effect usually ends being minimal.
Figure 7: Baseline geometry
Figure 8: OWN design variables
anyway once the design is complete.

The corresponding OWN geometry is shown in Fig. 8, where it has been parameterized using a total of 145 design variables; the wing planform, fuselage and nacelle dimensions do not vary. All design variables are self-explicit and include airfoil shape, wing twist at seven sections, nacelle incidence and toe in angles, horizontal nacelle position relative to the wing leading edge and angle of attack. Each airfoil is parameterized using the class shape function transformation (CST) method [37]:

\[
\left( \frac{y}{c} \right) = \pm \sqrt{\frac{x}{c}} \left( 1 - \frac{x}{c} \right) \sum_{r=0}^{n=9} A_r \frac{n!}{r! (n-r)!} \left( \frac{x}{c} \right)^r \left( 1 - \frac{x}{c} \right)^{n-r} + \frac{x}{c} \left( \frac{y}{c} \right)_{TE} \quad (1)
\]

The plus and minus signs account for the upper and lower surface, respectively, and the CST coefficients \( A_r \), \( \forall k = 1, \ldots, 9 \) represent degrees of freedom. The leading coefficient \( A_0 \) was held fixed to the baseline value in order to ensure a round leading edge. Samareh [73] provides an insightful review of parameterization methods for aerodynamic shape optimization. The CST method was selected over others because it was easy to implement and is well suited for wing design. The minimum and maximum bounds of the CST coefficients are taken at \( \pm 50\% \) of the baseline value.

Finally, the objective is to minimize drag subject to the airfoil thickness constraints and flight conditions shown in Fig. 8. The objective function is given by,

\[
f(x) = 200 \cdot C_D(x) + 650 \cdot (C_L(x) - 0.525)^2
\]

where the penalty term avoids the need to compute a separate adjoint solution for lift, as suggested by Nemec et al. [61].

**Modeling & Simulation Environment**

The computational framework used to carry out the analysis is shown in Fig. 9, where all computations were performed on a MacPro workstation running 2 \( \times 2.4\)GHz 6-Core Intel Xeon processors with 64 GB was available. Optimization was performed in MATLAB using Sequential Quadratic Programming (SQP) via the function `fmincon`,
Figure 9: Modeling & Simulation environment
which makes external calls to the CART3D design environment [61], an unstructured cartesian Euler solver augmented with an adjoint solver to compute the gradient. In turn, CART3D communicates with Vehicle Sketch Pad (VSP) [22, 21], an open-source parametric airplane geometry code used to generate the surface grid, through the intermediary of a geometry wrapper written in C++. Although collaborative efforts between government and academia are underway to make such communication a native feature of VSP, a CST [37] capability for airfoils had not yet been implemented at the time of this writing, which is why the C++ wrapper was necessary.

*Flow Solver*

CART3D uses cell-centered, finite-volume, upwind differencing, where the spatial discretization scheme was set to van Leer flux difference splitting with a van Albada flux limiter. This scheme is second-order accurate. In order to accelerate convergence, CART3D marches to steady state using an unstructured, nested multi-grid procedure based on a modified Runge-Kutta scheme. More detail about the solver, including an ONERA M6 validation study, can be found in publications by Aftosmis et al. [1, 3]. For all simulations, the CFL number was set to 1.2 and the number of multi-grid cycles to four. The mesh extents were taken at $\pm 30$ fuselage lengths in all directions and free stream boundary conditions were used throughout, except at the symmetry plane where a symmetric boundary condition was used instead.

Although CART3D also features automated mesh adaption [60, 63, 62], which adaptively refines the mesh where it matters, it was decided to use a fixed grid approach instead for computational efficiency. The selected grid resolution is shown in Fig. 10(a) and the associated mesh refinement study is shown in Fig. 10(c) using the baseline under-wing nacelle configuration. The adaptively refined grid is shown in Fig. 10(b) for comparison and the dashed lines in Fig. 10(c) are the associated force coefficients. All solutions were converged to at least 4 orders of magnitude (in the
Figure 10: Euler grid resolution (fuselage present but not shown)
L1 norm of density) using the van Leer flux function and force coefficients were no longer varying. As can be seen, the selected grid resolution is within 10 drag counts of the adaptively refined grid, which was deemed satisfactory.

In general, the primary advantage of Euler methods is that they can be used to enhance understanding of aerodynamic trends without the 20-50 times larger [2] computational expense of higher-order Navier-Stokes approaches. Moreover, despite the lack of viscous effects, CART3D has proven to be remarkably accurate for predicting flight loads on broad classes of vehicles with diverse flight conditions [12, 56]. However, despite this general success, Aftosmis et al. [2] explain that “the critical importance of viscous effects in particular aerodynamic situations means that practitioners of these approaches must be constantly wary of being misled by the simplified physical modeling used.” In particular, those authors explain that transonic simulations over high aspect-ratio configurations with supercritical airfoils is particularly difficult for inviscid approaches and go on to show several examples. In general, based on the examples shown by Aftosmis et al. [2], CART3D tends to over predict the strength of shock waves and causes them to terminate further aft relative to experimental or Navier-Stokes data. However, the overall aerodynamic trend is usually preserved, which makes them useful for comparative analysis and understanding what drives the design.

Finally, since the focus of this thesis is on the method, rather than the CFD predictions, Euler methods were deemed acceptable for purposes of testing the approach. They represent a sufficient level of fidelity, necessary to capture transonic trends correctly, and a realistic level complexity on which to test the main ideas behind this thesis. However, it is fully acknowledged that Euler methods are inaccurate compared to Navier-Stokes methods and it should be clear to the reader that they are only being used as testbed to facilitate research. Said otherwise, it is expected that the conclusions of this research would hold if a Navier-Stokes solver were used
Figure 11: Drag as a function of nacelle location at $C_L = 0.525$ and $M = 0.785$ instead, but that aerodynamic predictions would be somewhat shifted.

Results

The variation of total drag coefficient as a function of nacelle location is shown in Fig. 11. There are three curves; the first corresponds to a fixed OML, the second to a re-optimized OML and the horizontal line is the baseline reference drag coefficient. The convergence history of the optimizer is shown in Fig. 12, where it can be verified that all solutions are approaching a converged optimum. The corresponding surface Mach contours and $C_p$ distributions are shown in Figs. 14–20. It can be seen from Fig. 11 that re-optimizing the OML accounts for up to 58% reduction in drag. From an MDO perspective, this implies that displacing the nacelle without accounting for shape variables could yield up to 58% uncertainty in drag, which could propagate to other disciplines and cause the vehicle to be over-designed. This can be prevented
by accounting for the OML early on, in order to open up the design space. For example, before optimization in Fig. 11, there are no feasible engine location capable of outperforming the baseline; but after optimization the range of feasible engine locations opens up to $x/c \in [-1, -0.23] \cup [0.85, 1.00]$. This is particularly noteworthy because having a choice facilitates multi-disciplinary tradeoffs down the line. For example, installation drag and noise tend to be conflicting objectives with respect to engine location and, similarly, Fujino and Oyama [18] showed that such tradeoffs also exist for aero-elasticity. Having more than one feasible engine location therefore makes it easier to compromise across disciplines. Overall, these preliminary findings suggest that OWN concepts represent viable alternatives for the future and are worth considering under the ERA program.

**Model Fidelity**

Euler methods assume inviscid flow and, therefore, boundary layer displacement effects and shock-induced boundary layer separation cannot be captured. As a result, it
Figure 13: Euler solution at $x/c = -0.85$, $M = 0.785$, $C_L = 0.525$
Figure 14: Euler solution at $x/c = -0.54$, $M = 0.785$, $C_L = 0.525$
(a) Before optimization, $C_D = 0.0305$

(b) After optimization, $C_D = 0.0131$

**Figure 15:** Euler solution at $x/c = -0.23$, $M = 0.785$, $C_L = 0.525$
(a) Before optimization, $C_D = 0.0405$

(b) After optimization, $C_D = 0.0167$

Figure 16: Euler solution at $x/c = 0.08, M = 0.785, C_L = 0.525$
(a) Before optimization, $C_D = 0.0436$

(b) After optimization, $C_D = 0.0193$

Figure 17: Euler solution at $x/c = 0.38$, $M = 0.785$, $C_L = 0.525$
(a) Before optimization, $C_D = 0.0329$

(b) After optimization, $C_D = 0.0138$

Figure 18: Euler solution at $x/c = 0.69$, $M = 0.785$, $C_L = 0.525$
**Figure 19:** Euler solution at $x/c = 0.85$, $M = 0.785$, $C_L = 0.525$
Figure 20: Euler solution at $x/c = 1.00$, $M = 0.785$, $C_L = 0.525$
is often a concern that “lower fidelity Euler based CFD optimization methods are inadequate and would design the aircraft to eliminate non-existent or incorrectly placed flow features” [27]. Moreover, the absence of powered boundary conditions decreases fidelity even further. Therefore, strictly speaking, the results presented in the previous subsection are inaccurate and cannot be used to start cutting metal so to speak. For instance, it is quite likely that the strength and location of the shock waves would change somewhat if the same geometries were run using RANS analysis. However, at the conceptual design level, a small loss of accuracy is often acceptable in exchange for increased design freedom, provided trends and interactions are captured correctly. In fact, it is not uncommon to design $C_p$-distributions using Euler analysis, followed by RANS inverse design methods to correct the OML in order to recover the inviscid $C_p$-distributions; see for example the work by Smith et al. [79]. This subsection is thus concerned with the following research question:

**Question 3.** Does an Euler analysis with flow-through nacelles capture trends and interactions correctly?

This question will be answered by comparing the Euler results in Fig. 21(a) for the fixed OML case against those of a similar OWN study by Hooker et al. [27] in Fig. 21(b), which used RANS analysis and powered boundary conditions. Both figures show the variation of OWN drag as a function of engine placement, except that it is presented in different formats. In Fig. 21(a), this information is shown graphically, broken down into total and component drag. Each curve corresponds to drag measured on all components together (i.e. aircraft plus nacelles), just the aircraft and only the nacelles, respectively. Similarly, Fig. 21(b) shows the same type of information using contour plots of installation drag instead. The first column corresponds to the entire airplane, the second column to just the aircraft and the third
Figure 15: Engine BPR Effect on Interference Drag for the Low Wing Aircraft at Mach = 0.85 and $C_L = 0.48$

Figure 21: Comparison of Euler vs. RANS OWN displacement study

---

(a) Euler OWN study at $C_L = 0.52$ and $M = 0.785$ (fixed OML)

(b) RANS OWN study at $C_L = 0.48$ and $M = 0.85$ by Hooker et al. [27] (modified)
column to only the nacelles. In both cases, the same trend is observed: installation drag is lowest when the nacelle is either ahead of the wing LE or aft of the wing TE, there is favorable interference (i.e. negative $\Delta C_D$) on the aircraft when the nacelle is located ahead of the wing LE (but nacelle interference is unfavorable) and, finally, there is favorable interference on the nacelle when it is located in the vicinity of the wing trailing edge (but wing interference is unfavorable).

Careful analysis of Figs. 13–20 (before optimization) provides a physical explanation for these trends. In general, the presence of the nacelle over the wing increases wave drag, but at LE and TE engine locations this is accompanied by increased suction on the wing LE in Figs. 13–14 and on the nacelle lower lip in Figs. 18–20. Suction peaks are beneficial because they create a component of thrust if they act on portions of the geometry with forward facing normals, as illustrated notionally in Fig. 22. This is particularly noteworthy because it suggests OWN concepts have an alternative mechanism for reducing drag, not usually present in UWN configurations. However, in order to take advantage of it, aerodynamic shape optimization is required to reduce wave drag on the wing. In fact, doing so yields an OWN TE location that outperforms the UWN baseline by 10 drag counts in Fig. 20. The presence of the nacelle at this aft location slows down the flow on the wing, which makes it easier to reduce wave drag by reshaping the wing. There remains a strong shock on the nacelle, but the increased suction peak on the lower nacelle lip compensates for that effect. Not only is this answer similar to the one found by Hooker et al. [27], but it is supported by similar findings by Fujino et. al [17]. Overall, it appears that an Euler analysis captures OWN trends and interactions correctly, despite a loss of accuracy due to inviscid flow assumptions, which yields the following hypothesis:

**Hypothesis 1.** An Euler analysis captures general design trends and interactions correctly and, therefore, it is reasonable for comparative analysis.
2.3 Synthesis: The Need for a New Approach

In summary, current practices in PAI tend to freeze the OML when exploring the design space of possible nacelle locations and engine-airframe combinations. Shape variables are not included in the design space. As a consequence, there is large uncertainty in drag at any given nacelle location, which can propagate to other disciplines and cause the airplane to be over-designed by masking feasible regions of the design space. The easiest way to solve this problem would be to re-optimize the geometry at every candidate engine location. For example, doing so in the previous OWN displacement study, the trend in Fig. 11 changed significantly upon optimization and led to the discovery of several feasible engine installations, which were not obvious beforehand. However, depending on the problem and the number of candidate designs to be tested, the cost of repeated optimization can be excessive. For example, Hooker et al. [27] investigated 790 different candidate designs; an expensive proposition if optimization is to be performed each time. As a consequence, in practice, the variation in drag due to the OML is usually accounted for by relying on experts to subjectively assess expected improvement at a given nacelle location. Rather than re-optimizing the OML every time, candidate engine placements are down-selected based on a qualitative assessment of their potential for further improvement upon optimizing the OML; see in particular the work by Hooker et al [27] for an example.

Figure 22: Notional illustration of beneficial thrust created by suction peak [53]
Overall, there does not seem to be an approach capable of accounting for OML shape variables during early exploratory design phases, where gradient-free methods cannot be avoided. However, if such a capability were to exist, then design freedom would be greatly increased, possibly leading to the discovery of unintuitive, multidisciplinary solutions. While the usefulness of such a capability is ultimately problem-dependent, the ability to employ a genetic algorithm or create a surrogate model to solve, large aerodynamic problems is surely worth pursuing. For example, the ability to create surrogate models for large, CFD problems would enable the use of high-fidelity aerodynamics in MDAO for the first time. Based on these deliberations, it appears that it would be worth the effort to develop such a capability:

**Hypothesis 2.** Defaulting the OML introduces large uncertainty in the design process, due to the unknown effect of changing the OML at candidate engine locations. As a consequence, there is a risk that a satisfactory solution might not be found because the design space was too restrictive at the time it was being explored. A new approach is therefore needed to explore large CFD problem without defaulting important degrees of freedom.

However, as explained in the introduction, this is a challenging endeavor because useful parameterizations of aerodynamic geometries often result in high-dimensional design spaces, which leads back to the main research question:

**Question. 1** Given that a large number of design variables is usually required for aerodynamic design, how is it possible to overcome the curse of dimensionality in order to facilitate design space exploration for large CFD problems, without defaulting important degrees of freedom?
CHAPTER III

METHODOLOGY

The curse of dimensionality can be alleviated by increasing computational power, reducing the cost of individual function evaluations using model approximation techniques and, most of all, reducing dimensionality itself. Examples of model approximation techniques include Reduced Order Models (ROM) such as [46], which lower the cost of solving partial differential equations numerically, multi-fidelity methods such as [4] or [71], which avoid the need to rely exclusively on high order models during optimization, and statistical regression methods. In fact, Shan and Wang [78] provide an extensive review of all modeling and optimization strategies that have been proposed in the past to solve computationally expensive problems with a large number of variables. However, they conclude that “approximation techniques have [only] been successfully applied to low dimensional expensive black-box problems [but that] further research is worthy and needed for high dimensional problems.” This implies that Dimensionality Reduction (DR) is a critical prerequisite for model approximation and, therefore, this chapter will focus on it. The objective is to develop a method capable of reducing dimensionality to a point where model approximation techniques and other gradient-free methods can be considered for exploring large CFD problems.

3.1 Development Process and Main Hypothesis

The purpose of this section is to describe the development process and general reasoning that led to LSGT, the name of the method described in the next section. In a first step, a taxonomy of DR methods is presented in order to layout the various alternatives that exist for reducing dimensionality and then, in a second step, the reason for choosing PCA over other methods is explained, along with a review of
the theory behind it. Finally, a list of requirements and the main hypothesis behind LSGT are presented in the last subsection.

3.1.1 Taxonomy of Dimensionality Reduction Methods

Dimensionality reduction refers to the process by which an initial set of design variables is reduced to a minimum number, while preserving important trends in the response. This minimum number is known as the intrinsic dimensionality of the problem. According to Lee and Verlesyen [39], DR methods are well established in the fields data analysis, data mining, and machine learning, where they are used to understand and visualize the structure of complex data sets. On the hand, in aerospace engineering, their primary application has been to help solve design problems by reducing the number of design variables that must be considered.

Building upon the work of those authors, Fig. 23 presents a taxonomy of methods for dimensionality reduction. Borrowing terminology from the field of machine learning, they have been classified as either supervised or unsupervised learning. The former is a down-selection process, often referred to as screening in aerospace engineering, whereas the latter is a mapping from one parametric representation to another, best described as a change of coordinates. Salient examples of screening methods include Main Effect Screening (MES) [58] and Elementary Effect Screening (EES) [55], which are common in aerospace according to Keane and Nair [33]. A review of both these methods is provided in Appendix C, for convenience, since they will be employed in later chapters.

Unsupervised learning methods take advantage of inherent relationships that may exist among measured quantities of interest. For example, if one quantity increases or decreases, then other quantities may always increase or decrease proportionally, according to an identifiable relationship. These methods operate on the notion that if the number of parameters required to characterize this relationship is small, then
Figure 23: Taxonomy of unsupervised DR by Lee and Verleysen [38] (modified). Acronyms: PCA, principal component analysis [30]; LDR, linear dimensionality reduction; NLDR, nonlinear dimensionality reduction; Methods: NLM, nonlinear mapping [74]; CCA, curvilinear component analysis [13]; Isomap [80]; GNLM, geodesic NLM [45]; CDA, curvilinear distance analysis [44]; KPCA, Kernel PCA [76]; SDE=MVU, Semidefinite embedding=Maximum variance unfolding [88]; SOM, self-organizing maps [34]; GTM, generative topographic mapping [11]; LLE, locally linear embedding [72]; LE, Laplacian eigenmaps [5]; Isotop [43]; AA NN, auto-associative neural network [25]; MES, main effect screening [57]; EES, elementary effect screening [55].
they can be used as an alternative degrees of freedom to reduce dimensionality. These alternate degrees of freedom are referred to as latent variables (i.e. hidden) because they are not known \textit{a priori} and must be learned; hence the term \textit{unsupervised learning}.

For example, consider the classic Swiss roll example in Fig. 24. Instead of embedding this manifold in three-dimensional space as in Fig. 24(a), the same structure could be represented as a rectangle in two-dimensional space by unrolling the surface using some type of nonlinear mapping, as shown in Fig. 24(b). The purpose of unsupervised learning is thus to establish such a mapping. By contrast, screening methods would simply seek to remove X, Y or Z, which is not possible without losing the structure of the data and, therefore, dimensionality would not be reduced.

Within the branch of unsupervised learning methods in Fig. 23, a further distinction can be made between linear and nonlinear methods. In order to understand the difference, consider a database containing samples of measured quantities describing a system of interest. This data could be unstructured if the system being measured is random (i.e. a cloud of random points in $p$-dimensional space with no inherent relationship between points) or it could be structured, as in the Swiss roll example,
meaning there is an identifiable relationship between the data points. If this relationship can be described with a hyper-plane, then it is linear, otherwise it is nonlinear. The choice of between Linear Dimensionality Reduction (LDR) and Non-Linear Dimensionality Reduction (NLDR) methods depends on the nature of the relationship tying the data together. For example, Principal Component Analysis (PCA) is an LDR method.

Finally, NLDR methods can be further divided into other categories, depending on the manner in which they extract the shape of the manifold embedded in high dimensional space. The interested reader is referred to the text by Lee and Verleysen [39] for more detail. However, in general, those authors explain that beyond 50 dimensions “NLDR methods can suffer from the curse of dimensionality, get confused, and provide meaningless results” [42]. As a consequence, they recommend first using PCA before applying NLDR methods. Furthermore, the review by Maaten et al. [83] concludes that NLDR methods perform well on artificial data sets such as the Swiss roll, but do not outperform traditional PCA when it comes to real data. Given that aerodynamic design problems routinely exceed 50 dimensions and that NLDR does not outperform PCA for real data sets, the present research will focus on the latter.

### 3.1.2 Principal Component Analysis

Given that NLDR methods have been eliminated, one is left with the choice between screening methods and PCA to reduce dimensionality. This subsection first explains why the latter was chosen over the former and then provides a review of PCA.

