2016

Aerodynamic shape optimization by multi-fidelity modeling and manifold mapping

Jie Ren
Iowa State University

Aerodynamic shape optimization by multi-fidelity modeling and manifold mapping

by

Jie Ren

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Aerospace Engineering

Program of Study Committee:

Leifur Leifsson, Major Professor

Peng Wei

Christina L Bloebaum

Iowa State University

Ames, Iowa

2016

DEDICATION

I would like to dedicate this thesis to my mother Yalin and to my father Jianhua without whose support I would not have been able to complete this work. I would also like to thank my friends and family for their loving guidance and financial assistance during the writing of this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this thesis. First and foremost, Dr. Leifur Leifsson for his guidance, patience and support throughout this research and the writing of this thesis. His insights and words of encouragement have often inspired me and renewed my hopes for completing my graduate education. I would also like to thank my committee members for their efforts and contributions to this work: Dr. Peng Wei and Dr. Christina L Bloebaum. I would additionally like to thank Andrew Thelen, Anand Amrit, Xiaosong Du for their great ideas during our group discussion and Dr. Yonatan Afework Tesfahunegn for his guidance throughout the initial stages of my graduate career.

NOMENCLATURE

$\mathbf{A}$	Response correction matrix
$A_{baseline}$	Baseline cross sectional area [-]
A_c	Cross sectional area of low-fidelity model [-]
A_f	Cross sectional area of high-fidelity model [-]
A_{min}	Minimum cross sectional area [-]
a_l	Scalar terms of response correction matrix $\mathbf{A}$
a_d	Scalar terms of response correction matrix $\mathbf{A}$
$\mathbf{C}_d$	Drag coefficient matrix of low-fidelity model
$\mathbf{C}_l$	Lift coefficient matrix of low-fidelity model
C_d	Drag coefficient [-]
$C_{d.c}$	Drag coefficient of low-fidelity model [-]
$C_{d.f}$	Drag coefficient of high-fidelity model [-]
C_l	Lift coefficient [-]
$C_{l.c}$	Lift coefficient of low-fidelity model [-]
$C_{l.f}$	Lift coefficient of high-fidelity model [-]
C_m	Pitching moment coefficient [-]
$C_{m.c}$	Pitching moment coefficient of low-fidelity model [-]
$C_{m.f}$	Pitching moment coefficient of high-fidelity model [-]
C_p	Pressure coefficient [-]
$\mathbf{D}$	Response correction matrix
D	Drag [N]
d_l	Scalar terms of response correction matrix $\mathbf{A}$

d_d	Scalar terms of response correction matrix $\mathbf{A}$
$\mathbf{F}_d$	Drag coefficient matrix of high-fidelity model
$\mathbf{F}_l$	Lift coefficient matrix of high-fidelity model
$g(\mathbf{x})$	Inequality constraints
$h(\mathbf{x})$	Equality constraints
H	Scalar valued objective function
$\mathbf{l}$	Design variable lower bound
L	Lift [N]
M	Mach number [-]
$\mathbf{q}$	Additive response correction
$\mathbf{s}$	Surrogate model
$\mathbf{u}$	Design variable upper bound
x	Airfoil chord-wise location [-]
X	B-spline control polygon coordinates
$\mathbf{x}$	Design variable
$\mathbf{x}^*$	Optimized design variable
z	Airfoil thickness [-]
Z	B-spline control polygon coordinates
$\nabla f(\mathbf{x})$	Gradient of objective function f
z/c	Thickness to chord ratio
δ	Trust region radius [-]
ϵ_x	Argument termination criteria
ϵ_H	Objective function termination criteria
ϵ_δ	Trust region radius termination criteria
$\Delta\mathbf{F}$	Difference vector of high-fidelity model
$\Delta\mathbf{C}$	Difference vector of low-fidelity model
$\Delta\mathbf{C}^\dagger$	Pseudoinverse of $\Delta\mathbf{C}$

$\mathbf{V}_{\Delta\mathbf{C}}$	Decomposition factor of matrix $\Delta\mathbf{C}$
$\sum^{\dagger}_{\Delta\mathbf{C}}$	Decomposition factor of matrix $\Delta\mathbf{C}$
$\mathbf{U}^{T}_{\Delta\mathbf{C}}$	Decomposition factor of matrix $\Delta\mathbf{C}$
α	Angle of attack [degree]
τ	Weight factor in multipoint design [-]