**Why PCA?**

Design variables are independent by definition. However, given a family of optimal solutions, there will be an inherent relationship between the settings of the design variables that describe the family. If it was possible to identify this relationship beforehand, without the computational expense of optimization, then the parameters
of this relationship could be used as alternative design variables to describe the family. PCA was chosen on the belief that, in aerodynamics, the number of latent variables required to describe this relationship is likely to be less than the minimum number of design variables that would be found using typical screening methods.

This is motivated by the simple observations that often times, in aerodynamics, more than one variable influences the same feature of the geometry; see for example the Non-Uniform Rational B-Spline [47] (NURBS) methodology, where the influence of neighboring control nodes extends over the same surface grid points. The term feature is here defined as an arbitrary subset of connected, surface grid points that define a region on the surface of the body. If the aerodynamic objective function being minimized is sensitive to this particular feature of the geometry, then it will be sensitive to any design variable that imparts a change to it. Thus, if more than one design variable controls this feature, they will vary together as a group upon optimization, indicating that some inherent relationship is tying them together. For instance, if five design variables influenced an important feature of the geometry, screening methods would find that all of them must be retained in the design space. By contrast, methods like PCA might find that only one latent variable is sufficient to characterize the entire group.

However, in order to achieve this, there needs to be a way to identify the relationship without optimizing every design point, which would be counter-productive for a large design space. With the advent of adjoint design methods, the gradient is an efficient way of achieving this because it provides a rank ordering of the design variable at any given point in the design space. Thus, by sampling the gradient globally, throughout the entire design space, it becomes possible to tell what design variables are active at the same time. In other words, the correlation between the partial derivatives is indicative of the relationship between optimal settings of the design variables. At the outcome of this discussion, in response to Research Question 1,
the main hypothesis of this thesis can be summarized as follows:

**Hypothesis 3.** The curse of dimensionality can be overcome cost-effectively by applying principal component analysis to the gradient of the objective function in order to reduce dimensionality. This yields a low-rank transformation that maps a high-dimensional space onto an equivalent, low-dimensional space in which the use gradient-free methods becomes practical.

How PCA works

The theory of PCA is treated in great detail by Jolliffe [30], but a brief summary of the concept is useful nonetheless. Suppose a system is being described using $N$ observations of a random vector $\mathbf{y}$, where each observation $\mathbf{y}_1, \ldots, \mathbf{y}_N$ consists of a single measurement of $p$ descriptors $y_1, \ldots, y_p$. PCA is a linear transformation that allows the same system to be described using a smaller number of descriptors $\mathbf{v}_1, \ldots, \mathbf{v}_q$, where $q \leq p$. Fig. 25(a) shows a notional example for two descriptors $y_1$ and $y_2$, which are clearly positively correlated. However, notice that most of the variation in the data occurs along the main diagonal, while only a small amount occurs in a direction orthogonal to this one. These directions are the so-called principal components and are shown by the arrows. PCA is the change of coordinates that rotates the canonical basis in Fig. 25(a) onto the principal components in Fig. 25(b). The advantage of doing so is that correlation is removed in this new basis and, therefore, it becomes possible to neglect directions that do not contribute significantly to the variance for only a minor loss of information. Ignoring $v_2$, for example, the system can now be described with good approximation using only one dimension $v_1$, as shown in Fig. 25(c) after reconstruction. Note that without the coordinate transformation, $y_1$ and $y_2$ are responsible for equal amounts of variation and, therefore, dimensionality.
cannot be reduced. In order to understand how this works mathematically, suppose the data set $D$ is given by $N$ samples of $y$,

$$
D = \begin{pmatrix}
    y_1^\top \\
    \vdots \\
    y_N^\top
\end{pmatrix} = \begin{pmatrix}
    y_{11} & \cdots & y_{1p} \\
    \vdots & \ddots & \vdots \\
    y_{N1} & \cdots & y_{Np}
\end{pmatrix} \in \mathbb{R}^{N \times p}
$$

(3)

and assume $y_1, \ldots, y_p$ are correlated. An estimator for the mean $\mu$ and covariance matrix $\Sigma$ of $y$ is thus given by:

$$
\bar{y} = \frac{1}{N} \sum_{i=1}^{N} y_i \in \mathbb{R}^p
$$

(4)

$$
S_N = \frac{1}{N-1} \sum_{i=1}^{N} (y_i - \bar{y}) \otimes (y_i - \bar{y}) \in \mathbb{R}^{p \times p}
$$

(5)

Let $\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_p \in \mathbb{R}^p$ denote the eigenvectors of $S_N$ and $\hat{\lambda}_1, \hat{\lambda}_2, \ldots, \hat{\lambda}_p \in \mathbb{R}$ the associated eigenvalues. They are estimators of the true eigenvectors $a_1, a_2, \ldots, a_p$ and eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_p$ of $\Sigma$. The eigenvectors are the principal components (see Jolliffe [30] for proof). Define the matrix of eigenvectors $A$ such that its columns are organized in order of decreasing eigenvalues:

$$
A = \begin{pmatrix}
    \hat{a}_1 & \cdots & \hat{a}_p
\end{pmatrix} \in \mathbb{R}^{p \times p}
$$

(6)

$$
\hat{\lambda}_1 \geq \cdots \geq \hat{\lambda}_p \geq 0
$$
Let \( \mathbf{v} \) denote the representation of \( \mathbf{y} \) rotated onto the principal components. The linear transformation required to obtain \( \mathbf{v} \) is given by:

\[
A : \mathbb{R}^p \rightarrow \mathbb{R}^p, \mathbf{y} \mapsto \mathbf{v}
\]

\[
y = \mathbf{A} \mathbf{v}
\]  

(7)

Notice that \( \mathbf{A}^{-1} = \mathbf{A}^\top \) since \( \mathbf{A} \) is orthogonal. PCA is an eigenvalue decomposition of the covariance matrix \( \mathbf{S}_N \). Therefore, the eigenvalues \( \lambda_i \) have a special meaning: they represent the variance in eigenspace. Eigenvectors with large eigenvalues thus denote principal components responsible for large variation in the data; all others can be ignored for only a minimal lost of information. To determine how many principal components must be retained, the following criteria is usually used [31]:

\[
\Psi(q) = \frac{\sum_{j=1}^{q} \hat{\lambda}_j}{\sum_{j=1}^{p} \hat{\lambda}_j} \geq c
\]

(8)

where \( 0 \leq c \leq 1 \in \mathbb{R} \) is some fraction of the total variation. For example, a good choice might be \( c = 0.95 \) in order to capture 95% of the variability of the gradient. Let \( \mathbf{A}_q \in \mathbb{R}^{p \times q} \) represent the partition of \( \mathbf{A} \) composed of the first \( q \) columns. Dimensionality can then be reduced using the following linear transformation:

\[
\mathbf{A}_q : \mathbb{R}^p \rightarrow \mathbb{R}^q, \mathbf{y} \mapsto \mathbf{v}
\]

\[
\mathbf{v}_q = \mathbf{A}_q^\top \mathbf{y}
\]

(9)

However, notice that the transformation matrix \( \mathbf{A}_q \) is low-rank and, therefore, the mapping is “onto” but not “one-to-one.” Therefore, in order to reconstruct the data, the full rank matrix \( \mathbf{A} \) must be used; but in order to do so, some value must be assumed for the remaining \( p - q \) descriptors \( \mathbf{v}_{q+1}, \ldots, \mathbf{v}_p \). Since they do not contribute significantly to the variation, it is customary to assume zero.

### 3.1.3 Method Functional Decomposition

PCA is a coordinate transformation. Therefore, in order to take advantage of it for aerodynamics shape optimization, the design problem must be reformulated in the
new coordinate system. Doing so involves a series of necessary steps, which will be organized into a formal methodology in the next section. However, in general, PCA always involves the following:

1. Collect data

2. Scale and center the data (if necessary)

3. Apply PCA
   - Compute sample covariance matrix
   - Compute eigenvectors and eigenvalues
   - Organize eigenvectors in order of decreasing eigenvalues
   - Perform coordinate transformation by projecting data onto eigenspace

4. Reduce dimensionality by ignoring eigenvectors with small eigenvalues

5. Reconstruct original data by mapping backwards

These steps represent a functional decomposition of the basic tasks that any method involving PCA must be able to fulfill. In traditional applications of PCA, carrying them out is straightforward. However, when PCA is used with gradient information, certain provisions must be made to preserve important properties of the problem, as described in the next section. Finally, once the problem has been recast into an equivalent coordinate system of lower dimensionality, the next step is to take advantage of reduced dimensionality to explore the design space in ways that were not possible before. However, since PCA is inherently an approximation, it may be judicious to first insert a validation step in order to check that accuracy has not been compromised. These ideas will now be formalized into a rigorous mathematical framework.
3.2 Method: Latent Space Gradient Transformation

This section describes the featured method of this thesis, called Latent Space Gradient Transformation (LSGT). It reduces dimensionality by applying principal component analysis to the sample distribution of the gradient of an objective function. This yields a low-rank transformation that maps a high-dimensional space onto an equivalent, low-dimensional space, which is more affordable to explore. The idea of using a coordinate transformation to reduce dimensionality in aerodynamic problems is not new; see the work by Toal et al. [82] or Viswanath et al. [87] for example. However, the idea of applying PCA to the gradient is novel. Similar ideas have been recently presented by Lukaczyk et al. [51] through independent research and the reader is encouraged to learn about their work. However, the present method differs in three ways. First, it offers a general framework that does not necessarily rely on surrogate modeling (although it enables it). Second, it accounts for both linear and nonlinear behavior. Third, it ensures a one-to-one mapping between the original design space and the so-called, reduced latent space (to be defined shortly). While the method was developed with aerodynamic applications in mind, the ideas are general and applicable to any function for which gradient information is available.

3.2.1 Optimization Problem Setup

The method assumes the following bound constrained optimization problem, where \( f \in \mathbb{R} \) is some smooth objective function evaluated with CFD, \( \nabla_x f \in \mathbb{R}^p \) is computed with an adjoint solver and \( x \in \mathbb{R}^p \) is a vector of \( p \) continuous design variables with upper and lower limit \( x_U \) and \( x_L \). For the time being, it is assumed that equality and inequality constraints are enforced through penalty functions.

Find: \( x^* = \arg \min_x f(x) \) \hspace{1cm} (10)

Subject To: \( x_L \leq x \leq x_U \) \hspace{1cm} (11)
3.2.2 Step 1: Scale and Center Design Space

Let \( D := z \forall [z_L; z_U] \) denote the feasible design space, where \( z \in \mathbb{R}^p \) is a standardized version of \( x \in \mathbb{R}^p \) centered about an arbitrary reference point \( x_0 \in \mathbb{R}^p \):

\[
z = \begin{pmatrix} z_1 \ldots z_p \end{pmatrix}^\top = W (x - x_0) \quad \text{and} \quad \nabla z = W^{-1} \nabla x \quad (12)
\]

The scaling matrix \( W \) is given by,

\[
W = \begin{pmatrix} \frac{1}{|x_{T1}|} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{|x_{Tp}|} \end{pmatrix} \quad (13)
\]

where \( x_{Ti} \) are scaling variables usually taken to represent a typical size for the \( i^{th} \) variable, as described by Vanderplaats [85]. This avoids bias due to inconsistent standards of measurement across different variable types. Finally, the standardized design variables \( z_1, \ldots, z_p \) are assumed to be independent random variables sampled uniformly, while the random variables \( y_1, \ldots, y_p \) are taken as sensitivities with unknown distribution:

\[
y = \begin{pmatrix} y_1 \ldots y_p \end{pmatrix}^\top = \begin{pmatrix} \frac{\partial f}{\partial z_1} \ldots \frac{\partial f}{\partial z_p} \end{pmatrix}^\top = \nabla_z f \in \mathbb{R}^p \quad (14)
\]

3.2.3 Step 2: Collect Gradient Samples

In aerodynamics, if two or more design variables influence the same feature of the geometry and the flow field is sensitive to a change in that feature, then those design variables tend to move together as a group upon optimization. In other words, there is an inherent relationship between their partial derivatives. If the number of parameters required to characterize this relationship is small, then they can be used as an alternative set of design variables to reduce dimensionality. Sampling the gradient is a way to identify this relationship. Let the standardized variables \( z_1, \ldots, z_p \) be sampled uniformly according to \( U(z_L, z_U) \) and let \( z_i \in D \forall i = 1, \ldots, N \) denote the
$i^{th}$ realization of $z$:

$$z_i = \begin{pmatrix} z_1 & \ldots & z_p \end{pmatrix}^\top \in \mathbb{R}^p$$  \hspace{1cm} (15)

Similarly, let $y_i$ denote the corresponding $i^{th}$ realization of the gradient,

$$y_i = \begin{pmatrix} y_1 & \ldots & y_p \end{pmatrix}^\top \in \mathbb{R}^p$$  \hspace{1cm} (16)

and let there be a total of $N$ gradient observations, where $N$ is to be determined:

$$D = \begin{pmatrix} y_1^\top \\ \vdots \\ y_N^\top \end{pmatrix} = \begin{pmatrix} (\nabla_z f(z_1))^\top \\ \vdots \\ (\nabla_z f(z_N))^\top \end{pmatrix} = \begin{pmatrix} \left( \frac{\partial f}{\partial z_1} \right)_1 & \cdots & \left( \frac{\partial f}{\partial z_p} \right)_1 \\ \vdots & \ddots & \vdots \\ \left( \frac{\partial f}{\partial z_1} \right)_N & \cdots & \left( \frac{\partial f}{\partial z_p} \right)_N \end{pmatrix} \in \mathbb{R}^{N \times p}$$  \hspace{1cm} (17)

The sample covariance matrix $S_N$ is given by Eq. 5. It is an estimator for the true covariance matrix $\Sigma$. It can be shown [50, 35] that the quality of the estimator $S_N$ has an asymptotic limit given by:

$$\|S_N - \Sigma\|_\infty \asymp \|\Sigma\| \sqrt{\frac{r_e(\Sigma)}{N}}$$  \hspace{1cm} (18)

where the effective rank is defined according to:

$$r_e(\Sigma) = \frac{\text{Tr}(\Sigma)}{\|\Sigma\|}$$  \hspace{1cm} (19)

The quantities Tr(·) and $\| \cdot \|$ are the trace and matrix norm, respectively. Sampling stops when $\|S_N - \Sigma\|_\infty$ reaches asymptotic convergence. For arbitrary, high-dimensional distributions, Vershynin [86] shows that the number of samples required to obtain a good estimator is on the order of:

$$N = \mathcal{O}(p \log(p))$$  \hspace{1cm} (20)

### 3.2.4 Step 3: Apply Principal Component Analysis

Let $\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_p \in \mathbb{R}^p$ denote the eigenvectors of $S_N$ and $\hat{\lambda}_1, \hat{\lambda}_2, \ldots, \hat{\lambda}_p \in \mathbb{R}$ the associated eigenvalues. They are estimators of the true eigenvectors $a_1, a_2, \ldots, a_p \in \mathbb{R}^p$ and eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_p \in \mathbb{R}$. The eigenvectors are the so-called principal
components. Define the matrix of eigenvectors $A$ such that its columns are organized in order of decreasing eigenvalues:

$$A = \begin{pmatrix} \hat{a}_1 & \ldots & \hat{a}_p \end{pmatrix} \in \mathbb{R}^{p \times p}$$  \hspace{1cm} (21)

$$\hat{\lambda}_1 \geq \cdots \geq \hat{\lambda}_p \geq 0$$

Let $\xi$ and $\upsilon = \nabla_\xi f$ denote the representations of $z$ and $y$ rotated onto the principal components, respectively. For convenience, the hyperspace spanned by the principal component will be referred to as the *latent space* and $\xi$ will be called *latent variables*. The linear transformation required to obtain $\upsilon$ is given by:

$$A : \mathbb{R}^p \rightarrow \mathbb{R}^p, \ y \mapsto \upsilon$$

$$y - \bar{y} = A(\upsilon - \bar{\upsilon}) \iff \nabla_z = A\nabla_\xi$$  \hspace{1cm} (22)

where $\bar{\upsilon} = A^\top \bar{y}$ is the mean of the gradient in latent space. Notice that $A^{-1} = A^\top$ since $A$ is orthogonal. Similarly, the transformation for $\xi$ is,

$$B : \mathbb{R}^p \rightarrow \mathbb{R}^p, \ z \mapsto \xi$$

$$\xi = B^\top z$$  \hspace{1cm} (23)

where it is found that $B = A$ using the chain rule,

$$\nabla_z = \begin{pmatrix} \frac{\partial \xi_1}{\partial z_1} & \ldots & \frac{\partial \xi_p}{\partial z_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial \xi_1}{\partial z_p} & \ldots & \frac{\partial \xi_p}{\partial z_p} \end{pmatrix} \nabla_\xi = \nabla_z \xi^\top \nabla_\xi = \nabla_z z^\top B \nabla_\xi = B \nabla_\xi$$  \hspace{1cm} (24)

and comparing to Eq. 22:

$$\nabla_z = B \nabla_\xi = A \nabla_\xi \Rightarrow B = A$$  \hspace{1cm} (25)

So far, dimensionality has not yet been reduced; the optimization problem in Eqs. 10–11 has just been restated in an equivalent coordinate system:

Find: $\xi^* = \arg \min_\xi f(A\xi)$  \hspace{1cm} (26)

Subject to: $z_L \leq A\xi \leq z_U$  \hspace{1cm} (27)
3.2.5 Step 4: Reduce Dimensionality

PCA is an eigenvalue decomposition of the covariance matrix $S_N$. Therefore, the eigenvalues $\lambda_i$ have a special meaning: they represent the variance of the gradient in eigenspace. Eigenvectors with large eigenvalues denote principal components along which the gradient varies significantly, whereas eigenvectors with small eigenvalues indicate principal components along which it remains approximately constant. Since changes in the gradient are indicative of nonlinear behavior, PCA yields a rank ordering of directions in which the response is most nonlinear. However, care must be taken not to interpret small eigenvalues to imply negligible change in the response. They only indicate small curvature; but the average slope in that direction could still be steep. Therefore, in order to reduce dimensionality, both the mean and standard deviation of the gradient distribution must be accounted for.

Principal Components Responsible for Nonlinear Change in the Response

To determine which principal components have an important effect on the gradient, i.e. directions in which the response behaves nonlinearly, it is proposed to use the following equation:

$$\Psi(t) = \frac{\sum_{j=1}^{t} \sqrt{\hat{\lambda}_j}}{\sum_{j=1}^{p} \sqrt{\hat{\lambda}_j}} \geq c \quad (28)$$

where $0 \leq c \leq 1 \in \mathbb{R}$ is some fraction of the total variation and $t \leq p$. For example, a good choice might be $c = 0.95$ in order to capture 95% of the variability of the gradient. Notice that usually, in PCA, one uses the variance $\hat{\lambda}$ rather than the standard deviation $\sqrt{\hat{\lambda}}$; see Jolliffe [31] for example. The variance measures the mathematical dispersion of the data relative to the mean. However, its units are squared whereas those of the mean are not. Therefore, in order to distinguish between directions in which the function is predominantly linear from those in which it is predominantly nonlinear, it is necessary to keep units consistent by working with the
standard deviation. Otherwise, PCA tends to underestimate the number of principal components needed to account for all nonlinear behavior. Finally, for convenience of notation, let $A_t$ denote the first $t$ columns of $A$ and $\xi$ such that:

$$\xi_t = \left( \xi_1 \ldots \xi_t \right)^\top$$  \hspace{1cm} (29)

$$A_t = \left( \hat{a}_1 \ldots \hat{a}_t \right) \in \mathbb{R}^{p \times t}$$  \hspace{1cm} (30)

$$\hat{\lambda}_1 \geq \cdots \geq \hat{\lambda}_t \geq 0$$

Principal Components Responsible for Linear Change in the Response

Of the remaining latent variables, it is desired to capture the ones responsible for steep, linear changes in the response. This can be accomplished by looking at the mean of the gradient distribution. Let $J(s)$ be a selector of $s \leq p - t$ indices such that:

$$J(s) \subset \{t+1, \ldots, p\} \quad \text{and} \quad |J| = s \leq p - t$$  \hspace{1cm} (31)

The following criterion can then be used to select $J(s)$:

$$\Psi(s) = \frac{\sum_{j \in J(s)} \left| \mathbb{E} \left( \frac{\partial f}{\partial \xi_j} \right) \right| + \sum_{j=1}^t \left| \mathbb{E} \left( \frac{\partial f}{\partial \xi_j} \right) \right|}{\sum_{j=1}^p \left| \mathbb{E} \left( \frac{\partial f}{\partial \xi_j} \right) \right|} \geq c$$  \hspace{1cm} (32)

For convenience of notation, let $A_s$ denote the corresponding columns of $A$ and $\xi$ such that:

$$\xi_s = (\xi_i) \forall i \in J(s)$$  \hspace{1cm} (33)

$$A_s = (a_i) \forall i \in J(s)$$  \hspace{1cm} (34)

3.2.6 Step 5: Establish Forward and Backward Mappings

In order to take advantage of PCA, it is necessary to establish a forward and backward mapping between the original design space and the reduced latent space. Mapping forward is straightforward; however, mapping backward is more involved. This section explains how to carry out both tasks.
Forward Mapping Onto Reduced Latent Space

Let \( \Gamma \) define the set of all indices identified above,

\[
\Gamma = \{1, \ldots, t\} \cup J(s) \quad \text{and} \quad |\Gamma| = q
\]  

(35)

and define the matrix \( A_q \in \mathbb{R}^{p \times q} \) such that:

\[
A_q = (a_i) \quad \forall i \in \Gamma \quad \Leftrightarrow \quad A_q = \begin{pmatrix} A_t & A_s \end{pmatrix}
\]  

(36)

Similarly, let \( \xi_q \) denote the corresponding latent variables. The eigenspace spanned by the columns of \( A_q \) will be called the reduced latent space. A forward mapping onto reduced latent space is thus given by:

\[
A_q : \mathbb{R}^p \rightarrow \mathbb{R}^q, \ (z, y) \mapsto (\xi, \nu)
\]

\[
\xi_q = A_q^\top z
\]  

(37)

\[
\nu - \bar{\nu} = A_q^\top (y - \bar{y}) \quad \Leftrightarrow \quad \nabla \xi_q = A_q^\top \nabla z
\]  

(38)

Notice that the transformation matrix \( A_q \) is low-rank and, therefore, the mapping is “onto” but not “one-to-one.”

Backward Mapping Onto Original Design Space

Let \( \Omega \) denote the feasible space obtained by projecting \( \mathcal{D} \) onto reduced latent space:

\[
\Omega := \{ \xi_q = A_q^\top z \quad \forall \quad z \in \mathcal{D} \}
\]  

(39)

For convenience, let \( A_r \) denote the remaining \( p - q \) columns of \( A \) such that:

\[
A = \begin{pmatrix} A_q & A_r \end{pmatrix} \quad \text{and} \quad \xi = \begin{pmatrix} \xi_q & \xi_r \end{pmatrix}^\top
\]  

(40)

Therefore, Eq. 23 can be re-written as:

\[
z = A_q \xi_q + A_r \xi_r
\]  

(41)
Since the response is not sensitive to $\xi_r$, it can be defaulted to any convenient value, say $\xi_r^*$, in order to obtain a mapping from reduced latent space that is “one-to-one.” Typically, in PCA, this is achieved by setting $\xi_r^* = 0$. However, Lukaczyk et al. [51] correctly point out that such a simple map could yield a point such that $z \not\in D$, even though $\xi_q \in \Omega$, therefore eliminating a portion of the feasible design space. In order to resolve this issue, those authors proposed to solve an auxiliary linear programming problem to recover a feasible value of $\xi_r$ such that $z \in D$ if $\xi_q \in \Omega$. However, their formulation employs a zero objective function and so the solution is indeterminate, as they noted. This issue can be resolved using the following convex Quadratic Program (QP) instead:

\[
\text{Find: } \xi_r^* = \arg\min_{\xi_r} \xi_r^\top \xi_r \tag{42}
\]

\[
\text{Subject To: } z_L \leq A_q \xi_q + A_r \xi_r \leq z_U \tag{43}
\]

This QP admits a unique minimum over the set of all feasible solutions. If it exists, the solution is obtained nearly instantaneously since it does not depend on the objective function. If it does not, the point is infeasible $\xi_q \not\in \Omega$. The solution to Eqs 42–43 amounts to finding the value of $\xi_r$ closest to zero such that $\xi_q$ remains feasible, as illustrated notionally in Fig. 26 for two dimensions. Finally, it should be clear to the reader that Eqs. 41–43 do not constitute an inverse map for Eqs. 37; it is only an approximation, albeit a good one.

### 3.2.7 Step 6: Check Model Accuracy

Before proceeding to the next steps, it is necessary to first assess the quality of approximate model that results due to reducing dimensionality. This step can be seen as a sanity check. Let $f(\xi_q)$ denote the approximation for $f(z)$ due to the mapping established in the previous step, where the reader is reminded that $\xi_r^*$ is a
function of \( \xi_q \):

\[
f(\xi_q) \approx f(z)
\]

(44)

Let \( f_i \forall i = 1, \ldots, N \) denote the objective function values associated with the dataset \( D \) in Eq. 17. The error due to the model approximation is therefore given by:

\[
\epsilon_i = f_i - f(A_q^\top z_i) \quad \forall i = 1, \ldots, N
\]

(45)

This information can then be used to compute the mean \( \mu_\epsilon \) and standard deviation \( \sigma_\epsilon \) of the error distribution, as well as the R-squared value of the model. If prediction accuracy does not meet acceptable user tolerance, steps 4–6 should be repeated until satisfaction using increasing values of \( c \) in Eqs. 28 and 32.

3.2.8 Step 7: Design Space Exploration

Presuming that dimensionality has been reduced sufficiently, gradient-free methods can now be used to explore the design space. The original optimization problem in Eqs. 10–11 is now approximately equivalent to solving the following in reduced latent
Find: \( \xi_q^* = \arg \min_{\xi_q} f(\xi_q) \) \hspace{1cm} (46)

Subject To: \( z_L \leq A_q\xi_q + A_r\xi_r^* \leq z_U \) \hspace{1cm} (47)

where Eq. 47 represents nonlinear constraints because \( \xi_r^* \) is a nonlinear function of \( \xi_q \) according to Eqs. 42–43. Any gradient-free method can be used to solve Eqs. 46–47; this methodology does not assume one in particular. For example, surrogate models could be used.

### 3.2.9 Step 8: Gradient-Based Optimization

The solutions found in the previous step constitute a set of “good” initial guesses for gradient-based optimization. They can thus be used as starting points to solve Eqs. 10–11 with increased chances of finding more than one local minimum.

### 3.3 Concepts & Limitations

This section illustrates how LSGT works using simple, analytical examples that are easy to visualize. Its purpose is to explain fundamental concepts and point out known limitations. In order to so, consider the general form of a Response Surface Equation (RSE) in \( p \)-dimensions,

\[
 f(\mathbf{x}) = \beta_0 + \sum_{i=1}^{p} \beta_i x_i + \sum_{i=1}^{p} \beta_{ii} x_i^2 + \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} \beta_{ij} x_i x_j + \sum_{i=1}^{m} \gamma_i \phi_i(\mathbf{x}) \tag{48}
\]

and it’s gradient,

\[
 \nabla f(\mathbf{x}) = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} + \begin{pmatrix} 2\beta_{11} & \cdots & \beta_{1p} \\ \vdots & \ddots & \vdots \\ \beta_{1p} & \cdots & 2\beta_{pp} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_p \end{pmatrix} + \sum_{i=1}^{m} \gamma_i \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{pmatrix} \nabla \phi_i(\mathbf{x}) \tag{49}
\]
where the basis functions $\phi_i$ correspond to $m$ higher-order terms (e.g. $x_1^2x_2$, $x_1x_2x_3$, etc.), denoted HOT. Provided with a sufficient number of terms, RSE can be made to fit any smooth response and, therefore, represent a general model that is well suited for illustration.

### 3.3.1 Effect of Correlation

PCA is a linear dimensionality reduction method, which means that it assumes linear correlation between variables being observed. In other words, it assumes that the nonlinear terms in Eq. 49 are zero. The purpose of this subsection is to explain what limitations this imposes on the LSGT method.

#### Linear Correlation

For simplicity, consider the following quadratic RSE in two dimensions:

$$ f = x_1^2 + x_2^2 + \beta_{12} \cdot x_1x_2 \quad \forall x_1, x_2 \in [-1; 1] $$

$$ \nabla f = \begin{pmatrix} 2 & \beta_{12} \\ \beta_{12} & 2 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} $$

The second-order interaction $\beta_{12} \cdot x_1x_2$ is responsible for linear coupling in the gradient, which results in linear correlation when it is sampled randomly. For instance, let $x_1$ and $x_2$ be sampled uniformly according to $U(-1, 1)$. The mean and covariance matrix of the gradient is then given by:

$$ \mathbb{E}(\nabla f) = \begin{pmatrix} 2 & \beta_{12} \\ \beta_{12} & 2 \end{pmatrix} \cdot \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} $$

$$ \text{Cov}(\nabla f - \mathbb{E}(\nabla f)) = \begin{pmatrix} 2 & \beta_{12} \\ \beta_{12} & 2 \end{pmatrix} \cdot \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix} \cdot \begin{pmatrix} 2 & \beta_{12} \\ \beta_{12} & 2 \end{pmatrix} $$
Figure 27: Projection along first principal component, $\mathbf{a}_1$, $\beta_{12} = 1.5$
If \( x_i \sim U(-1, 1) \) \( \forall i = 1, 2 \), then \( \mu_1 = \mu_2 = 0 \) and \( \sigma^2_1 = \sigma^2_2 = 1/3 \) such that:

\[
\mathbb{E}(\nabla f) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}
\]

(54)

\[
\text{Cov}(\nabla f - \mathbb{E}(\nabla f)) = \frac{1}{3} \cdot \begin{pmatrix}
4 + \beta_{12}^2 & 4\beta_{12} \\
4\beta_{12} & 4 + \beta_{12}^2
\end{pmatrix}
\]

(55)

When \( \beta_{12} \neq 0 \), the eigenvalues are given by:

\[
\begin{pmatrix}
\lambda_1 & 0 \\
0 & \lambda_2
\end{pmatrix} = \begin{pmatrix}
(\beta_{12} + 2)^2 & 0 \\
0 & (\beta_{12} - 2)^2
\end{pmatrix}
\]

(56)

Similarly, the eigenvectors are given by:

\[
A = \begin{pmatrix}
a_1 \\ a_2
\end{pmatrix} = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}
\]

(57)

Results are illustrated visually in Figs. 27(a)–27(b) for the case where \( \beta_{12} = 1.5 \) such that \( \lambda_1 = 12.5 \) and \( \lambda_2 = 0.25 \). Since \( \lambda_1 \gg \lambda_2 \), most of the variation in the gradient occurs along the first principal component \( a_1 = (1, 1) \). As a consequence, \( a_2 = (-1, 1) \) can be ignored for only a minor loss of information, as shown in Figs. 27(c)–27(d). Furthermore, the noise due to ignoring \( \xi_2 \) is small in comparison to the variability due to \( \xi_1 \) and, therefore, a good fit can be obtained, as shown. In general, the success of LSGT requires many second-order interactions \( x_i x_j \) in order to yield the type of linear correlation illustrated in this example. Fortunately, this is usually the case for most engineering functions according the sparsity-of-effects principle [59].

**Zero Correlation**

When \( \beta_{12} = 0 \), there is no correlation and \( \lambda_1 = \lambda_2 = 4 \) in the previous example. Dimensionality can therefore not be reduced in that case. PCA simply finds that any basis is as good as another, as illustrated in Figs. 28(a)–28(b). A design problem with zero correlation in the gradient is therefore ill suited for the LSGT method.
Nonlinear Correlation

Consider now the effect of nonlinear correlation using the following example:

\[ f = x_1^2 + x_1^2 x_2^2 + x_1 x_2 \quad \forall \ x_1, x_2 \in [-1; 1] \]  

(58)

This results in nonlinear coupling of the gradient, as shown in Fig. 29:

\[
\nabla f = \begin{pmatrix}
2 & 1
1 & 0
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2
\end{pmatrix}
+ \begin{pmatrix}
2 & 1 & 0 & 1
1 & 0 & 2 & 0
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2 \\
x_1 x_2^2 \\
x_1^2 x_2
\end{pmatrix}
\]

(59)

Even though this violates the assumption behind PCA that correlation is linear, it is not an impediment to the method. PCA simply returns the best linear transformation that maximizes variation in the gradient, as shown Fig. 29. In the worst case, if nonlinear effects are so pronounced that variability in the gradient is the same in every direction, then dimensionality cannot be reduced as all eigenvectors are of equal importance. However, the sparsity-of-effects principal [59] ensures that the number of
directions in which the response exhibits high-order interactions is likely to be small. As a result, not only will PCA reduce dimensionality effectively in most cases, but it also provides a way to target directions in which the response is most nonlinear.

### 3.3.2 Effect of Non-Zero Mean Gradient

So far, all examples yield a zero mean gradient, $\mathbb{E}(\nabla_z f) = 0$, but consider what happens when $\mathbb{E}(\nabla_z f) \neq 0$ due to adding linear terms in Eq. 51:

$$f = x_1^2 + x_2^2 + \beta_{12} \cdot x_1 x_2 + \beta_1 \cdot x_1 + \beta_2 \cdot x_2 \quad \forall x_1, x_2 \in [-1; 1] \quad (60)$$

The mean of the gradient is therefore no longer zero:

$$\mathbb{E}(\nabla f) = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} + \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} \quad (61)$$

---

**Figure 29:** Effect of nonlinear correlation on PCA
Figure 30: Projection along 1st principal component, $\beta_1 = 0.9$, $\beta_2 = -\beta_1$, $\beta_{12} = 1.5$

However, the covariance does not change:

$$
\text{Cov}(\nabla f - \mathbb{E}(\nabla f)) = \begin{pmatrix}
2 & \beta_{12} \\
\beta_{12} & 2
\end{pmatrix}
\cdot
\begin{pmatrix}
\sigma_1^2 & 0 \\
0 & \sigma_2^2
\end{pmatrix}
\cdot
\begin{pmatrix}
2 & \beta_{12} \\
\beta_{12} & 2
\end{pmatrix}
\cdot
\begin{pmatrix}
4 + \beta_{12}^2 & 4\beta_{12} \\
4\beta_{12} & 4 + \beta_{12}^2
\end{pmatrix}
$$

As a result, the principal components and associated eigenvalues are still given by Eq. 57 and 55, respectively. Let $\beta_{12} = 1.5$ such $\lambda_1 = 12.5$ and $\lambda_2 = 0.25$ as before, but consider what happens when $\beta_1 = 0.9$ and $\beta_2 = -\beta_1$ in Fig. 30. By comparison to Fig. 27(d), it can be seen that even though $\lambda_1 >> \lambda_2$, the second principal component has a large effect on the response. This demonstrates that applying PCA to the gradient alone cannot capture important linear effects, unless certain provisions are made in Eq. 32 to account for them separately.

3.3.3 Effect of Centering the Design Space

The effect of centering the design space is illustrated in Fig. 31, which shows how various values of $x_0$ alter the 1D approximation of Eq. 51 due to ignoring the second
Figure 31: Effect of choosing $x_0$
principal component. The left column shows the location of $x_0$ in latent space (i.e. the selected origin for the basis), and the right column shows the predicted response after ignoring the second principal component, as before in Fig. 27(d). While the variability of the error due to ignoring $\xi_2$ does not change, it can be seen that the choice of $x_0$ determines how the approximation cuts through this variability, shifting the prediction curve up or down in Figs. 31(b), 31(d) and 31(f). The effect of the QP problem in Eqs. 42–43 is also clearly visible in Figs. 31(a), 31(c) and 31(e), where it can be seen that it “bends” the reduced latent space along the boundaries so that all values of $\xi_1$ can be reached. However, the QP can also introduce local optima that are not truly there; see for example Fig. 31(d). While this is an admitted shortcoming of LSGT, it is not foreseen to be a deterrent since the ensuing error will always be on the order of the “background noise,” which is presumably less than or equal to user tolerance according to step 6. Note that this effect is minimized by taking $x_0$ at the center of the design space.

3.3.4 Synthesis

This section provided insight into concepts and known limitations behind the LSGT method. They are summarized as follows. First, scaling the design space is necessary when it contains heterogeneous variable types that are measured on scales of different magnitude. Second, centering the design space at the mid-range of each dimension minimizes calls to Eqs. 42–43 and, therefore, minimizes the risk of artificial local minima, as explained. Third, success of LSGT depends on the presence of many second-order interactions, because they yield linear correlation in the gradient and PCA is a linear method. Fortunately, this is the case for most engineering problems according to the sparsity-of-effects principal \[59\]. Finally, even though PCA is a linear method, nonlinear correlation is not an impediment. In fact, applying PCA to the gradient is a useful way to identify directions in which the response is most nonlinear.
CHAPTER IV

STEP-BY-STEP CANONICAL EXAMPLE

The ideas, concepts and formal methodology presented in the previous chapter are not necessarily intuitive. Therefore, before applying them to applied aerodynamic problems, it is useful to first illustrate the method on a simple example, taken to be weight estimation of a light aircraft wing. Consider the following, empirical formula taken from the conceptual design text by Raymer [68]:

\[
W_w = 0.036 \cdot S_w^{0.758} W_{fw}^{0.0035} \left( \frac{A}{\cos^2 \frac{\Delta x}{180}} \right)^{0.6} q^{0.006} \lambda^{0.04} \left( \frac{100 \frac{L}{c}}{\cos \frac{\Delta x}{180}} \right)^{-0.3} (N_z W_{dg})^{0.49} + S_w W_p
\]

(63)

This example is the same one used in the text by Forrester et al. [14] to exemplify a specific screening method called Elementary Effect Screening (EES), which was developed by Morris [55] and is summarized Appendix C. This test function thus offers the advantage that it is a real engineering function that has already been studied by others; however, it is not high-dimensional (i.e. \( p > 50 \)). Application to high-dimensional problems is deferred to the next chapters in favor of focusing on simplicity. A definition of all design variables is shown in Table 1, where the baseline values are representative of a Cessna C172 Skyhawk aircraft. The significant variables are known to be \( S_w, A, t/c, N_z, W_{dg} \) and \( W_{fw} \) [14]; all others are negligible.

4.1 Optimization Problem Setup

Before LSGT can be applied, the problem must first be defined. The optimization problem is given as follows, using the wing weight in Eq. 63 as the objective function,

Find: \( \mathbf{x}^* = \arg \min_{\mathbf{x}} f(\mathbf{x}) \equiv W_w \)  

(64)

Subject To: \( \mathbf{x}_L \leq \mathbf{x} \leq \mathbf{x}_U \)  

(65)
### Table 1: Definitions and ranges of design variables (same as Forrester et al. [14])

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
<th>Baseline</th>
<th>Minimum</th>
<th>Maximum</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_w$</td>
<td>Wing area</td>
<td>174</td>
<td>150</td>
<td>200</td>
<td>ft$^2$</td>
</tr>
<tr>
<td>$W_{fw}$</td>
<td>Fuel weight stored in wing</td>
<td>252</td>
<td>220</td>
<td>300</td>
<td>lb</td>
</tr>
<tr>
<td>$A$</td>
<td>Aspect ratio</td>
<td>7.52</td>
<td>6</td>
<td>10</td>
<td>—</td>
</tr>
<tr>
<td>$\Lambda$</td>
<td>Quarter-chord sweep</td>
<td>0</td>
<td>-10</td>
<td>10</td>
<td>deg</td>
</tr>
<tr>
<td>$q$</td>
<td>Dynamic pressure at cruise</td>
<td>34</td>
<td>16</td>
<td>45</td>
<td>lb/ft$^2$</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Taper ratio</td>
<td>0.672</td>
<td>0.5</td>
<td>1</td>
<td>—</td>
</tr>
<tr>
<td>$t/c$</td>
<td>Airfoil thickness to chord ratio</td>
<td>0.12</td>
<td>0.08</td>
<td>0.18</td>
<td>—</td>
</tr>
<tr>
<td>$N_z$</td>
<td>Ultimate load factor</td>
<td>3.8</td>
<td>2.5</td>
<td>6</td>
<td>—</td>
</tr>
<tr>
<td>$W_{dg}$</td>
<td>Design gross weight</td>
<td>2000</td>
<td>1700</td>
<td>2500</td>
<td>lb</td>
</tr>
<tr>
<td>$W_p$</td>
<td>Paint weight</td>
<td>0.064</td>
<td>0.025</td>
<td>0.08</td>
<td>lb</td>
</tr>
</tbody>
</table>

**Figure 32:** $\frac{\partial f}{\partial \Lambda}$ vs. $\Lambda$ holding all other variables fixed at baseline
It is fully acknowledged that aerodynamic constraints should be taken into account when solving a structural optimization problem; however this was is not the purpose of this chapter. The goal is simply to illustrate how the method is applied. In addition, it should be noted that the objective function is not smooth everywhere with respect to $\Lambda \in \mathbb{R}$, where $C$ is a constant term involving the other variables:

$$\frac{\partial f}{\partial \Lambda} = C \cdot \frac{180}{\pi} \left( \frac{1}{\cos \frac{\Lambda \cdot \pi}{180}} \right)^{0.7} \left( \frac{1}{\cos^2 \frac{\Lambda \cdot \pi}{180}} \right)^{0.6} \sin \frac{\Lambda \cdot \pi}{180}$$ (66)