ABSTRACT

Aerodynamic shape optimization (ASO) is important in contemporary engineering design of complex systems such as aircraft and wind turbines. The use of high-fidelity partial differential equation (PDE) simulations within the design process is becoming the standard. However, the overall computation cost of the ASO problem can be very high when considering the following key challenges: (1) time-consuming PDE simulations, (2) large number of design variables, and (3) conventional optimization require many system evaluations. Combined these form an optimization problem which may be prohibitive to solve, even when using high performance computing (HPC) systems. In this work, a computationally efficient optimization algorithm for aerodynamic design is presented. In our approach, direct optimization of a computationally expensive model is replaced by an iterative updating and re-optimization of a fast physics-based replacement model, following the surrogate-based optimization paradigm. The surrogate is constructed using a low-fidelity model which is corrected using manifold mapping (MM) to become a reliable representation of the high-fidelity one during the optimization process. Only one high-fidelity PDE simulation is required per design iteration. The version of MM utilized here does not require gradient information. The proposed method is validated and characterized by applying it to several benchmark ASO problems, including lift-constrained airfoil drag minimization in inviscid and viscous transonic flows, and comparing the results with state of the art techniques. MM yielded optimized shapes, with 8 B-spline design variables, that are comparable to the shapes obtained by direct optimization algorithms equipped with adjoint sensitivities and trust regions. In the inviscid benchmark case, MM needs less than 150 equivalent high-fidelity model evaluations (only flow solu-

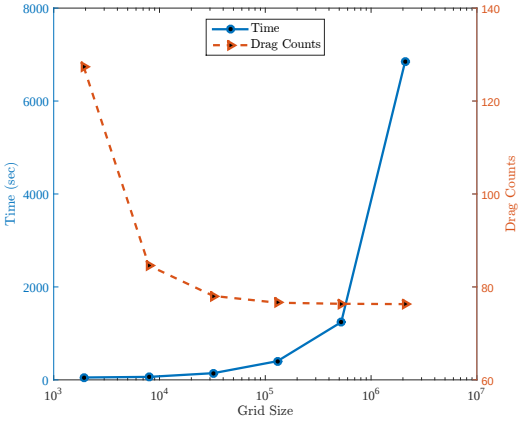
tions), or approximately 460 minutes on a HPC with 32 processors, whereas the direct algorithm needs 391 high-fidelity model evaluations (flow and adjoint), or approximately 4,494 minutes on the same HPC. In other words, the MM algorithm is an order of magnitude faster than the gradient-based search with adjoint sensitivities in this case. For the viscous case, MM yields an optimized shape using less than 300 equivalent high-fidelity evaluations, taking approximately 80 hours on the HPC. In this case, the direct algorithm is not able to reach a comparably efficient shape. MM is able to handle vector-valued design problems efficiently. This is demonstrated on a multipoint design problem as well as on an inverse design problem. The multipoint design shows that the optimized airfoil outperformed that original airfoil in terms of flight conditions and robustness in multiple cruise conditions. In the inverse design case, MM needs an order of magnitude less equivalent high-fidelity model evaluations than a derivative-free algorithm.

CHAPTER 1. INTRODUCTION

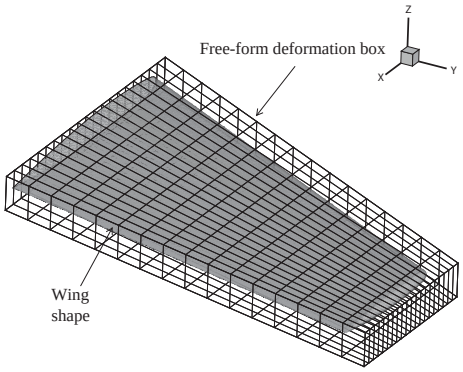
1.1 Motivation and Challenges

Aerodynamic shape optimization is important in contemporary engineering design of complex systems such as aircraft and wind turbines (see, e.g., [2] and [3]). Nowadays, the use of high-fidelity partial differential equation (PDE) simulations within the design process is becoming the standard. Typically, the main purpose with using high-fidelity PDE simulations is to capture any nonlinear physics encountered by the system. Moreover, it may be impossible to rely on prior designs when considering unconventional configurations. In addition, there may be nonlinear couplings with other disciplines within the system. For example in aircraft, there is a strong coupling between the aerodynamics and structures which needs to be resolved using accurate PDE simulations. Using experimental data will be too costly and time-consuming. Therefore, high-fidelity PDE simulations are essential in modern engineering design.

It is challenging to use high-fidelity PDE simulations as a part of the design process. Design optimization using numerical techniques is not widely used in industry. In typical engineering practices, computational models are used in hands-on parametric studies. One of the reasons for optimization not being widespread is that the sheer computational cost can be so high that performing automated design optimization in a timely manner may not be possible, even when using high performance computing (HPC). The key challenges with automated PDE-constrained design optimization can be summarized as follows:



(a) Simulation time increases quickly with grid size. Here, a grid study of the flow past the NACA 0012 airfoil using the compressible Euler equations at a Mach number of 0.85 and zero lift is shown. One drag count is defined as $\Delta C_d = 0.0001$.