However, as shown in Fig. 32, it is smooth over the ranges of the design space. Using the information in Table 1, the design vector and associated bound constraints are thus defined as:

$$\begin{align*}
\mathbf{x} &= \begin{pmatrix}
S_w \\
W_{fw} \\
A \\
\Lambda \\
q \\
\lambda \\
\frac{t}{c} \\
N_z \\
W_{dg} \\
W_p
\end{pmatrix} & \quad \mathbf{x}_L = \begin{pmatrix}
150 \\
220 \\
6 \\
-10 \\
16 \\
0.5 \\
0.08 \\
2.5 \\
1700 \\
0.025
\end{pmatrix} & \quad \mathbf{x}_U = \begin{pmatrix}
200 \\
300 \\
10 \\
10 \\
45 \\
1 \\
0.18 \\
6 \\
2500 \\
0.08
\end{pmatrix}
\end{align*}$$ (67)

### 4.2 Step 1: Scale and Center Design Space

The first step in the process is to scale and center the design space. This step is critical in order to correctly identify the relative order of importance of each design variable. For example, an educated guess for the design gross weight $W_{dg}$ for this class of airplanes is on the order of $\mathcal{O}(10^3)$, whereas an educated guess for thickness-to-chord ratio $t/c$ is $\mathcal{O}(10^{-1})$. These are two drastically different scales. As a consequence, a
change of 0.01 units of measurement is meaningful for \( t/c \) but not for \( W_{dg} \). In other words, changing the gross weight by \( \pm 0.01 \) will hardly be noticeable, but modifying the thickness-to-chord ratio by \( \pm 0.01 \) will change the wing weight significantly. Does this mean wing weight is not sensitive to design gross weight? Of course not; one would be hard-pressed to make that case. However, if one were to compute the gradient without scaling, that is the result that would be found (see Appendix D). This can all be solved by scaling the design variables appropriately, using a typical value. For example, in this problem, a typical value \( x_T \) for the design variables and the corresponding scaling matrix \( W \) is given as follows:

\[
x_T = \begin{pmatrix}
100 \\
100 \\
1 \\
1 \\
10 \\
0.1 \\
0.1 \\
1 \\
1000 \\
0.01
\end{pmatrix}
\Rightarrow
W = \begin{pmatrix}
\frac{1}{100} \\
\frac{1}{100} \\
\frac{1}{1} \\
\frac{1}{1} \\
\frac{1}{10} \\
\frac{1}{0.1} \\
\frac{1}{0.1} \\
\frac{1}{1} \\
\frac{1}{1000} \\
\frac{1}{0.01}
\end{pmatrix}
\]

One may check Appendix D to verify that scaling by typical value provides the right result. Finally, the reference point is taken at the center of the design space:

\[
x_0 = \frac{x_L + x_U}{2}
\]

4.3 Step 2: Sample Gradient

The convergence history of the estimation error for the sample covariance matrix is shown in Fig. 33. It can be seen that after 100 gradient samples, the error between
the sample covariance matrix and the true covariance matrix is no longer changing and the estimation error has dropped three orders of magnitude. This implies that there is no benefit to take any additional samples and, therefore, sampling may stop. Finally, recall that PCA will only be effective in the next step if there is correlation in the gradient. Upon visual inspection of the scatter plot matrix in Fig. 34, it can be verified that this is indeed the case for this problem. For example, it can clearly be seen that there exists a linear correlation between $\partial f/\partial z_1$ and $\partial f/\partial z_2$ and similarly for many others. The presence of correlation implies that PCA is likely to be helpful for this problem.

4.4 Step 3: Apply Principal Component Analysis

The gradient information collected in the previous step can now be used for PCA. The coefficients associated with the resulting principal components are visualized in Fig. 35. The columns correspond to specific principal components $a_1, \ldots, a_{10}$,
Figure 34: Scatter plot matrix for $\nabla_z f$
while the rows are the coefficients associated with each design variable for a given principal component. They have been color coded according to their value. This matrix thus shows the active design variables along any given direction (i.e. principal component). For example, it can be seen that along the first principal component, which explains most of the variation in the data, the driving design variables are $S_w$, $W_{dg}$ and $t/c$. Intuitively, this result makes sense since one would expect those variables to be important drivers of wing weight.

### 4.5 Step 4: Reduce Dimensionality

In order to reduce dimensionality, one must consider both the mean and standard deviation of the gradient distribution, expressed in latent space. This is most easily visualized using Pareto plots, as shown in Fig. 36. Each plot shows a rank ordering of mean and standard deviation, respectively, associated with each partial derivative $\partial f/\partial \xi_i \forall i = 1, \ldots, 10$. They provide an easy way to visualize which principal components are responsible for nonlinear behavior and which ones are responsible
for important linear change. For example, in this problem, it can be seen that the first principal component denotes a large mean and standard deviation relative to the others. This implies that most of the nonlinear behavior is concentrated along the first principal component and, in addition, there is also a steep average slope in that direction. Notice that the values of the $y$-axis in Fig. 36(b) are much larger than in Fig. 36(b). This implies that this function is predominantly linear with only slight curvature. Finally, in this problem, it can be seen that 99% of the total change is explained by the first $q = 6$ principal components, which represents a 40% dimensionality reduction.

4.6 Step 5–6: Establish Mapping & Check Model Accuracy

Once the previous steps have been completed, the original function can be approximated using only the first six principal components. In order to do so, a suitable mapping between $\xi$ and $z$ must first be created, as already explained in the previous chapter. This mapping occurs behind the scenes, in a computer program, and the reader may refer to the function map.m in Appendix E for an example. Once this is done, the accuracy of such a model remains to be verified. In this example, this was
Figure 37: Model validation error for $f(\xi_q)$ accomplished by taking 1000 random samples in the scaled design space $z_i$, computing the true value of the objective $f(z_i) \forall i = 1, \ldots, 1000$ for each sample, projecting each sample onto reduced latent space $\xi_q$ and, finally, using the map to evaluate the approximate model $f(\xi_q) \forall i = 1, \ldots, 1000$. The results are shown in Fig. 37, where it can be seen that the prediction is excellent. The average wing weight prediction error was $\pm 0.84$ lb with a bias of 0.25 lb.

4.7 Step 7–8: Design Space Exploration & Optimization

Finally, in order to explore the design space, it was decided to use a third-order Response Surface Model (RSE). In order to do so, a latin hypercube sampling plan consisting of 1000 training samples was constructed in reduced latent space, where dimensionality was $q = 6$. This sampling plan is shown in Fig. 38 in the form of a scatter plot matrix. Notice that bound constraints have been respected, as can be seen from the fact the sampling plan is bounded by linear constraints. An additional 1000 samples were collected for validation and the associated validation error of the surrogate models is shown in Fig. 39. Notice that the surrogate model $\hat{f}(\xi_q)$ is an
Figure 38: Scatter plot matrix of sampling plan

Figure 39: Model validation error for \( \hat{f}(\xi_q) \)
Figure 40: Sensitivity plot of wing weight, holding all other variables fixed to baseline
Table 2: Comparison of predicted and actual minimum

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
<th>Predicted</th>
<th>Actual</th>
<th>Error</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$W_w$</td>
<td>Wing weight</td>
<td>122.54</td>
<td>123.25</td>
<td>5.41</td>
<td>lb</td>
</tr>
<tr>
<td>$S_w$</td>
<td>Wing area</td>
<td>150</td>
<td>150</td>
<td>0</td>
<td>ft²</td>
</tr>
<tr>
<td>$W_{fw}$</td>
<td>Fuel weight stored in wing</td>
<td>220</td>
<td>220</td>
<td>0</td>
<td>lb</td>
</tr>
<tr>
<td>$A$</td>
<td>Aspect ratio</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>—</td>
</tr>
<tr>
<td>$\Lambda$</td>
<td>Quarter-chord sweep</td>
<td>3.7</td>
<td>0</td>
<td>3.7</td>
<td>deg</td>
</tr>
<tr>
<td>$q$</td>
<td>Dynamic pressure at cruise</td>
<td>16</td>
<td>16</td>
<td>0</td>
<td>lb/ft²</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Taper ratio</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>—</td>
</tr>
<tr>
<td>$t/c$</td>
<td>Airfoil thickness to chord ratio</td>
<td>0.2</td>
<td>0.08</td>
<td>0</td>
<td>—</td>
</tr>
<tr>
<td>$N_z$</td>
<td>Ultimate load factor</td>
<td>2.5</td>
<td>2.5</td>
<td>0</td>
<td>—</td>
</tr>
<tr>
<td>$W_{dg}$</td>
<td>Design gross weight</td>
<td>1700</td>
<td>1700</td>
<td>0</td>
<td>lb</td>
</tr>
<tr>
<td>$W_p$</td>
<td>Paint weight</td>
<td>0.025</td>
<td>0.025</td>
<td>0</td>
<td>lb</td>
</tr>
</tbody>
</table>

approximation of $f(\xi_q)$, which is already an approximation of $f(z)$ itself. It can be seen that the RSE fits are excellent and the average wing weight prediction error was measured at ±1.70 lb with a bias 0.21 lb. The corresponding sensitivity plots taken about the baseline are shown in Fig. 40. It can be seen that the surrogate model does a good job predicting the right trend and that they are mostly linear as concluded from the Pareto plots in step 4.

Finally, the surrogate model was used to explore the design space in search for a local minimum. In this case, there was only one, summarized in Table 2 and compared against the true answer. It can be seen that the approach successfully recovered the right answer, albeit this was a particularly simple function. More complicated examples are treated in the next chapters. Furthermore, recall that this answer does not take into account the multi-disciplinary nature of aircraft design problems and, therefore, most design variables were driven to the extreme of the design space. For example, without account for aerodynamics, minimizing wing weight tends towards the smallest allowable aspect ratio, as expected. Again, the main purpose of this example was simply to illustrate the method using a realistic engineering function that
has been used by others, so that the reader may understand how the method is applied. The application of the method in the context of multi-disciplinary optimization is deferred to later discussion.
The purpose of this chapter is to assess the performance of the method when it is applied to practical aerodynamic problems. In particular, two problems will be considered: a 2D staggered airfoil problem, reminiscent of OWN integration, and a 3D OWN engine integration problem. In a first step, the 2D problem will be used to compare LSGT against established screening methods in terms of dimensionality reduction and, in a second step, it will be used to verify that LSGT is capable of recovering a known local optimum. Finally, the 3D problem will be used to assess whether or not LSGT is cost-effective for realistic applications. This will be achieved by quantifying the maximum amount of dimensionality reduction possible and the cost of achieving it, while retaining good predictive accuracy. This chapter is broken up into two sections: one for the 2D problem and the other for the 3D problem.

5.1 Experiment 2: Staggered Airfoil Problem

This research is motivated by OWN concepts. However, for purposes of testing, it is often informative to work with a simplified problem that is less expensive to compute, but which retains the salient characteristics of the former. A 2D staggered airfoil optimization problem was thus selected because it shared similarities with OWN, as shown in Fig. 41, where the baseline airfoil is the ONERA D [75] and the vertical separating distance is 0.3 chord length. All computations were performed using CART3D, an Euler solver developed by NASA, previously described in Section 2.2.2. It can be seen that both problems present nonlinearities with the same type of trends and, therefore, a staggered airfoil problem offers adequate inference about how the method handles transonic flow behavior. This experiment will be used to answer the
Question 4. How does LSGT compare against established screening methods?

Question 5. Is LSGT capable of recovering a known minimum?

5.1.1 Problem Setup

The computational framework used to implement LSGT is shown in Fig. 42. It is similar to Fig. 9, except that the MATLAB portion of the environment was modified to implement LSGT. Everything else is identical to the computational setup described in Section 2.2.2 and the reader is invited to review it for more detail. The design space contains 42 design variables including the angle of attack, which are described
Table 3: Staggered airfoil design space

<table>
<thead>
<tr>
<th>$x$</th>
<th>Symbol</th>
<th>Description</th>
<th>$x_L$</th>
<th>$x_U$</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>$\alpha$</td>
<td>Angle of attack</td>
<td>-1.0</td>
<td>3.0</td>
<td>deg</td>
</tr>
<tr>
<td>$x_2$</td>
<td>$A_{U0}^T$</td>
<td>Top airfoil upper surface CST coefficient no. 0</td>
<td>0.1694</td>
<td>0.3146</td>
<td>—</td>
</tr>
<tr>
<td>$x_3$</td>
<td>$A_{U1}^T$</td>
<td>Top airfoil upper surface CST coefficient no. 1</td>
<td>0.0728</td>
<td>0.1352</td>
<td>—</td>
</tr>
<tr>
<td>$x_4$</td>
<td>$A_{U2}^T$</td>
<td>Top airfoil upper surface CST coefficient no. 2</td>
<td>0.3718</td>
<td>0.2860</td>
<td>—</td>
</tr>
<tr>
<td>$x_5$</td>
<td>$A_{U3}^T$</td>
<td>Top airfoil upper surface CST coefficient no. 3</td>
<td>0.0434</td>
<td>0.0334</td>
<td>—</td>
</tr>
<tr>
<td>$x_6$</td>
<td>$A_{U4}^T$</td>
<td>Top airfoil upper surface CST coefficient no. 4</td>
<td>0.3770</td>
<td>0.2900</td>
<td>—</td>
</tr>
<tr>
<td>$x_7$</td>
<td>$A_{U5}^T$</td>
<td>Top airfoil upper surface CST coefficient no. 5</td>
<td>0.2340</td>
<td>0.1800</td>
<td>—</td>
</tr>
<tr>
<td>$x_8$</td>
<td>$A_{U6}^T$</td>
<td>Top airfoil upper surface CST coefficient no. 6</td>
<td>0.1053</td>
<td>0.0810</td>
<td>—</td>
</tr>
<tr>
<td>$x_9$</td>
<td>$A_{U7}^T$</td>
<td>Top airfoil upper surface CST coefficient no. 7</td>
<td>0.4511</td>
<td>0.3470</td>
<td>—</td>
</tr>
<tr>
<td>$x_{10}$</td>
<td>$A_{U8}^T$</td>
<td>Top airfoil upper surface CST coefficient no. 8</td>
<td>0.0868</td>
<td>0.0668</td>
<td>—</td>
</tr>
<tr>
<td>$x_{11}$</td>
<td>$A_{U9}^T$</td>
<td>Top airfoil upper surface CST coefficient no. 9</td>
<td>0.2418</td>
<td>0.1860</td>
<td>—</td>
</tr>
<tr>
<td>$x_{12}$</td>
<td>$A_{L0}^T$</td>
<td>Top airfoil lower surface CST coefficient no. 0</td>
<td>-0.1687</td>
<td>-0.2410</td>
<td>—</td>
</tr>
<tr>
<td>$x_{13}$</td>
<td>$A_{L1}^T$</td>
<td>Top airfoil lower surface CST coefficient no. 1</td>
<td>-0.0457</td>
<td>-0.0653</td>
<td>—</td>
</tr>
<tr>
<td>$x_{14}$</td>
<td>$A_{L2}^T$</td>
<td>Top airfoil lower surface CST coefficient no. 2</td>
<td>-0.2919</td>
<td>-0.4170</td>
<td>—</td>
</tr>
<tr>
<td>$x_{15}$</td>
<td>$A_{L3}^T$</td>
<td>Top airfoil lower surface CST coefficient no. 3</td>
<td>0.1625</td>
<td>0.1250</td>
<td>—</td>
</tr>
<tr>
<td>$x_{16}$</td>
<td>$A_{L4}^T$</td>
<td>Top airfoil lower surface CST coefficient no. 4</td>
<td>-0.2961</td>
<td>-0.4230</td>
<td>—</td>
</tr>
<tr>
<td>$x_{17}$</td>
<td>$A_{L5}^T$</td>
<td>Top airfoil lower surface CST coefficient no. 5</td>
<td>-0.1190</td>
<td>-0.1700</td>
<td>—</td>
</tr>
<tr>
<td>$x_{18}$</td>
<td>$A_{L6}^T$</td>
<td>Top airfoil lower surface CST coefficient no. 6</td>
<td>0.0608</td>
<td>0.0468</td>
<td>—</td>
</tr>
<tr>
<td>$x_{19}$</td>
<td>$A_{L7}^T$</td>
<td>Top airfoil lower surface CST coefficient no. 7</td>
<td>-0.3794</td>
<td>-0.5420</td>
<td>—</td>
</tr>
<tr>
<td>$x_{20}$</td>
<td>$A_{L8}^T$</td>
<td>Top airfoil lower surface CST coefficient no. 8</td>
<td>0.1188</td>
<td>0.0914</td>
<td>—</td>
</tr>
<tr>
<td>$x_{21}$</td>
<td>$A_{L9}^T$</td>
<td>Top airfoil lower surface CST coefficient no. 9</td>
<td>-0.0798</td>
<td>-0.1140</td>
<td>—</td>
</tr>
<tr>
<td>$x_{22}$</td>
<td>$x$</td>
<td>Airfoil LE-to-LE stagger position</td>
<td>-1.0</td>
<td>1.0</td>
<td>m</td>
</tr>
<tr>
<td>$x_{23}$</td>
<td>$A_{L0}^B$</td>
<td>Bottom airfoil upper surface CST coefficient no. 0</td>
<td>0.1694</td>
<td>0.3146</td>
<td>—</td>
</tr>
<tr>
<td>$x_{24}$</td>
<td>$A_{L1}^B$</td>
<td>Bottom airfoil upper surface CST coefficient no. 1</td>
<td>0.0728</td>
<td>0.1352</td>
<td>—</td>
</tr>
<tr>
<td>$x_{25}$</td>
<td>$A_{L2}^B$</td>
<td>Bottom airfoil upper surface CST coefficient no. 2</td>
<td>0.3718</td>
<td>0.2860</td>
<td>—</td>
</tr>
<tr>
<td>$x_{26}$</td>
<td>$A_{L3}^B$</td>
<td>Bottom airfoil upper surface CST coefficient no. 3</td>
<td>0.0434</td>
<td>0.0334</td>
<td>—</td>
</tr>
<tr>
<td>$x_{27}$</td>
<td>$A_{L4}^B$</td>
<td>Bottom airfoil upper surface CST coefficient no. 4</td>
<td>0.3770</td>
<td>0.2900</td>
<td>—</td>
</tr>
<tr>
<td>$x_{28}$</td>
<td>$A_{L5}^B$</td>
<td>Bottom airfoil upper surface CST coefficient no. 5</td>
<td>0.2340</td>
<td>0.1800</td>
<td>—</td>
</tr>
<tr>
<td>$x_{29}$</td>
<td>$A_{L6}^B$</td>
<td>Bottom airfoil upper surface CST coefficient no. 6</td>
<td>0.1053</td>
<td>0.0810</td>
<td>—</td>
</tr>
<tr>
<td>$x_{30}$</td>
<td>$A_{L7}^B$</td>
<td>Bottom airfoil upper surface CST coefficient no. 7</td>
<td>0.4511</td>
<td>0.3470</td>
<td>—</td>
</tr>
<tr>
<td>$x_{31}$</td>
<td>$A_{L8}^B$</td>
<td>Bottom airfoil upper surface CST coefficient no. 8</td>
<td>0.0868</td>
<td>0.0668</td>
<td>—</td>
</tr>
<tr>
<td>$x_{32}$</td>
<td>$A_{L9}^B$</td>
<td>Bottom airfoil upper surface CST coefficient no. 9</td>
<td>0.2418</td>
<td>0.1860</td>
<td>—</td>
</tr>
<tr>
<td>$x_{33}$</td>
<td>$A_{L0}^L$</td>
<td>Bottom airfoil lower surface CST coefficient no. 0</td>
<td>-0.1687</td>
<td>-0.2410</td>
<td>—</td>
</tr>
<tr>
<td>$x_{34}$</td>
<td>$A_{L1}^L$</td>
<td>Bottom airfoil lower surface CST coefficient no. 1</td>
<td>-0.0457</td>
<td>-0.0653</td>
<td>—</td>
</tr>
<tr>
<td>$x_{35}$</td>
<td>$A_{L2}^L$</td>
<td>Bottom airfoil lower surface CST coefficient no. 2</td>
<td>-0.2919</td>
<td>-0.4170</td>
<td>—</td>
</tr>
<tr>
<td>$x_{36}$</td>
<td>$A_{L3}^L$</td>
<td>Bottom airfoil lower surface CST coefficient no. 3</td>
<td>0.1625</td>
<td>0.1250</td>
<td>—</td>
</tr>
<tr>
<td>$x_{37}$</td>
<td>$A_{L4}^L$</td>
<td>Bottom airfoil lower surface CST coefficient no. 4</td>
<td>-0.2961</td>
<td>-0.4230</td>
<td>—</td>
</tr>
<tr>
<td>$x_{38}$</td>
<td>$A_{L5}^L$</td>
<td>Bottom airfoil lower surface CST coefficient no. 5</td>
<td>-0.1190</td>
<td>-0.1700</td>
<td>—</td>
</tr>
<tr>
<td>$x_{39}$</td>
<td>$A_{L6}^L$</td>
<td>Bottom airfoil lower surface CST coefficient no. 6</td>
<td>0.0608</td>
<td>0.0468</td>
<td>—</td>
</tr>
<tr>
<td>$x_{40}$</td>
<td>$A_{L7}^L$</td>
<td>Bottom airfoil lower surface CST coefficient no. 7</td>
<td>-0.3794</td>
<td>-0.5420</td>
<td>—</td>
</tr>
<tr>
<td>$x_{41}$</td>
<td>$A_{L8}^L$</td>
<td>Bottom airfoil lower surface CST coefficient no. 8</td>
<td>0.1188</td>
<td>0.0914</td>
<td>—</td>
</tr>
<tr>
<td>$x_{42}$</td>
<td>$A_{L9}^L$</td>
<td>Bottom airfoil lower surface CST coefficient no. 9</td>
<td>-0.0798</td>
<td>-0.1140</td>
<td>—</td>
</tr>
</tbody>
</table>

82
in Table 3. The top airfoil is allowed to translate horizontally above the bottom airfoil at fixed vertical distance, while all airfoil coordinates are parameterized using the CST method [37], which was previously described in Section 2.2.2. The ranges of the design space are shown in Table 3, where CST coefficients were allowed to vary ±30% of the baseline value. This ensured a minimum thickness of 9% with a round leading edge and a manufacturable trailing edge. Some example airfoil shapes are shown in Fig. 43. The objective function is taken similar to experiment 1, except that the Mach number is \( M = 0.8 \):

\[
    f(x) = 200 \cdot C_D(x) + 650 \cdot (C_L(x) - 0.525)^2
\]

Finally, a mesh refinement study is shown in Fig. 44 so that the reader may verify that the grid was sufficiently refined. The extents of the flow domain were taken ±40
Figure 43: Example of CST shapes over selected design space

chord lengths away from the body, to which free stream boundary conditions were applied. Grid refinement was accomplished by uniformly refining the grid in a small, rectangular region enclosing both airfoils. The selected grid resolution was within 5 drag counts of the finest grid tested, with a difference of 0.01 for the lift coefficient.

5.1.2 Dimensionality Reduction: LSGT vs. Screening Methods

The purpose of this subsection is to compare LSGT against established screening methods in terms of dimensionality reduction. In particular, LSGT will be compared to Main Effects Screening (MES), Elementary Effects Screening (EES) and screening using partial derivatives. A description of MES and EES can be found in Appendix C. At this point, it is assumed that the reader is well familiar with the steps of LSGT,
(a) Selected grid resolution (98,849 cells)

(b) Force coefficients as a function of cell count (solid markers = selected)

**Figure 44:** Staggered airfoil grid sensitivity study at $M = 0.8$ and $C_L = 0.525$
Figure 45: Convergence history of $\|S_N - \Sigma\|_\infty$

such that results can simply be stated. If this is not the case, the reader should review Chapter 4 for an example.

**LSGT**

The convergence history of the sample covariance matrix is depicted in Fig. 45. It can be seen that it took on the order of $O(150)$ samples to reach asymptotic convergence, which is an excellent number for a 42-variable problem. The Pareto plots obtained after applying PCA are shown in Fig. 46, where it can be seen that most of the variation is concentrated in the first principal component, referred to as PC1. Furthermore, the magnitude of the $y$-axis in Fig. 51(a) compared to Fig. 51(b) shows that the standard deviation of the gradient is relatively large along PC1, which indicates strong nonlinear behavior in that direction. In fact, breaking up the first two principal components in terms of their coefficients in Fig. 47, one finds that PC1 is primarily driven by $x_{22} = x/c$, which should not be surprising given Fig. 41. The angle of attack $\alpha$ also contributes to PC1, but not as much as $x/c$. On the other hand, $\alpha$ is responsible for most of the variation along PC2, although other variables also contribute significantly to it. These observations simply serve as a quick sanity check.
Figure 46: Dimensionality reduction with PCA (truncated beyond 19 PC)
Figure 47: Coefficient values for first and second principal components

since one would expected $x/c$ and $\alpha$ to actively contribute to the first few principal components, which account for most of the variation.

However, in order to obtain a model that has good prediction error, Fig. 48 shows that at least $q = 20$ principal components must be retained in the basis, which represents 52% dimensionality reduction. This ensures that the predicted force coefficients will have an error of 5% on average and 10% at the most, relative to the true value, with 95% confidence. These results were obtained by taking 100 random validation points, evaluating them using the true model, re-evaluating using the approximate model due to PCA for each different value of $q$ and, finally, computing the error. While 20 variables is a significant improvement over 42, it is unfortunately not less than 10, which is usually considered the cutoff point for design space exploration.
Main Effects Screening

The theory behind MES can be found in Appendix C. Results are summarized in the main effects plot of Fig. 49 using a resolution IV Fractional Factorial Design (FFD IV) containing 128 runs. FFD IV are well established, two-level screening designs which minimize the number of runs required for a given number of design variables, while preventing aliasing between main effects and two-factor interactions. The 42 main effects $x_1, \ldots, x_{42}$ are shown on the left $y$-axis, listed in order of decreasing importance.
**Figure 49:** Main effects plot of $f(z) = 200C_d(z) + 650(C_l(z) - 0.525)^2$
and their associated contrasts are shown on the x-axis. It is convention to display the
importance of main effects using contrasts [57], which correspond to twice the value
of the regression coefficients of a linear model fit through the data. Finally, the right
y-axis shows the p-value associated with each main effect, calculated using analysis
of variance [54]. Main effects are statistically significant with 95% confidence if their
p-value is less than 0.05. This limit is shown by the bold, horizontal line in Fig. 49,
below which design variables can be discarded.

Fig. 49(a) shows results over the full range of the design space. There are thus
8 significant factors: \(x_{30}, x_9, x_{28}, x_4, x_1, x_{16}, x_7, x_{22}\). However, notice that \(x_{22} = x/c\)
barely made the cut, even though Fig. 41 showed that drag is a strong nonlinear
function of \(x/c\). This is not surprising because MES cannot capture nonlinear effects
since it operates by sampling the extremes of the design space and fitting a linear
model through the data. Based on the magnitude of the slope in any given direction,
it then makes an assessment as to whether or not the variable associated with that
direction is important. Therefore, since drag is similar at either end of the range
for \(x/c\) in Fig. 41, the slope would have been small. Of course, this issue can be
resolved by halving the bounds of the design space and repeating the process, which
is the purpose of Fig. 49(b). It can be seen that \(x_{22}\) and the angle of attack \(x_1\) are
now identified as the most important variables, as one might expect. The union of
significant main effects in Figs. 49(a) and 49(b) yields a total of 27 significant factors,
which represents 35% dimensionality reduction; less than LSGT. Furthermore, the
total cost of repeating MES twice is \(2 \times 128 = 256\), which is on par with LSGT.

**Elementary Effects Screening**

The theory behind Elementary Effects Screening (EES) is provided in Appendix C
and the reader is encouraged to consult it, as some of the terminology in this section
is taken from the appendix. Results are shown in Fig. 50 using \(k = 4\) and \(C = 1\) in
Eq. 86, which are the same settings used in the original paper by Morris [55]. There were a total of 40 random samples of $F$, which amounted to $40 \times 42 = 1680$ individual elementary effects samples. The convergence history of the sample covariance matrix is shown as a function of the number of elementary effect samples in Fig. 50(c), where the formula used is the same one as in Eq. 19. Notice that even after 1680 samples, asymptotic convergence has not yet been reached. The corresponding mean and variance plot is shown in Figs. 50(a) and 50(b), where the numbers correspond to the design variables listed in Table 3. It can be seen that EES correctly identified the angle of attack $x_1 = \alpha$ and the stagger distance $x_{22} = x/c$ to be significant factors. However, by comparison to Fig. 46, it can be seen that it takes a much larger number of variables to explain the same amount of variation using EES than it does using LSGT. For example, in order to explain 80% of the total variation, it takes at a minimum 19 variables using EES whereas it only takes 2 using LSGT. Overall, LSGT was $1680/150 = 11.2$ times more efficient than EES for this problem and $19/2 = 9.5$ more effective.

_Screening Using Partial Derivatives_

An elementary effect is an approximate measure of sensitivity. It is constructed in the same way as a first order finite difference, except that the finite step is not small. A partial derivative, by contrast, is an exact measure of sensitivity and, therefore, is much more precise. It could thus be used in the same way as an elementary effect is used for screening, with the added benefit that an adjoint solver could be used to improve efficiency drastically. Said otherwise, one can randomly sample the gradient and use the data to estimate the distribution of each partial derivative. Then, based on the mean and standard deviation, one can make an assessment as to whether or not a given variable has an important effect on the response. Variables associated with partial derivative distributions that have either a large mean or large standard
Figure 50: EES results
deviation (or both) are deemed significant; others can be discarded. This is the same idea as LSGT, except that PCA is not performed. The results of such a process are presented in Fig. 51 using the same data that was used for LSGT. Compared to Fig. 46, it can be seen that without PCA it takes more degrees of freedom to explain the same amount of variation. For example, 19 variables only explain \( \approx 87\% \) of the total variation in the mean and standard deviation of the gradient in Fig. 51, whereas it explains \( \approx 98\% \) in Fig. 46. This result simply emphasizes the benefit of PCA compared to a simple screening test. The reason for this comes back to the discussion about unsupervised vs. supervised learning methods for aerodynamic design problems in Section 3.1.2. The reader should therefore refer to that section for more detail.

5.1.3 Optimization: Comparison With and Without LSGT

The purpose of this subsection is to assess how well LSGT is capable of recovering a known local minimum by comparing results with and without it. They are shown in Figs. 52 and 53 starting from two different initial guesses, where it can be seen that in both cases the final value of the objective function is close, as one would expect. Moreover, flow patterns show good agreement between the actual and predicted solutions, but the actual and predicted geometries are not exactly identical; yet this is expected. Recall that PCA is based on the notion that a small loss of information is acceptable in exchange for large dimensionality reduction, provided most variation in the gradient (and the response) is captured. A small loss of gradient information implies a solution near the true optimum, but not necessarily on top of it. As a consequence, one should not expect to recover exactly the same geometry that would be obtained using gradient-based optimization without LSGT; albeit it should be close. Such small loss of accuracy is likely to be acceptable at the conceptual design level, where designers willingly trade small penalties for large dimensionality reduction,
Figure 51: Dimensionality reduction with PCA (truncated beyond 19 variables)
which enables a larger portion of the design to be explored for the same cost.

However, despite these encouraging results, comparison of convergence histories in Figs. 52(d) and 53(d) suggests LSGT offers no advantage for gradient-based optimization. It will yield the right answer, but convergence could be slower compared to using adjoint methods directly without dimensionality reduction. This is surprising because one would expect dimensionality reduction to enhance convergence. This unexpected result is most likely explained by the fact that the benefit of reducing dimensionality is offset by the addition of $2 \times (p - q)$ nonlinear constraints in Eqs. 46–47. The cost involved in setting up the method is therefore not worth the effort if gradient-based optimization is the only goal. On the other hand, it is worth it if there is a need to explore the design space using gradient-free methods, which would be impractical without dimensionality reduction.

5.2 Experiment 3: OWN Problem

The usefulness of LSGT is entirely dependent on the ability to reduce dimensionality sufficiently enough for a realistic problem. As a consequence, there remains one important question:

**Question 6.** Is the method cost-effective for a realistic problem? i.e. How many gradient samples are necessary to reduce dimensionality and is dimensionality reduced sufficiently for exploration?

The answer to this question requires counting the number of gradient samples necessary to reduce dimensionality and then computing prediction accuracy as a function of the reduced number of design variables. This will be accomplished using the OWN engine integration problem of Chapter 2 as a representative example.
Figure 52: Comparison of optimization with and without LSGT

(a) True optimum, $f(z) = 0.29$

(b) Predicted optimum, $f(\xi_{q=20}) = 0.37$

(c) Initial Guess

(d) Optimizer convergence history
Figure 53: Comparison of optimization with and without LSGT
5.2.1 Problem Description

The optimization problem setup is the same one described in Section 2.2.2. There are a total of 145 design variables, previously shown in Fig. 8 and, as before, the objective function is given by:

\[ f(x) = 200 \cdot C_D(x) + 650 \cdot (C_L(x) - 0.525)^2 \]  

(71)

Once again, the goal is to minimize drag at Mach 0.785 subject to a fixed lift coefficient of \( C_L = 0.525 \). The typical value used for scaling was unity, except for the CST coefficients, which was taken to be \( 10^{-1} \). The computational framework is the same one shown in Fig. 42 and everything else is identical to the setup in Section 2.2.2, including the description of the CFD solver; the reader is invited to review that section for more detail.

5.2.2 Cost-Effectiveness of LSGT on a Real Application

The efficiency of LSGT depends on the number of gradient samples required to construct a good estimator of the covariance matrix in step 2. In order to determine this number, it suffices to track the convergence history of Eq. 20. The results are shown in Fig. 54, where it can be seen that 150 gradient samples were sufficient to
Principal components

\( \lambda \)

0%
12%
23%
35%
46%
58%
69%
81%
93%

(a) Pareto plot for std. dev. of \( \partial f / \partial \xi_j \)

(b) Pareto plot for mean of \( \partial f / \partial \xi_j \)

Figure 55: Dimensionality reduction with PCA

obtain a good estimate, which is remarkably efficient for such a large problem. The corresponding results for dimensionality reduction using PCA are shown in Table 4 and Fig. 55, where it was found that 79 variables were sufficient to capture nearly all the variation. This is a little over half the starting number of design variables. However, further improvement is possible. Consider Fig. 56, which shows the variation of force coefficient prediction error as a function of \( q \). Each data point was obtained by taking 100 random samples in \( p \)-dimensional space, evaluating them using the true model \( f(z) \), then again with the approximate model \( f(\xi_q) \) and, finally, calculating the prediction error. Upon examination, it can be seen that it only takes 17 degrees of freedom (down from 145) to obtain a model with a maximum prediction error of \( \pm 7\% \) with 95\% confidence for drag and even better for lift. Such a model explains 90\% of the total variation according to Fig. 55. Exploring the design space therefore just went from impractical (i.e. 145 variables) to conceivable (i.e. 17 variables). Presuming that this test problem is representative of a realistic design problem, LSGT can thus be deemed cost-effective for practical applications.

Finally, in order to emphasize the advantage of PCA compared to screening, consider Fig. 57, which shows the results of a screening process based on partial derivative
Table 4: DR with PCA

<table>
<thead>
<tr>
<th>$\psi(t)$</th>
<th>$\psi(s)$</th>
<th>$q$</th>
<th>Principal Components</th>
<th>DR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.70</td>
<td>0.00</td>
<td>10</td>
<td>1–6</td>
<td>99%</td>
</tr>
<tr>
<td>0.84</td>
<td>0.84</td>
<td>17</td>
<td>1–12, 15–17, 22, 31</td>
<td>89%</td>
</tr>
<tr>
<td>0.90</td>
<td>0.90</td>
<td>29</td>
<td>1–20, 22, 25, 27, 28, 30–32, 36, 42</td>
<td>80%</td>
</tr>
<tr>
<td>0.99</td>
<td>0.99</td>
<td>79</td>
<td>1–64, 66–70, 74, 75, 78, 80, 83–85, 87, 95, 103</td>
<td>46%</td>
</tr>
<tr>
<td>1.00</td>
<td>1.00</td>
<td>145</td>
<td>1–145</td>
<td>0%</td>
</tr>
</tbody>
</table>

(a) Drag coefficient $R^2$

(b) Drag coefficient prediction error

(c) Lift coefficient $R^2$

(d) Lift coefficient prediction error

Figure 56: Model quality of $f(\xi_q)$ as a function of $q$ (100 validation points)
distributions, as described in Section 5.1.2. The only difference between Fig. 55 and Fig. 57 is the coordinate transformation due to PCA; everything else is otherwise the same. By comparing both figures, it can be seen that PCA reduces dimensionality much more effectively than screening for this problem. For instance, in order to explain the same percentage of variation, it takes many times less variables using PCA than using screening. As explained in Section 3.1.2, this is because there is redundancy in the design space (i.e. more than one variable affects the same thing), which is likely to be true for most aerodynamic shape optimization problems.

5.3 Synthesis

This chapter tested LSGT on two different problems: a 2D airfoil problem and a more realistic 3D OWN integration problem. In general, it was found that LSGT outperforms typical screening methods often used in aerospace engineering and is capable of successfully recovering a known optimum. However, when tested on the 2D problem, it was found that LSGT does not improve convergence for gradient-based optimization, even though dimensionality is reduced. It was explained that this is most likely due to the additional $p - q$ inequality constraints necessary to solve
the problem in reduced latent space. LSGT is thus only useful is there is a need to reduce dimensionality for purposes of enabling gradient-free methods. Otherwise, LSGT is not recommended. Finally, the method was found to be extremely cost-effective when applied to a realistic problem. It successfully reduced dimensionality from 145 variables to just 17 at a cost of only 150 gradient samples, while ensuring a model that was accurate within ±7% for drag and even better for lift. The results presented in this chapter therefore support Hypothesis 3 and suggest that, indeed, applying PCA to the gradient distribution is an effective way to reduce dimensionality for large, aerodynamic problems. However, it makes no guarantee that dimensionality will be reduced below ten degrees of freedom, which is usually the cut-off point for design space exploration. For example, in the 3D problem, it was found that at least 17 variables are needed to retain good predictive accuracy. This is a significant improvement over 145, but admittedly still a large number for exploration. Additional research is therefore necessary to learn how to explore moderately dimensional design space with 10 to 20 variables. Several ideas are presented in the next chapter.
This chapter summarizes key findings and discusses opportunities for further research. However, beforehand, a summary of all research questions and hypotheses is provided.

6.1 Summary of Research Questions & Hypotheses

The purpose of this section is to compile a list of all research questions that guided this thesis, in order to remind the reader how they were addressed, what hypotheses were made and what experiments were conducted. The main research question driving this thesis was the following:

Question. 1 Given that a large number of design variables is usually required for aerodynamic design, how is it possible to overcome the curse of dimensionality in order to facilitate the use of gradient-free methods for large CFD problems without defaulting important degrees of freedom?

It was developed in Chapter 1 after observing that no method currently existed to enable the use of gradient-free methods for exploring large CFD problems. Aerodynamic design problems are high-dimensional because useful parameterization of aerodynamic shapes tend to require a large number of design variables. The answer to this question culminated into LSGT, the featured method developed for this research, which applies principal component analysis to gradient information in order to reduce dimensionality under the following hypothesis:
Hypothesis. 3 The curse of dimensionality can be overcome cost-effectively by applying principal component analysis to the gradient of the objective function in order to reduce dimensionality. This yields a low-rank transformation that maps a high-dimensional space onto an equivalent, low-dimensional space in which the use gradient-free methods becomes practical.

In order to justify the need for the approach, a brief literature review was conducted in Chapter 2 to understand how industry currently addresses the problem of high-dimensionality for engine integration. It was discovered that this problem is avoided by defaulting the OML to some pre-selected, baseline shape during design space exploration, in favor of concentrating on a handful of design variables such as nacelle location, engine type or Mach number. This led to another research question:

Question. 2 What is the consequence of defaulting OML shape variables during engine placement selection?

Experiment 1 was conducted in order to answer this question, which compared OWN drag coefficients as a function of nacelle displacement before and after optimizing the OML. It was found that accounting for the OML changes the trend significantly and opens the design space, allowing unforeseen solutions to be discovered. As a consequence, the following hypothesis was made:

Hypothesis. 2 Defaulting the OML introduces large uncertainty in the design process, due to the unknown effect of changing the OML at candidate engine locations. As a consequence, there is a risk that a satisfactory solution might not
be found because the design space was too restrictive at the time it was being explored. A new approach is therefore needed to explore large CFD problem without defaulting important degrees of freedom.

However, Experiment 1 employed an Euler analysis in order to reach this conclusion. Concerns were that Euler methods are often considered inadequate because they do not capture boundary layer displacement effects or shock-induced boundary layer separation. It was therefore necessary to verify that inviscid flow assumptions were able to capture trends and interactions correctly:

**Question. 3** Does an Euler analysis with flow-through nacelles capture trends and interactions correctly?

This was accomplished by comparing results to a similar OWN study conducted by industry in Chapter 2. Trends and important physical behavior were found to be in good agreement and, therefore, the following hypothesis was made:

**Hypothesis. 1** An Euler analysis captures general design trends and interactions correctly and, therefore, it is reasonable for comparative analysis.