(b) The optimization cost depends strongly on the number of designable parameters. The figure shows Free-form deformation (FFD) boxes for a trapezoidal wing shape. The control points are located at the intersection of the lines of the FFD box. A typical transport wing requires 300 to 700 control points [4].

Figure 1.1: Key challenges with automated PDE-constrained design optimization.

- **Time-consuming PDE simulations:** To illustrate the computational cost, let us look at two-dimensional transonic flow past an airfoil. Figure 1.1(a) shows a grid independence study of such a flow past the NACA 0012 airfoil at a fixed lift. The study shows that grids with at least million cells are needed to accurately reflect the physics. The computational cost increases rapidly with the grid size and approaches 20 minutes per PDE simulation on HPC with 32 processors. This is a simplified model of the actual fluid flow. A viscous flow simulator is more appropriate to represent the real fluid flow. The computational cost will be at least at an order of magnitude greater with such a solver, bringing the time for one PDE simulation to around 3 to 4 hours. An even realistic model would be to consider a wing shape instead of an airfoil shape. This would require a three-dimensional viscous fluid flow simulation with millions of computational cells, which would increase the cost by another order of magnitude. This brings the simulation time to around a day on HPC. Still, the model would not be complete. Components such as the fuselage,

tail, and nacelles should be included. Clearly, the computational cost will grow, rendering the use of PDE simulations within the design task even more impractical.

- **Large number of design variables:** In order to describe the aerodynamic surfaces properly and accurately, a large number of parameters may be needed. Figure 1.1(b) shows an example of a trapezoidal wing which is parameterized using the Free-form Deformation technique [5]. The number of control parameters (or design variables) needed to describe a shape like this within a design optimization task is on the order of one hundred (see, e.g., [4]). The computational cost of an optimization task depends strongly on the number of design variables as well as the size of the search space, i.e., the parameter ranges.

- **Conventional optimization techniques require many system evaluations:** Gradient-based search techniques [6] are typically used to perform automated PDE-constrained design optimization. Such techniques often need a large number of system evaluations to reach a converged design. Moreover, calculations of the gradients of the objective functions and constraints are required. Calculating the gradients is expensive. Fortunately, this cost can be reduced dramatically by using adjoint sensitivity information [7] where the gradients of a functional with respect to the design variables can be estimated by solving the adjoint equation. In general, the computational cost of one adjoint simulation is similar to the cost of one primary flow simulation. However, the adjoint simulation has be performed for each function. The use of adjoint technology is transformational for aerodynamic design as the cost of obtaining the gradients is essentially independent of the number of design variables. However, solving the performing the additional adjoint equation simulation can still be computationally expensive. Furthermore, the gradient-based search techniques still need a large number of system evaluations.

The overall computation cost of the aerodynamic shape optimization problem can be very high when considering the three challenging facts discussed here above. The combination of the high computational cost of the PDE simulations and a multi-dimensional design space constitutes a design problem which can be very impractical to solve with conventional optimization techniques [8, 3, 9, 10, 11, 12, 13], even when using HPC clusters and adjoint sensitivity information. Therefore, efficient design methodologies are needed to address this issue.

1.2 Research Contributions

The focus of this research work is in the area of aerodynamic shape optimization, and the main contributions are follows:

- An aerodynamic shape optimization technique using multi-fidelity computational fluid dynamics (CFD) models and manifold mapping (MM) [1].
- The proposed technique is implemented in a computational framework which seamlessly integrates airfoil geometry generators, computational mesh generators, and CFD solvers to create an aerodynamic shape design tool.
- The proposed technique is validated and characterized by applying it to benchmark design problems and comparing it with state of the art. The benchmark cases include two cases developed by the AIAA Aerodynamic Design Optimization Discussion Group[1] (ADODG), as well as other problems which highlight the unique features of the proposed technique, such as inverse design and multi-point design problems.
- Two different alternatives to optimize the fast multi-fidelity surrogate are investigated.

[1] https://info.aiaa.org/tac/ASG/APATC/AeroDesignOpt-DG/default.aspx

1.3 Thesis Outline

The thesis is organized as follows. Chapter 2 gives the background of aerodynamic design and aerodynamic shape optimization techniques. In Chapter 3, the optimization methodology is described, including the problem formulation, and the optimization techniques utilized in the course of this study. Chapter 4 presents the results of numerical applications of the proposed technique to transonic inviscid and viscous problems involving direct and inverse design of two-dimensional airfoil shapes. Chapter 5 concludes the thesis.

CHAPTER 2. BACKGROUND

This chapter briefly describes aerodynamic design in general, and then gives a literature review of techniques for solving aerodynamic shape optimization problems.