A new method called LSGT was then developed in Chapter 3, as a solution to Research Question 1, and a step-by-step example was provided in Chapter 4. Finally, the method was put to the test in Chapter 5 in order to answer the following questions:
Research Questions 4 and 5 were answered using the results of Experiment 2, which consisted of applying LSGT to a 2D staggered airfoil optimization problem and comparing results to established screening methods in aerospace engineering. It was found that LSGT outperforms typical screening methods and is capable of recovering a known local minimum with good predictive accuracy. Finally, Research Question 6 was answered by applying LSGT to a 3D OWN problem, representative of a more realistic level of difficulty. It was found that the method is cost-effective, successfully reducing dimensionality from 145 variables to just 17 variables at a cost of only 150 gradient samples, while ensuring a model that had a prediction error within $\pm 7\%$ for drag and even better for lift with 95% confidence.

### 6.2 Summary of Findings

Historically, OWN concepts are notorious for generating large amounts of drag at high speeds compared to conventional UWN concepts. However, it was found that through proper shape optimization, OWN concepts can be made aerodynamically competitive with conventional UWN engine installations; a result that is corroborated by similar
findings by industry. In particular, within the fidelity of an Euler analysis, two zones of feasible engine placements were discovered. The first occurs when the nacelle inlet is located ahead of the wing leading edge, with the nozzle exit plane possibly overlapping the leading edge up to the 40% chord. The second occurs when it is located near the wing trailing edge, with the inlet plane possibly overlapping the trailing edge up to the 85% chord. This second location is special because it enjoys a beneficial suction peak on the lower lip of the nacelle, which serves to counter-act wave drag. In addition, the presence of the nacelle at the trailing edge helps slow down the flow ahead, making it easier to eliminate the local shock wave on the wing upon shape optimization. Overall, these findings offer supporting evidence that OWN civil transports can be made aerodynamically efficient. Furthermore, it was shown that an Euler analysis captures transonic OWN trends correctly and, therefore, it is expected that similar conclusions would be reached using higher order methods, albeit with improved predictive accuracy.

However, in order to achieve aerodynamic efficiency, a large design space consisting of 145 design variables had to be considered, which is typical of useful parameterizations of aerodynamic shapes. In order to alleviate the cost associated with exploring so many variables, a brief literature review revealed that the standard practice is to freeze the OML during exploratory phases, in favor of concentrating on a handful of preferred variables taken to be: nacelle location, Mach number and engine-airframe combination. As a consequence, it was shown that there is a large uncertainty in drag predictions, due to the unknown effect of changing the OML at any given nacelle location. It was further explained that such uncertainty could propagate to other disciplines and cause the airplane to be over-designed by masking feasible regions of the design space. The easiest way to solve this problem is to re-optimize the geometry at every candidate engine location. However, depending on the problem and the number of candidate designs to be tested, the cost of repeated optimization is usually
excessive. As a consequence, in practice, the variation in drag due to the OML is usually accounted for by relying on experts to subjectively assess *expected improvement* at a given nacelle location. Rather than re-optimizing the OML every time, candidate engine placements are down-selected based on a qualitative assessment of their potential for further improvement upon optimizing the OML.

It was therefore concluded that there did not exist an approach capable of accounting for all necessary variables simultaneously during early exploratory design phases, where gradient-free methods cannot be avoided. Design space exploration is a complementary step that is useful to characterize trends within the context of multi-disciplinary trade studies, understand what drives the response and identify designs of interest worth optimizing further. However, in order to take advantage of it, a large number of samples is usually required and the computational burden quickly becomes intractable beyond 10 variables. Therefore, based on these deliberations, a new method called Latent Space Gradient Transformation (LSGT) was developed in order to facilitate the use of gradient-free methods to explore large, computationally expensive design problems. This was accomplished by using an innovative application of PCA, where the latter is applied to the gradient of the objective function in order to reduce dimensionality; something that had not been done before. This yields a low-rank transformation that maps a high-dimensional design space onto an equivalent, low-dimensional space in which gradient-free methods become more affordable.

The method was successfully tested on a 2D staggered airfoil problem and a 3D OWN integration problem. Results showed that LSGT outperforms conventional screening methods typically used in aerospace engineering and is capable of successfully recovering a known optimum with good accuracy. Moreover, LSGT successfully reduced dimensionality from 145 variables to just 17 at a cost of only 150 gradient samples for the 3D problem, while ensuring a model that had a prediction error within ±7% for drag and even better for lift with 95% confidence. The method can thus be
deemed cost-effective for realistic, aerodynamic design problems, which verifies the main hypothesis of this research that applying PCA to gradient information is an cost-effective approach for reducing dimensionality in large CFD problems.

6.3 Remaining Gaps & Opportunities for New Research

Despite these encouraging results, the method offers no guarantee that dimensionality will be reduced to less than ten variables, which is usually the cutoff point for gradient-free methods. For example, when applied to the OWN integration problem, it was found that good prediction accuracy required at least 17 variables. While this might be manageable using a large HPC cluster, it is admittedly still a large number. The problem therefore went from impractical to practical, but expensive. The usefulness of the method is therefore entirely dependent on whether or not computational resources can handle the intrinsic dimensionality of the problem, which is problem-dependent. The method presented in this thesis is thus a step forward in the right direction and opens the door to new research opportunities.

In the author’s opinion, this hurdle could be overcome in one of three ways. The first alternative would be to employ high-dimensional statistical regression methods such as the LASSO [81, 9], which can handle a larger design space using a relatively small number of training data. The second alternative would be to employ gradient-enhanced regression methods [84, 15], in order to take advantage of the vast amounts of gradient information that are available at no additional cost in step 2. However, in order to be successful, both these approaches will require the ability to create a sampling plan that exhibits good space filling properties beyond 10 variables, subject to nonlinear bound constraints, which is not trivial. As a third alternative, one could adopt a set-based design philosophy, where portions of the design space are progressively eliminated as the number of principal components is gradually increased. For instance, even though fewer than ten principal components might be too inaccurate
for optimization, it might be sufficient to eliminate large regions of the design space with high confidence. The regions that remain would thus represent clusters of points containing local minima. PCA could then be refined within each cluster of points and the process repeated iteratively to desired accuracy.

Finally, notice that in the current formulation of LSGT equality and inequality constraints are dealt with by means of penalty functions. However, in certain instances, it may be preferable to handle them directly. For example, there will always exist situations where constraints have to be met precisely or, in the context of MDAO, have to be passed on to another discipline along with their gradient. As a consequence, it would be worth the effort to update the formulation and find a way to incorporate LSGT into an MDAO framework.
APPENDIX A

REVIEW OF ADJOINT DESIGN METHODS

The purpose of this chapter is to introduce the reader to the general idea behind adjoint methods, so that he or she may understand where their computational efficiency comes from. The derivation to follow is taken directly from Giles and Pierce [20] and the interested reader is referred to their work for more details.

A.1 Nomenclature

\( J \)  Aerodynamic objective function
\( R \)  Governing flow equations
\( \mathcal{U} \)  Set of flow variables at discrete grid point (i.e. pressure, density and velocity)
\( x \)  Design variables that control the shape of the geometry
\( \nu \)  Adjoint solution at discrete grid points

A.2 Direct Formulation

In aerodynamic optimization, the goal is to minimize the objective function \( J(x, \mathcal{U}) \), which is a function of the design variables and the flow variables, subject to the constraint that the governing equations must be satisfied at all grid points:

\[
\begin{align*}
\text{Find: } & \quad x^* = \arg \min_x J(x, \mathcal{U}) \\
\text{Subject to: } & \quad R(\mathcal{U}, x) = 0
\end{align*}
\]  \tag{72}

Using the chain rule, the gradient of the objective function can be expressed as:

\[
\frac{dJ}{dx} = \frac{\partial J}{\partial \mathcal{U}} \frac{\partial \mathcal{U}}{\partial x} + \frac{\partial J}{\partial x}
\]  \tag{74}
Similarly, the gradient of the constraints must satisfy:

$$\frac{\partial R}{\partial U} \frac{\partial U}{\partial x} + \frac{\partial R}{\partial x} = 0$$  \hspace{1cm} (75)$$

Define the following terms:

$$u = \frac{\partial U}{\partial x}, \quad g^\top = \frac{\partial J}{\partial U}, \quad A^\top = \frac{\partial R}{\partial U}, \quad f^\top = -\frac{\partial R}{\partial x}$$  \hspace{1cm} (76)$$

The gradient of the objective function can then be expressed in standard form:

$$\frac{dJ}{dx} = g^\top u + \frac{\partial J}{\partial x}$$  \hspace{1cm} (77)$$

Provided the gradient of the constraints satisfy the following condition:

$$A^\top u = f$$  \hspace{1cm} (78)$$

The term $\frac{\partial J}{\partial x}$ falls out from the objective function; it is presumably specified analytically. On the other hand, under the direct approach, the term $g^\top u$ must be computed directly using finite difference. If the problem has $p$ variables, this involves solving $A^\top u = f$ at least $p + 1$ times since $U$ depends on $x$, which amounts to finding $p + 1$ flow solutions. This is an expensive proposition for high-dimensional problems.

### A.3 Adjoint Formulation

On the other hand, one could take advantage of the adjoint formulation and achieve the same result for the equivalent cost of approximately two flow solutions only. Let $\upsilon$ denote a vector that satisfies the following equation, known as the dual form:

$$A^\top \upsilon = g \Rightarrow \upsilon^\top f = \upsilon^\top Au = (A^\top \upsilon)^\top u = g^\top u$$  \hspace{1cm} (79)$$

Therefore, rather than solving directly for $A^\top u = f$ it is equivalent to solve for $A^\top \upsilon = g$ instead. This is more efficient because computing $g$ and $A$ does not require any new flow solutions and the cost of solving the adjoint equation $A^\top \upsilon = g$ is approximately the same as the cost of one flow solution. Finally, the term $f$ depends only on changes to the mesh due to $x$; it does not require any new flow solutions to be computed either.
APPENDIX B

REVIEW OF HISTORICAL OWN CONCEPTS

Good engineering practice suggests that the design of a new airplane configuration, whether it is evolutionary or revolutionary, should always start where history left off. The purpose of this chapter is therefore to provide a literature review of historical OWN concepts, in order to explain what is different about them today that they can be termed “unconventional.”

B.1 Why OWN

Historically, for the vast majority of commercial transports, engines have either been mounted under and ahead of the wing or on the rear of the fuselage. Experience has shown that conventional locations are effective choices for reducing installation drag to acceptably low levels [7], and industry has not had reason to venture outside this envelope of low-risk engine installations. However, due to a desire for increased efficiency, engine bypass ratio has grown significantly over the years, and the large fan diameters that have resulted are forcing the engines so close to the wing that there is no room left for them to grow any larger due to ground clearance constraints. As bypass ratio increases even further in the future, conventional under-wing installations will therefore no longer be possible without drastic modification of the wing and landing gear.

Alternatively, engines could be placed above the wing. However the historical track record of Over-Wing Nacelle (OWN) concepts warrants concerns about unacceptably high drag levels at transonic speeds. This is because the majority of airplanes that have ever been designed with engines above the wing were usually
not designed for transonic flight or even commercial applications. This represents a challenge for NASA, which must pioneer the design of a new airplane with over-wing nacelles capable of flying efficiently at transonic speeds; an endeavor that has not yet been attempted for a civil airliner. However, if successful, then over-wing concepts could provide game changing advantages previously unattainable with conventional under-wing designs. For example, over-wing nacelles are believed to hold significant potential for reducing the perceived noise levels of commercial transports by taking advantage of the wing to shield ground observers from the sound of the engines (but cabin noise would increase). As Berton explains [8], unlike older technology engines where jet noise was prominent, modern high Bypass Ratio (BPR) engines extract significantly higher amounts of energy from the core flow, effectively reducing the velocity differential between the core jet and fan discharge. This significantly reduces jet noise, which depends on this velocity differential, leaving fan discharge as the dominant noise source. This is desirable because unlike jet noise, which emanates from the engine plume downstream of the airplane, fan discharge noise is localized near the engine and can thus be effectively shielded by the wing. While the HWB offers a similar advantage, OWN might offer a sufficient leap forward to meet ERA goals without drastically departing from conventional tube and wing concepts. For this reason, this research will use OWN as the primary application of the method.

B.2 Historical OWN Concepts

The idea of placing engines on top of the wing is not new, and this section introduces the reader to several historical OWN concepts. It is intended to develop an understanding of how these concepts have been used in the past.

B.2.1 Boeing YC-14 (1970’s)

Wimpress et al. [28] explain that the Boeing YC-14, shown in Fig. 58, was a military experimental aircraft meant to replace the Lockheed C-130 and capable of carrying
large, bulky payloads such as tanks or trucks into and out of short unprepared airfields. It was designed at the outcome of the Vietnam conflict, where it had become apparent that the Air Force faced an airlift dilemma. On one hand, airplanes such as the Lockheed C-141 and C-5A had good payload capability, range and speed; however, they required elaborate air bases with long, paved runways to operate effectively. On the other hand, helicopters could operate from anywhere, but they were slow, vulnerable and could only carry cargo over limited range. In order to solve this dilemma, the Air Force set out to acquire a tactical aircraft that could “interface effectively with the heavy logistics transports and carry men and materials to a point where they could be used directly by the operational troops or picked up and delivered efficiently by helicopters” [28]. This required the airplane to takeoff and land out of 2,000 feet unprepared runways, and carry a 38,000 lb payload over 2,600 nm. Note that that this mission is very different from those flown by modern-day civil transports.

The final prototype resulted in a high wing, T-tail configuration with supercritical airfoils and over-wing nacelles featuring revolutionary Upper Surface Blowing (USB) technology that enabled the YC-14 to reach a maximum lift coefficient of 3.3 at
The airplane was originally designed to cruise at Mach 0.74, but it was soon discovered that holding such a high cruise speed conflicted with the requirement to have very high lift. The former favored a swept wing, whereas the latter favored a straight wing [28]. In order to satisfy Short Take-Off and Landing (STOL) requirements, the Air Force eventually relaxed the cruise speed to Mach 0.66 [64]. Unfortunately, by the time the YC-14 was ready for production, military priorities had changed and the Air Force decided it needed a larger transport that would fly to standard, conventional airfields rather than into battle zones.

B.2.2 Fokker VFW-614 (1970’s)

The first and only commercial transport ever introduced with an over-wing nacelle concept was the Fokker-VFW 614, a quiet short haul airliner designed to cruise at Mach 0.65, shown in Fig. 59. The description by Kathen [32] suggests that the airplane was developed with the vision of capturing a then yet untapped market for small capacity, short haul transport, which is today dominated by airplanes such as the Canadair CRJ or the Embraer ERJ families. It specifically targeted small provisional airfields with low traffic density, and short, potentially unprepared runways.
The airplane measured 70.5 ft in span, 67.5 ft in length, and had a maximum takeoff weight of 44,000 lb, capable of accommodating up to 44 passengers. It was designed for low maintenance, high payload configuration flexibility, quick turn around time, and short takeoff and landing [32].

The top-mounted nacelles allowed ground clearance constraints to be relaxed, resulting in shorter robust landing gear, easier payload access and reduced risk of foreign object damage. As a welcome but unintended benefit of the engine location, the shielding effect of the wing made the VFW-614 the most quiet airplane of its time [32]. However, despite its capabilities, the VFW-614 program was canceled in 1977 due to a lack of demand. In the present author’s opinion, it is reasonable to conjecture that this is perhaps because the market for regional jet transport did not pick up until years later. Today, it still flies as a research aircraft known as the Advanced Technology Testing Aircraft System (ATTAS) operated by the Deutschen Zentrums für Luft- und Raumfahrt (DLR). While its example provides evidence for the advantage of noise shielding in commercial applications, it is safe to say that at Mach 0.65 the VFW-614 did not experience transonic wave drag, and is therefore not representative of modern day civil transports.

B.2.3 ASKA (1970’s)

The ASKA shown in Fig. 60 was a four-engine research aircraft developed by the Japanese National Aeronautics Laboratory (NAL) to investigate the applications of Upper Suction Blowing (USB) technology on commercial transports [10]. It was a conversion of the Kawasaki C-1 into a USB powered-lift aircraft, capable of Short Take-Off and Landing (STOL) by taking advantage of the Coandă effect. While the airplane performed successfully well at low speeds [10], it was never designed for high-speed flight [89]. It’s intended cruise speed was Mach 0.565 [64]. As a consequence, at transonic speeds, it suffered a plethora of aerodynamic inflictions
ranging from strong shock waves and boundary layer separation to scrubbing drag and buffeting between the nacelles [10]. In fact, the design culminated into installation drag reaching 80 percent of the wing-body drag [10], an unacceptably high number for any airplane. However, this is not typical of OWN concepts that employ USB technology. During the development of the YC-14 for example, a staple of USB technology, Wimpress et al. [28] recall an exchange with high-speed aerodynamicists at NASA where the authors revealed that “Boeing had solved, very successfully, the high-speed drag problem of over-wing nacelles” (before the air-force eventually relaxed the target cruise speed from Mach 0.74 to Mach 0.66 for other reasons). Thus, care must be taken not to make a hasty generalization out of the ASKA experience.

B.2.4 NASA QSRA (1970’s)

As explained by Hall [23], the Quiet Short-Haul Research Aircraft (QSRA), shown in Fig. 61, was a research aircraft developed by Boeing for NASA in order to investigate the application of upper surface blowing for quiet short takeoff and landing. Powered lift was generated using four high-bypass turbofan engines blowing the exhaust over the top surface of the wing. By taking advantage of the Coandă effect, the flow
could be deflected downwards using a curved trailing edge flap, effectively converting a portion of the thrust into lift. However, an important feature of the QSRA was to achieve this at a low noise level. It successfully accomplished this using the wing to shield the noise produced by the jet exhaust. Combined with the help of additional acoustic materials through the engine, the noise footprint of the QSRA was as small as 1/7 of an equivalent transport of that period [23]. However, “the QSRA was designed solely as a low-speed research aircraft” [23], which means it was not intended for transonic flight. Even though the isolated wing itself was capable of efficient cruise at Mach 0.74, the airplane as a whole was not designed for those speeds [23]. At such low cruise speed, the QSRA is thus not representative of modern civil transports.

**B.2.5 Honda Jet (Present)**

The Honda Business Jet (HBJ), shown in Fig. 62, is a modern example of a successful over-wing configuration that operates in the transonic regime. The aforementioned work by Fujino et al. [16, 17, 18, 19] provides insight into the development process that led to the final design. In the case of the HBJ, an over-wing concept was selected in order to increase cabin space by removing the engine support structure from
the fuselage. This was motivated by strategic marketing needs to increase passenger comfort, rather than by compelling technical reasons. At the outcome of the investigation, it was discovered that the optimum location of the nacelle with respect to wave drag outperformed even the clean wing. Because this location occurred slightly aft of the shock wave, the authors deduced that drag was lower because the presence of the nacelle decelerates the flow ahead of it, thereby weakening the shock wave.

However, they also found out that the optimum aerodynamic location of their design turned out to be aft of the aero-elastic axis, causing flutter speed to drop. This effect became worse as the engine moved outboard, suggesting the need to either stiffen the inboard wing sections or constrain the engine’s span-wise location to a possibly suboptimal inboard distance. Thus, at the expense of additional design effort to correct undesirable flutter characteristics, the HBJ was able to successfully take advantage of favorable aerodynamic interference to improve fuel efficiency, while meeting strategic marketing goals. This being said, the HBJ is a small, low bypass ratio business jet, which only measures 39.75 ft in span, 42.62 ft in length, and can only carry up to six passengers [16]. It is therefore unclear if such results are directly scalable to sizeable civil transports with ultra-high bypass ratio engines, carrying at
least 150 passengers. The HBJ therefore provides excellent motivation for over-wing nacelle research.

### B.2.6 Other Aircraft (Present)

Russian aerospace companies such as Antonov and Beriev have also developed several OWN concepts, examples of which are shown in Fig. 63. According to product descriptions on the company website, these airplanes do not fly in the transonic regime. For example, the product description of the Beriev Be-200 and A-40 explains that they are multi-purpose amphibious aircraft that cruise at Mach 0.55 and 0.65, respectively. It is therefore reasonable to suggest that the engine location above the wing was selected to prevent seawater damage. Similarly, the product description of the Antonov AN-72 states that it is a military airlifter that cruises at Mach 0.65, and uses upper surface blowing to achieve short takeoff and landing from unprepared airfields. The product description of the Antonov AN-74 explains that it is a modification of the AN-72, designed to support research, reconnaissance and transport operations in the Arctic and Antarctic. It also cruises at the Mach 0.65. It can thus be seen that many of the historical over-wing concepts in use today are usually not intended for high-speed, civil transport missions.

### B.2.7 Synthesis

This brief review of historical OWN concepts served to emphasize that the idea of placing the engines over the wing is not new; however, what is different today is

---

**Figure 63**: Russian OWN concepts
the application. Historically, the primary application of over-wing nacelles has been short takeoff and landing. This was true for the Boeing YC-14, the ASKA, the QSRA and the AN-72/74, which all feature Upper Surface Blowing (USB) technology, while other applications included amphibious aircraft such as the Be-200 and A-40. The only commercial application to date is the VFW-614, but it flew at subsonic speeds. The Honda Business Jet operates in the corporate arena and represents the only modern example an OWN capable of flying efficiently in the transonic regime, but it is a small airplane. In summary, OWN applications are historically not intended for high-speed civil transport, which is why they can still be deemed “unconventional.”
APPENDIX C

REVIEW OF SCREENING METHODS

This chapter reviews two screening methods typically used in aerospace engineering according to Keane and Nair [33]. The first is called Main Effect Screening (MES) and the other is called Elementary Effect Screening (EES). Both of them are compared to Principal Component Analysis (PCA) in Chapter ?? and, therefore, the purpose of this chapter is to familiarize the reader with these methods.

C.1 Main Effect Screening

The theory of MES can be found in Myers and Montgommery [57], but the main points are summarized here. Consider the general linear regression model given by,

\[
f(x) = \sum_{j=1}^{m} \beta_j \phi_j(x)
\]  

(80)

where \( f \in \mathbb{R} \) is some deterministic function of interest, \( x \in \mathbb{R}^p \) is a vector of standardized design variables, \( \beta_j \) are unknown regression coefficients to be found and \( \phi_j \) is some assumed basis function associated with the \( j^{th} \) coefficient. The magnitudes of the coefficients \( \beta_j \) are indicative of the relative importance of a given basis function \( \phi_j \). Consider a dataset of \( N \) samples,

\[
\begin{pmatrix}
  f_1 \\
  \vdots \\
  f_N
\end{pmatrix}
= 
\begin{pmatrix}
  \phi_1(x_1) & \cdots & \phi_m(x_1) \\
  \vdots & \ddots & \vdots \\
  \phi_1(x_N) & \cdots & \phi_m(x_N)
\end{pmatrix}
\begin{pmatrix}
  \beta_1 \\
  \vdots \\
  \beta_m
\end{pmatrix}
\Leftrightarrow
f = X\beta
\]  

(81)

where \( f \in \mathbb{R}^N, X \in \mathbb{R}^{Nm} \) and \( \beta \in \mathbb{R}^m \). Under the standard setting where \( m \leq N \), the solution to Eq. 81 is usually obtained using the Least Squares Estimator (LSE),

\[
\hat{\beta} = \arg \min_{\beta \in \mathbb{R}^m} |f - X\beta|^2
\]  

(82)
The concept behind MES is to choose the basis functions $\phi_j$ such that they are functions of only specific design variables (i.e. $\phi_j(x) = \phi_j(x_j)$). As a result, the magnitude of the coefficient $\beta_j$ provides information about the importance of the quantity it multiplies (i.e. the design variables associated with it). In the methods by Myers and Montgomery [57], the basis functions are taken to form a linear model containing main effects and two-factor interactions only,

$$f = \beta_0 + \sum_{i=1}^{p} \beta_i x_i + \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} \beta_{ij} x_i x_j$$  \hspace{1cm} (83)

where higher-order interactions are neglected under the Sparsity-of-Effects Principle [59]. The number of samples needed is obtained by counting the number of regression coefficients $\beta$:

$$N = 1 + p + p(p - 1)/2$$  \hspace{1cm} (84)

For large design spaces, this number can be large. However, it can be reduced by grouping terms together in a concept known as aliasing [57]. For example, a three-variable model can be reduced from $2^3 = 8$ coefficients to just four using aliasing and ignoring Higher Order Terms (HOT):

$$f = \beta_0 + \beta_1 (x_1 + x_2 x_3) + \beta_2 (x_2 + x_1 x_3) + \beta_3 (x_3 + x_1 x_2)$$  \hspace{1cm} (85)

However, it is no longer possible to tell main effects $x_i$ apart from interactions $x_j x_k$ since the magnitude of $\beta_j$ only provides information about the group $x_i + x_j x_k$. The ability to distinguish main effects apart from interactions is a property known as resolution [57]. It is determined by the sampling plan. In MES, sampling plans are typically chosen to be fractional factorial designs of resolution III and IV [57] or Plackett-Burman designs [66]:

- **Resolution III designs** are designs in which no main effects are aliased with any other main effect, but main effects are aliased with two-factor interactions, and two-factor interactions may be aliased with each other.
Resolution IV designs are designs in which no main effects are aliased with any other main effect or with any two-factor interaction, but two-factor interactions may be aliased with each other.

Once the regression coefficients $\beta$ are found, the last step in MES is to perform an analysis of variance [54] (ANOVA) in order to rank them in order of relative importance. Main effects associated with statistically significant $\beta$’s are deemed important, others are discarded.

C.2 Elementary Effect Screening

The theory behind EES was developed by Morris [55] and is summarized here-under. EES is a screening method that does not rely on the sparsity of effects principal [57], and unlike MES, it holds the advantage of discerning nonlinear effects. Provided a smooth objective function $f$ evaluated at some point $x$ in the design space, the $j^{th}$ elementary effect associated with $x_j$ is defined as,

$$d_j(x) = \frac{f(x_1, \ldots, x_j + \Delta, \ldots, x_p) - f(x)}{\Delta} \quad \forall \quad j = 1, \ldots, p \quad \text{and} \quad \Delta = \frac{C}{1 - k} \quad (86)$$

where $C$ is some user-defined constant, $x \in \mathbb{R}^p$ is a vector of $p$ design variables belonging to the unit hypercube and $k$ is some preselected integer used to discretize the unit hypercube into a $k$-level lattice. An elementary effect is therefore akin to a partial derivative, except that $\Delta$ is not necessarily small. The central idea of the method is that each $d_j$ is a random variable with unknown distribution $F_j$. Taken together, they form a random vector:

$$F = \begin{pmatrix} d_1 & \ldots & d_p \end{pmatrix}^\top \quad (87)$$
In order to estimate this distribution, EES proceeds by collecting \( r \) random observations of \( F \):

\[
\begin{pmatrix}
F_1 \\
\vdots \\
F_r
\end{pmatrix}
= \begin{pmatrix}
d_{11} & \ldots & d_{1p} \\
\vdots & \ddots & \vdots \\
d_{r1} & \ldots & d_{rp}
\end{pmatrix}
\in \mathbb{R}^{rp}
\tag{88}
\]

Note that each observation \( F_i, \forall i = 1, \ldots, r \) requires the evaluation of \( p \) elementary effects, each taken at different, randomly selected points in the design space. This amounts to a total of \( rp \) samples of elementary effects. Since each elementary effect requires two function evaluations itself, a total of \( 2rp \) function evaluations is thus needed. Provided with a sufficient number of samples, the mean and variance of \( F_j \) can then be found for each elementary effect \( d_j \). Large measures of central tendency indicate variables with large influence on the output, whereas large measures of spread indicate variables responsible for nonlinear effects. Elementary effects with small mean and variance therefore indicate design variables that can be discarded.
This chapter presents some complementary results for Chapter 4. In that chapter, it was explained in Section 4.2 that there is a need to scale the variables and it was proposed to use a typical size to do so. The purpose of this chapter is to show through example that scaling by the typical size yields the right result indeed.

**D.1 Screening using Elementary Effects**

The driving parameters for the problem described in Chapter 4 are given by Forrester et al. [14], where those authors used elementary effect screening [55] (EES). They are summarized in Fig. 64, which is taken directly from their text. The reader may refer to Appendix C for an overview of EES. This figure shows the sample standard deviation of each elementary effect on the $y$-axis and the corresponding sample mean on the $x$-axis. The location of the parameters on the plot therefore provides an easy way of spotting those whose elementary effect distribution has either a large mean or large standard deviation (or both). All others can be discarded. A description of the parameters was previously provided in Table 1 from Chapter 4. It can be seen that EES predicts that $W_{dg}$, $t/c$, $A$, $S_w$ and $N_z$ are driving the weight of the wing; the others are less important and could possibly be ignored. These parameters correspond to the design gross weight, thickness-to-chord ratio, aspect ratio, wing area and ultimate load factor, respectively. Forrester et al. [14] provide an insightful discussion for why these parameters make sense.
Figure 64: Screening using elementary effects (reprinted [14])

D.2 Screening using Partial Derivatives

An elementary effect is an approximate measure of sensitivity. It is constructed in the same way as a first order finite difference, except that the finite step is not small. A partial derivative, by contrast, is an exact measure of sensitivity and, therefore, is much more precise. It could thus be used in the same way as an elementary effect is used for screening in EES. Said otherwise, one can randomly sample the gradient and use the data to estimate the distribution of each partial derivative. Then, based on the mean and standard deviation, one can make an assessment as to whether or not a given variable has an important effect on the response. Variables associated with partial derivative distributions that have either a large mean or large standard deviation (or both) are deemed significant; others can be discarded. This is the same idea as LSGT, except that PCA is not performed. The results of such a process will now be presented for two cases.

In the first case, design variables and partial derivatives are scaled by typical value
Figure 65: Screening using scaled partial derivatives (symbols: $Sw = S_w$, $Wfw = W_{fw}$, $A = A$, $L = \Lambda$, $q = q$, $h = \lambda$, $tc = t/c$, $Nz = N_z$, $Wdg = W_{dg}$, $Wp = W_p$)
Figure 66: Screening using unscaled partial derivatives (symbols: $S_w = Sw$, $Wfw = W_{fw}$, $A = A$, $L = \Lambda$, $q = q$, $h = \lambda$, $tc = t/c$, $Nz = N_z$, $Wdg = W_{dg}$, $Wp = W_p$)
using the same typical values described in Section 4.2. The reader should refer to Section 3.2.2 for what is meant by scaling. Associated screening results are shown in Fig. 65, presented in the form Pareto plots. It can be seen that they agree very well with Fig. 64, which is known to be correct. Scaling by typical value therefore yields the right answer.

By contrast, in the second case, design variables and partial derivatives are not scaled at all, i.e. $W = I$. These results are shown in Fig. 66. It can be seen that they do not agree with the correct answer in Fig. 64 and, in fact, do not even make sense intuitively. In particular, the weight of the paint $W_p$ is found to be the second most important driver of the weight of the wing, which of course is senseless. This counter-example serves to show the importance of scaling the variables appropriately by some typical order of magnitude. Failure to do so will lead to the wrong answer.
APPENDIX E

CANONICAL EXAMPLE MATLAB CODE

This chapter provides the MATLAB code that was used in chapter 4, for the reader’s convenience, in case he or she wishes to use it as template for implementing this method on their own.

E.1 Main Program

```matlab
% Empirical wing weight estimation example
% -----------------------------------------
% Ref: 1. Alexander I. J. Forrester, Andrés Sbester and Andy J. ... Keane,
% "Engineering Design via Surrogate Modeling," Wiley & ... Sons, p.10

clear all
close all
clc

% Symbol Parameter Typical light aircraft value (C172)
% S_w -Wing area (ft^2) 174
% W_fw -Weight of fuel in the wing (lb) 252
% A -aspect ratio 7.52
% Lambda -quarter-chord sweep (deg) 0
% q -dynamic pressure at cruise (lb/ft^2) 34
% lambda -taper ratio 0.672
% tc -aerofoil thickness to chord ratio 0.12
```
% N,z  -ultimate load factor (1.5x limit load factor) 3.8
% Wdg  -flight design gross weight (lb)    -2000
% W,p  -paint weight (lb/ft^2)            -0.064

% Names
names{1} = 'Sw';
names{2} = 'Wfw';
names{3} = 'A';
names{4} = 'L';
names{5} = 'q';
names{6} = 'h';
names{7} = 'tc';
names{8} = 'Nz';
names{9} = 'Wdg';
names{10} = 'Wp';

% Design space (taken from Table 1 in Ref. 1)
x0 = [174 252 7.52 0 34 0.672 0.12 3.8 2000 0.064];
xL = [150 220 6.00 -10 16 0.500 0.08 2.5 1700 0.025];
xU = [200 300 10.0 10 45 1.000 0.18 6.0 2500 0.080];
xT = [100 100 1.0 1 10 0.100 0.10 1.0 1000 0.010];

% Take reference point at center of design space
xref = 0.5*(xL+xU);

% Objective function
f = @(x) FunTest(x);

% Number of design variables
p = length(x0);

% Step 1: Scaling and centering
% -----------------------------
W = diag(1./xT);
z0 = (W*(x0-xref)');
zL = (W*(xL-xref)');
zU = (W*(xU-xref)');

% Step 2: Sample the gradient
% -----------------------------

% Number of gradient samples to collect
n = round(5*p*log(p)); % some integer multiple of p log p

% Generate a random sampling plan
xdoe = @(n,p) repmat(xL,n,1) + rand(n,p).*repmat((xU-xL),n,1);
x = xdoe(n,p);

% Loop
y = zeros(n,1);
dydx = zeros(n,p);
for i = 1:n
    [y(i),dydx(i,:)] = f(x(i,:));
end

% Scale gradient
dydz = (W\dydx)';

% Plot convergence history
estimation_error(dydz); % This is calling a supporting function
print('-depsc','step2_convergence.eps');

% Write to file
csvwrite('step2_dydx.csv',dydx);
csvwrite('step2_dydz.csv',dydz);

% Step 3: PCA
% -----------------
[pc,~,latent] = pca(dydz);
latent = sqrt(latent);

% Write to file
csvwrite('step3_pc.csv',dydx);
csvwrite('step3_latent.csv',latent);

% Make a colorful matrix for A = pc
figure
imagesc(pc);
colormap(jet);
colorbar;
xlabelNames = ... {
    {'a_1','a_2','a_3','a_4','a_5','a_6','a_7','a_8','a_9','a_10'};
set(gca,'XTickLabel',xlabelNames); % gca gets the current axis
set(gca,'YTickLabel',names); % gca gets the current axis
print('-dpng','step3_pc.png');

% Step 4: dimensionality reduction
% --------------------------------------------
% Make Pareto plots
pareto(latent)
xlabel('Principal components','fontname','times');
ylabel('$\sqrt{\hat{\lambda}}$','interpreter','latex');
grid on
print('-depsc',strcat('step3_pareto_1.eps'));

%
pareto(abs(mean(dydz*pc)))
xlabel('Principal components');
ylabel('$|\bar{\upsilon}|$', 'interpreter', 'latex');
grid on
print('-depsc', strcat('step3_pareto_2.eps'));

% Set of PCs responsible for nonlinear effects
 t = find(cumsum(latent) / sum(latent) ≥ 0.99, 1); % std. dev
%
% Set of PCs responsible for linear effects
  [mu, I] = sort(abs(mean(dydz)*pc), 'descend');
  s = find(cumsum(mu) / sum(mu) ≥ 0.99, 1); % mean

% Take union of both sets and re-arrange columns of A = pc ...
    accordingly
 pcols = 1:p;
 qcols = union(1:t, I(1:s));
 rcols = setdiff(pcols, qcols);
 q = length(qcols);
 fprintf('q = %i
', q);
 pc = pc(:, [qcols rcols]);
 latent = latent([qcols rcols]);

% Display PC retained
 disp('PC used:')
 disp(qcols)

% This information must be stored in a structure called "inp" for ...
    later use
 inp.W = W;
 inp.grad_z = dydz;
 inp.ximin = zL;
inp.xmax = zU;
ip.xmin = zL;
inp.zmax = zU;
inp.p = p;
inp.q = q;
inp.pc = pc;
inp.g = 0;

% Step 5-6: Mapping & Validation
% ------------------------------

% Number of validation points
n = 1000;

% Generate a random sampling plan
zdoe = @(n,p) repmat(zL',n,1) + rand(n,p).*repmat((zU'-zL'),n,1);
ztrue = zdoe(n,p);

% Loop
ytrue = zeros(n,1);
for i = 1:n
    % Convert back to x
    x = W\ztrue(i,:)+ xref';
    % Evaluate
    ytrue(i) = f(x);
end

% Convert DOE to q-dimensional latent space
xiq = ztrue*pc(:,1:q);
% Re-evaluate DOE using PCA approximation
ypred = zeros(n,1);
for i = 1:n
    % Convert back to z
    z = map(inp,xiq(i,:)); % This is calling a supporting function
    % Convert back to x
    x = W\z+xref';
    % Evaluate
    ypred(i) = f(x);
end

% Make validation plots
make_validation_plots(ypred,ytrue,'FunTest'); % This is calling ... an external function

% Step 7: Design space exploration
% ---------------------------------------------
% Handle to generate a latin hypercube sampling plan
zdoe = @(n,p) repmat(zL',n,1) + ...
hshsdesign(n,p,'iterations',10).*repmat((zU'-zL'),n,1);

% Training data for surrogate models
n = 1000; % Number of samples
doe = zdoe(n,p);
xtrain = doe*inp.pc(:,1:q);
csvwrite('step7_training_data.csv',xtrain)

% Validation data
% NOTE: same thing
doe = zdoe(n,p);
xival = doe*inp.pc(:,1:q);
csvwrite('step7_validation_data.csv',xival)

% NOTE: Ideally, the sampling plan should have been created in ... reduced latent space directly, but this is challenging because ... it is a rotated hypercube and, therefore, denotes linear ... constraints. In this simple example, I got around this by ... creating the sampling plan in the original p-dimensional ... space, which does not have constraints (other than the ranges ... of the design space) and projecting it onto reduced latent ... space. This is okay for this problem because p = 10, which is ... not that high. However, for larger problems, this is a really ... inefficient approach. In those cases, a sampling plan should ... be used, which is capable of accounting for constraints directly.

% Visualize sampling plan
gplotmatrix(xitrain)

% Evaluate training and validation data
ytrain = zeros(n,1);
yval = zeros(n,1);
for i=1:n

% Training
ztrain = map(inp,xitrain(i,:));
x = W'ztrain+xref';
ytrain(i,1) = f(x);

% Validation
zval = map(inp,xival(i,:));
x = W\zval+xref';

yval(i,1) = f(x);

end

% Fit surrogate models using RSE
b = make_fit(xitrain,ytrain,,3,'RSE'); % Calls support function

% Create function handles
FunRSE_xi = @(xi) RSE(xi,b); % RSE in terms of xi
FunRSE_z = @(z) RSE(pc(:,1:q)'*reshape(z,p,1),b); % RSE in terms ...
   of z
FunRSE_x = @(x) RSE(pc(:,1:q)'*W*(reshape(x,p,1)-xref'),b); % ...
   RSE in terms of x

% Validate RSE
ypred = zeros(n,1);
for i = 1:n
    ypred(i) = FunRSE_z(ztrue(i,:)); ...
    %FunRSE(inp.pc(:,1:q)'*ztrue(i,:'));
end
make_validation_plots(ypred,ytrue,'step7_RSE');

% Sensitivity plots centered a x0 (compare actual vs. predicted)
N = 100;
x=zeros(N,p);
y_pred=zeros(N,1);
y_true=zeros(N,1);
for i = 1:p
    for j=1:N
        x(j,:) = x0';
        x(j,i) = xL(i)+(xU(i)-xL(i))/99*j;
        y(j,i) = xL(i)+(xU(i)-xL(i))/99*j;
    end
end
\[ y_{\text{pred}}(j) = \text{FunRSE}(x(j,:)); \]
\[ y_{\text{true}}(j) = f(x(j,:)); \]
\end

\texttt{fig=figure('units','inches','position',[6 6 6 6]);}
\texttt{axpos = [1.5 1.5 3.25 3.25];}
\texttt{ax = ...}
\texttt{axes('parent',fig,'units','inches','position',axpos,'fontname','times ...}
\texttt{new roman','fontsize',11});
\texttt{hold on}
\texttt{plot(x(:,i),y_{\text{pred}})}
\texttt{plot(x(:,i),y_{\text{true}},'--')}\]
\texttt{legend('Predicted','Actual')}\]
\texttt{axis([xL(i) xU(i) 200 300])}
\texttt{xlabel(names(i))}
\texttt{ylabel('Wing weight')}\]
\texttt{print('-depsc',strcat('W_{vs\_x}',num2str(i),'.eps'))};
\texttt{end}

% Surrogate-Based Exploration using SQP with multiple restarts
% NOTE: any other approach to exploration is valid, this was just ...
% my preference

% Number of multiple restarts
\texttt{n = 100;}

% Optimizer options
\texttt{options = optimoptions('fmincon',...}
\texttt{'Algorithm','sqp',...}
\texttt{'MaxIter',1000,...}
\texttt{'MaxFunEvals',1000,...}
\texttt{'Display','iter');}
% Initialize arrays for speed
xguess = zeros(n,p);
xbest_pred = zeros(n,p);
xbest_true = zeros(n,p);
ybest_pred = zeros(n,1);
ybest_true = zeros(n,1);
exitflag_pred = zeros(n,1);
exitflag_true = zeros(n,1);

% Nonlinear constraints
fcon = @(xi) nonlcon(inp,xi); % This handle calls a supporting ...
    function

% Linear inequality constraints
A = [];
b = [];

% Linear equality constraints
Aeq = [];
beq = [];

% Bound constraints (taken care of in nonlcon.m)
lb = [];
ub = [];

% Loop and optimize
for i = 1:n
    fprintf('Initial guess %i of %i
',i,n);
    % Randomly choose a starting point by sampling reduced latent ... space
% NOTE: in this case I chose points from the training data ...
since they're available

xiguess = xitrain(randi(length(ytrain),1),:);

% Convert back to z-space
zguess = map(inp,xiguess);

% Convert back to x-space
xguess(i,:) = (W\zguess+xref')';

% There are two options to solve using LSGT
option = 2;
if (option == 1)

    % Optimize in reduced latent
    [xibest,ybest_pred(i),exitflag_pred(i)] = ...
        fmincon(FunRSE_xi,xiguess,A,b,Aeq,beq,lb,ub,fcon,options);
    zbest = map(inp,xibest);
    xbest_pred(i,:) = (W\zbest+xref')';

else

    % Take advantage of surrogates to solve in p-dimensional ...
    % space
    % NOTE: This option is only useful if surrogate models ...
    % exist, otherwise choose option 1.
    %
    % The advantage is that it's easier to setup.
    [xbest_pred(i,:),ybest_pred(i),exitflag_pred(i)] = ...
        fmincon(FunRSE_x,xguess(i,:),[],[],[],[],xL,xU,[],options);

end
% Optimize using true response (for comparison)
[xbest_true(i,:),ybest_true(i),exitflag_pred(i)] = ...
    fmincon(f,xguess(i,:),[],[],[],[],[],[],options);

end

% Write to file (so that user may compare externally)
csvwrite('step_7-8_xbest_true.csv',xbest_true);
csvwrite('step_7-8_xbest_pred.csv',xbest_pred);
csvwrite('step_7-8_ybest_true.csv',ybest_true);
csvwrite('step_7-8_ybest_pred.csv',ybest_pred);

% Step 8: Optimization
% ----------------------------
% Usually, one would take the answers from step 7 and use them as ...
% initial guess for step 8. However, the answer from step 7 was ...
% already pretty close to the true optimum. Step 8 is left as a ...
% trivial exercise to the reader.

E.2 Supporting Functions Called by Main Program

This section provides the support functions called by the main program.

E.2.1 FunTest.m

function [W,dW] = FunTest(design)
% This is the test function used for the problem.
% W = liftsurfw(design)
% S_w = design(1);
% W_fw = design(2);
% A = design(3);
% Lambda = design(4)*pi/180;
% q = design(5);
% lambda = design(6);
% tc = design(7);
% N_z = design(8);
% W_dg = design(9);
% W_p = design(10);

% Symbol Parameter Typical light aircraft value (C172)
% S_w -Wing area (ft^2) 174
% W_fw -Weight of fuel in the wing (lb) 252
% A -aspect ratio 7.52
% Lambda -quarter-chord sweep (deg) 0
% q -dynamic pressure at cruise (lb/ft^2) 34
% lambda -taper ratio 0.672
% tc -aerofoil thickness to chord ratio 0.12
% N_z -ultimate load factor (1.5x limit load factor) 3.8
% W_dg -flight design gross weight (lb) -2000
% W_p -paint weight (lb/ft^2) -0.064

% *********************************************************
% W -wing weight lbs (est. by the eq./actual) 245/236

% Copyright 2007 A Sobester

% This program is free software: you can redistribute it and/or ... 
modify it
% under the terms of the GNU Lesser General Public License as ... 
published by
% the Free Software Foundation, either version 3 of the License, ... or any
% later version.
%
% This program is distributed in the hope that it will be useful, but
% WITHOUT ANY WARRANTY; without even the implied warranty of
% MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the ...
    GNU Lesser
% General Public License for more details.
%
% You should have received a copy of the GNU General Public ...
    License and GNU
% Lesser General Public License along with this program. If not, see
% <http://www.gnu.org/licenses/>.
%
% Modified by Steven H. Berguin (added gradient)

x1 = design(1);
x2 = design(2);
x3 = design(3);
x4 = design(4)*pi/180;
x5 = design(5);
x6 = design(6);
x7 = design(7);
x8 = design(8);
x9 = design(9);
x10 = design(10);

W = ...
    0.036*x1^0.758*x2^0.0035*(x3/cos(x4)^2)^0.6*x5^0.006*x6^0.04*...
    (100*x7/cos(x4))^0.49 + x1*x10;
\[ dW_{dx1} = \ldots \\
0.036 \times 0.758 \times x_1 \times (0.758-1) \times x_2 \times 0.0035 \times (x_3 / \cos(x_4))^2 \times 0.6 \times x_5 \times 0.006 \times \ldots \\
x_6 \times 0.4 \times (100 \times x_7 / \cos(x_4))^{-0.3} \times (x_8 \times x_9)^0.49 + x_{10}; \]

\[ dW_{dx2} = \ldots \\
0.036 \times x_1 \times 0.758 \times 0.0035 \times x_2 \times (0.0035-1) \times (x_3 / \cos(x_4))^2 \times 0.6 \times x_5 \times 0.006 \times \ldots \\
x_6 \times 0.4 \times (100 \times x_7 / \cos(x_4))^{-0.3} \times (x_8 \times x_9)^0.49; \]

\[ dW_{dx3} = \ldots \\
0.036 \times x_1 \times 0.758 \times x_2 \times 0.0035 \times 0.6 \times (x_3 / \cos(x_4))^2 \times (0.6-1) \times x_5 \times 0.006 \times \ldots \\
x_6 \times 0.4 \times (100 \times x_7 / \cos(x_4))^{-0.3} \times (x_8 \times x_9)^0.49; \]

\[ a = (0.036 \times x_1 \times 0.758 \times x_2 \times 0.0035 \times x_3 \times 0.6); \]

\[ b = (x_5 \times 0.006 \times x_6 \times 0.04 \times (100 \times x_7)^{-0.3}); \]

\[ c = (x_8 \times x_9)^0.49; \]

\[% W(x_4) = a \times (1 / \cos(x_4)^2)^0.6 \times b \times (1 / \cos(x_4))^{-0.3} \times c + x_1 \times x_{10}; \]

\[ dW_{dx4} = \ldots \\
0.9 \times a \times b \times c \times (1 / (\text{eps} + \cos(x_4)))^0.7 \times (1 / (\text{eps} + \cos(x_4)^2))^{0.6} \times \sin(x_4); \]

\[ dW_{dx5} = \ldots \\
0.036 \times x_1 \times 0.758 \times x_2 \times 0.0035 \times (x_3 / \cos(x_4))^2 \times 0.6 \times x_5 \times 0.006 \times (0.006-1) \times \ldots \\
x_6 \times 0.4 \times (100 \times x_7 / \cos(x_4))^{-0.3} \times (x_8 \times x_9)^0.49; \]

\[ dW_{dx6} = \ldots \\
0.036 \times x_1 \times 0.758 \times x_2 \times 0.0035 \times (x_3 / \cos(x_4))^2 \times 0.6 \times x_5 \times 0.006 \times 0.04 \times \ldots \\
x_6 \times (0.04-1) \times (100 \times x_7 / \cos(x_4))^{-0.3} \times (x_8 \times x_9)^0.49; \]

\[ dW_{dx7} = \ldots \\
0.036 \times x_1 \times 0.758 \times x_2 \times 0.0035 \times (x_3 / \cos(x_4))^2 \times 0.6 \times x_5 \times 0.006 \times 0.04 \times \ldots \\
(-0.3) \times (100 / \cos(x_4)) \times (100 \times x_7 / \cos(x_4))^{-0.3-1} \times (x_8 \times x_9)^0.49; \]

\[ dW_{dx8} = \ldots \\
0.036 \times x_1 \times 0.758 \times x_2 \times 0.0035 \times (x_3 / \cos(x_4))^2 \times 0.6 \times x_5 \times 0.006 \times 0.04 \times \ldots \\
(100 \times x_7 / \cos(x_4))^{-0.3} \times 0.49 \times x_9 \times (x_8 \times x_9)^0.49-1; \]

\[ dW_{dx9} = \ldots \\
0.036 \times x_1 \times 0.758 \times x_2 \times 0.0035 \times (x_3 / \cos(x_4))^2 \times 0.6 \times x_5 \times 0.006 \times 0.04 \times \ldots \\
(100 \times x_7 / \cos(x_4))^{-0.3} \times 0.49 \times x_8 \times (x_8 \times x_9)^0.49-1; \]

\[ dW_{dx10} = x_1; \]
% dW_dx4 is in lb/rad but x4 is provided in deg => must convert ... 
    back to deg

dW_dx4 = dW_dx4*pi/180;

% Compile partial into gradient vector

dW=[dW_dx1;dW_dx2;dW_dx3;dW_dx4;dW_dx5;dW_dx6;dW_dx7;dW_dx8;dW_dx9;dW_dx10];

E.2.2 estimation_error.m

function estimation_error(x)

    % This function computes the convergence history of the ... estimation error:
    % |S_N - Sigma|_inf = ||Sigma|| * sqrt(re(Sigma)/N)

    % Inputs:
    % x = data to be used for computing covariance matrix, S_N

    % Size
    [n,\_] = size(x);

    % Check n large enough
    if (n < 2)
        return
    end

    % Initialize
    max_eig = zeros(size(x,1),1);
    Tr = zeros(size(x,1),1);
    re = zeros(size(x,1),1);
    err = zeros(size(re));
% Calculate
for i = 1:length(max_eig)
    max_eig(i) = norm(cov(x(1:i,:)));  
    Tr(i) = trace(cov(x(1:i,:)));  
    re(i) = Tr(i)/max_eig(i);  
    err(i) = max_eig(i)*sqrt(re(i)/i);  
end

% Setup figure
fig=figure('units','inches','position',[6 6 6]);  
axpos = [1.25 1.25 3.5 3.5];  
axes('parent',fig,'units','inches','position',axpos,'fontname','times ...  
new roman','fontsize',11);

% Plot
plot(1:i,err(1:i))  
xlim([0 i-1])  
xlabel('Number of samples')  
ylabel('$\|S_N\|\sqrt{re(S_N)/N}$','interpreter','latex')  
print('-depsc',strcat('convergence_plot.','eps'));}

E.2.3  map.m

function [z,xi]=map(inp,xi_q)
%
% This function establishes the backwards mapping necessary  
% to go from reduced latent space in terms of xi_q back to the  
% original design space, in terms of z. This is achieved by  
% solving a quadratic programming problem, as explained
}

150
% in my thesis.

% Inputs:
% inp  = structure containing the following information:
%       inp.p  = size of original design space
%       inp.pc = matrix whose columns are the principal ...
%                  components
%       inp.zmin= lower bound for z (i.e. vector of scaled ... variables)
%       inp.zmax= upper bound for z (i.e. vector of scale ... variables)
%       xi.q  = vector of reduced latent variables to be mapped back

% Dimensions
dims = size(xi_q);

% Check dimensions
if (dims(1) == 1)
   xi_q = xi_q';
end

% Dimensions
q = length(xi_q);
p = inp.p;

% Transformation matrix
A = inp.pc;

% Find a suitable value for defaulted variables xi_r
if (q < p)
   % Partition
A_q = A(:,1:q);  
A_r = A(:,q+1:p);

% Linear constraints
M = [A_r; -A_r];  
b = [inp.zmax; -inp.zmin]-[A_q; -A_q]*xi_q+eps;

% Quadratic program
H = eye(p-q);  
f = zeros(p-q,1);  
xi_r0 = zeros(p-q,1);  
lb = [];  
ub = [];
options = optimset('Algorithm','active-set','Display','off');  
xi_r = quadprog(H,f,M,b,lb,ub,xi_r0,options);

% Concatenate
if (isempty(xi_r))  
  xi = [xi_q; xi_r0];
else  
  xi = [xi_q; xi_r];
end
else
  xi = xi_q;
end

% Map xi back to z
z = A * xi;

E.2.4 make_validation_plots.m
function make_validation_plots(pred, actual, name)
% This program makes validation plots given predicted and actual data.
%
% Inputs:
% pred = vector of predicted data values
% actual = vector of actual data values
% name = string to be used for output file names

% Filter out NaN
actual = actual(isnan(pred)==0);
pred = pred(isnan(pred) == 0);

% Model Representation Error (MRE), more commonly known as validation error
MRE = (pred - actual);

% Compute validation R-squared
TSS = sum((pred-mean(pred)).^2);
RSS = sum((pred-actual).^2);
R2 = real(1 - RSS/TSS);

% MRE histogram
fig = figure('units','inches','position',[6 6 6 6]);
axpos = [1.5 1.5 3.25 3.25];
axes('parent',fig,'units','inches','position',axpos,'fontname','times new roman','fontsize',11);
hist(MRE)
xlabel('Validation error','fontname','times')
ylabel('Count','fontname','times')
title(strcat('\mu = ',num2str(round(10000*mean(MRE))/10000),', ... \
sigma = ... 
',num2str(round(10000*std(MRE))/10000)),'fontname','times')

print('-depsc',strcat(name,'MRE','.eps'));

% Actual by predicted plot
fig=figure('units','inches','position',[6 6 6 6]);
axpos = [1.5 1.5 3.25 3.25];
axes('parent',fig,'units','inches','position',axpos,'fontname','times new roman','fontsize',11);
hold on
plot(pred,actual,'r+','MarkerSize',5);
plot(actual,actual,'k');
legend('Validation','Perfect fit line',4)
xlabel('Predicted')
ylabel('Actual')
title(cat(2,'R^2 Validation =',num2str(round(10000*R2)/10000)))
hold off
print('-depsc',strcat(name,'_actual_by_predicted','.eps'));

% Residual by predicted plot
fig=figure('units','inches','position',[6 6 6 6]);
axpos = [1.5 1.5 3.25 3.25];
axes('parent',fig,'units','inches','position',axpos,'fontname','times new roman','fontsize',11);
hold on
plot(pred,MRE,'r+','MarkerSize',5);
xlabel('Predicted')
ylabel('Residual')
print('-depsc',strcat(name,'_residual_by_predicted','.eps'));

E.2.5 make_fit.m
function b = make_fit(x,y,n,fname)

% This function fits and n'th order RSE fit using using either ... the Least
% Squares Estimator (LSE) or the LASSO estimator, depending on ... whether or
% not the number of training data is greater or less than the ... number of
% unknown regression coefficients.

% Inputs:
% x = N x p matrix of training data containing sample locations
% y = N x 1 vector of training data containing sample values
% n = scale indicating desired RSE order
% fname = name to call RSE output function

% Outputs:
% b = regression coefficients
% fname.m = output file containing RSE (written to current directory)
% (note: use will have to provide b to fname.m)

% Number of samples
N = length(y);
q = size(x,2);

% Create assumed RSE
m = build_rse4(q,n,'F'); % This calls a supporting function

% This next line seems pointless, but it actually fixed a problem ... I was
% having.
ls F.m
% Build X and y for linear regression: y=X*b (b is what we're solving for)
Y = y;
X = zeros(N,m);
for i = 1:N
    [~,phi] = F(x(i,:),zeros(m,1));
    X(i,:) = phi';
end

% Regress
tic
if (length(Y) < m)
    disp('n < m using LASSO')
    % LASSO regression
    [B,stats] = lasso(X,Y,'CV',10);
    % Get MSE and standard error of MSE
    i = stats.IndexMinMSE;
    ME = sqrt(stats.MSE(i));
    SE = sqrt(stats.SE(i));
    % Cross validation plot
    lassoPlot(B, stats, 'PlotType', 'CV')
    title(strcat('\sqrt(MSE^*) = ',num2str(ME),'\pm',num2str(SE)))
    saveas(gcf,strcat('CV_',fname,'.eps'),'eps')
    % Return coefficients
    b = B(:,i);
    b(1) = stats.Intercept(:,i);
else
    disp('n ≥ m using LSE')
    \[ b = \text{regress}(Y, X); \]
end
fprintf('Elapsed time is %f minutes.
', toc/60);
copyfile('F.m', strcat(fname, '.m'));

E.2.6 build_rse4.m

function m = build_rse4(p, n, fname)

% This is a quick and dirty function that generates the basis ...
    functions for
% a Response Surface Equation (RSE) up to 4th order.
%
% Author: Steven H. Berguin
%
% p = number of design variables
% n = order of polynomial
% fname = name of function
%
% Output:
% m = number of basis functions
% fname.m = m-file containing RSE equation (written to current ...
    directory)
% phi_i = m-files containing basis function equation (i = 1,...,m)
%
% Intercept
t = 1 + n*p ...
    + \text{factorial}(p)/(\text{factorial}(p-2) \ast \text{factorial}(2)) ... 
    + \text{factorial}(p)/(\text{factorial}(p-3) \ast \text{factorial}(3)) + p \ast (p-1) ...
\[ + \frac{\text{factorial}(p)}{\text{factorial}(p-4) \times \text{factorial}(4)} + p \times (p-1) \times (p-2); \]

\[
\phi = \text{cell}(t,1);
\]

\[
m = 1;
\]

\[
\phi\{m,1\} = '1';
\]

% First order terms
\[
\text{for } i = 1:p
\]
\[
m = m + 1;
\]
\[
\phi\{m,1\} = \text{strcat('x(',num2str(i),'))'};
\]

end

% Second order terms
\[
\text{if } (n > 1)
\]
\[
\text{for } i = 1:p
\]
\[
\text{for } j = i:p
\]
\[
m = m + 1;
\]
\[
\phi\{m,1\} = \ldots
\]
\[
\text{strcat('x(',num2str(i),')*','x(',num2str(j),')');}
\]

end

end

% Third order terms
\[
\text{if } (n > 2)
\]
\[
\text{for } i = 1:p
\]
\[
\text{for } j = i:p
\]
\[
\text{for } k = j:p
\]
\[
m = m + 1;
\]
\[
\phi\{m,1\} = \ldots
\]
\[
\text{strcat('x(',num2str(i),')*','x(',num2str(j),')*','x(',num2str(k),')');}
\]

end

end
% Fourth order terms
if (n > 3)
    for i = 1:p
        for j = i:p
            for k = j:p
                for l = k:p
                    m = m + 1;
                    phi{m,1} = ...
                    strcat('x(',num2str(i),')*','x(',num2str(j),')*','x(',num2str(k),')*','x(',num2str(l),')');
                end
            end
        end
    end
end

% Higher-order polynomial terms
if (n > 4)
    for i = 5:n
        for j = 1:p
            m = m + 1;
            phi{m,1} = strcat('x(',num2str(j),')ˆ',num2str(i));
        end
    end
end

% Write output function
filename = strcat(fname,'.m');
outFile = fopen(filename,'w');
fprintf(outFile,strcat('function [y,phi] = ',fname,'(x,b)\n'));
fprintf(outFile, strcat('phi = zeros(',num2str(m),',1);\n'));

for i = 1:m
    fprintf(outFile, strcat('phi('. num2str(i) ')= ', phi{i}, ';'\n'));
end

fprintf(outFile, 'y=0; \n');

fprintf(outFile, strcat('for i=1:',num2str(m),'
'));

fprintf(outFile, strcat(' y = y + b(i)*phi(i);\n'));

fprintf(outFile, 'end');

fclose(outFile);

E.2.7 nonlcon.m

function [c,ceq] = nonlcon(inp,xi_q)

This function computes nonlinear constraints for the matlab ...

% fmincon.m . Note that this function does not return the ...

% assumes the optimizer uses finite differencing to compute the ...

% inputs:

% inp = structure containing the following information:

% inp.p = size of original design space

% inp.pc = matrix whose columns are the principal ...

% components

% inp.zmin= lower bound for z (i.e. vector of scale ...

% variables)

% inp.zmax= upper bound for z (i.e. vector of scale ...

% variables)

% xi_q = vector of reduced latent variables to be mapped back
% Outputs:
% c       = a vector of inequality constraint values
% ceq     = a vector of equality constraint values

% Check dimensions
[q,n] = size(xi_q);
if (q == 1)
    xi_q = xi_q';
    q = n;
end

% Transformation matrices
A_q = inp.pc(:,1:q);
A_r = inp.pc(:,(q+1):end);

% Map back to z
[z,xi] = map(inp,xi_q); % Solve QP problem
xi_r = xi(q+1:end);

% Exit in case of failure and return infeasible
if isempty(xi_r)
    c=ones(2*size(inp.pc,1),1);
    ceq=[];
    return
end

% Latent space inequality constraints
M_q = [A_q; -A_q];
M_r = [A_r; -A_r];
b = [inp.zmax; -inp.zmin];
c = M_q*xi_q - (b - M_r*xi_r)-1e-6;
% Equality constraints
ceq = [];

REFERENCES