2.1 Aerodynamic Design

Aerodynamic design is an important part of aircraft design. Figure 2.1 shows a general flow of the aerodynamic design process. Early attention was mainly focused on maximizing lift to provide high cruise wing loading requirements of jet transport [14]. Lately, however, the focus has turned to multidisciplinary trade-offs, e.g., the speed of aircraft, ratio of lift and drag, wing loading, and specific fuel consumption of the engines. Typically, the aerodynamic designer would like to maximize VL/D (where V is the cruise speed, and L/D is the lift-to-drag ratio) considering the increment of corresponding structure weight [7]. As shown in Fig. 2.1, the inner loop of the aerodynamic design process involves mesh generation, computational fluid dynamics (CFD) analysis, visualization and performance evaluations - all of which are time-consuming [15].

Direct design is the most common approach to aerodynamic design. Here, the designer manipulates the shape of the aerodynamic surface to directly minimize (or maximize) a given figure of merit subject to a set of constraints. In wing (or airfoil) design, this is done for a given lift coefficient, Mach number, and Reynolds number. Essentially, this is a single-point, single-objective constrained nonlinear minimization problem. On the other hand, transport aircraft operate at various operating conditions due to different

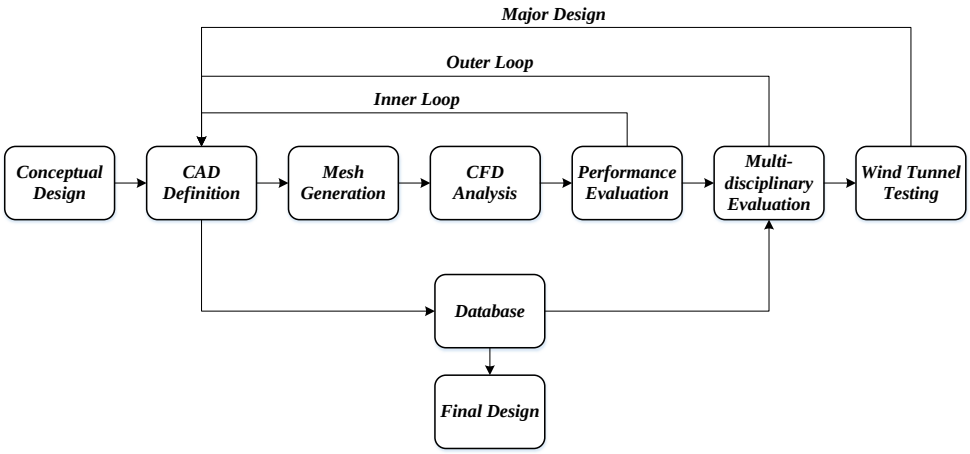


Figure 2.1: A flowchart of the aerodynamic design process.

flight environment and fuel burn requirements [16]. Near the region where single-point design performs well, efficiency may increase dramatically with small changes in operating conditions [17]. So, instead of designing for a single point the design can be performed at all the required operating points. Leading to the so-called multi-point design [18], which essentially is a multi-point, single-objective constrained nonlinear minimization problem.

Inverse design is another approach to aerodynamic design. It requires a target specification to be developed, and then the task it to minimize the difference between the target and current aerodynamic information. For example, a pressure distribution can be the target, and then the geometry yielding that target distribution is sought by iteratively changing the shape to minimize the norm of the difference between the target and current distributions [19]. The inverse design problem can be ill posed and there is no guarantee that there exists a solution. Moreover, the target distribution may no yield the best design in terms of the figures of merit. Furthermore, constructing the target distribution is not straight forward.

In both direct and inverse design, the underlying problem can be formulated as constrained nonlinear minimization. The task is to solve the minimization problem to yield the optimal aerodynamic design. This is known as aerodynamic shape optimization

(ASO). Several ASO techniques have been developed in recent years [20]. Next section gives a brief review of ASO techniques.

2.2 Aerodynamic Shape Optimization Techniques

ASO techniques can be broadly categorized into being either direct approach or surrogate based approach [21]. Direct optimization (DO) techniques include gradient-based search (e.g., steepest descent, and sequential quadratic programming [22]) and derivative-free search (e.g., heuristic algorithms, such as genetic algorithms [23], or pattern search [21]). Derivative-free approaches are very computationally intensive and are not suitable for ASO. Gradient-based search is the most widely adopted approach to solve ASO problems. ASO with gradient-based search algorithms was first attempted by Hicks and Henne back in the mid 1970's [24]. They developed numerical optimization algorithms for the design of subsonic and transonic airfoils and wings. The simulation models used in their studies were state of the art at that time, but are now considered being low-fidelity methods. The derivatives were calculated using finite difference methods. In 1988, Jameson introduced adjoint sensitivity to ASO [7]. This technology transformed ASO as it enabled the calculation of the derivatives independent of the number of design variables. Gradient-based search using adjoint sensitivities is currently the state of the art for ASO (see, e.g., [12]).

As mentioned in Chapter 1, there are several reasons why ASO of high-fidelity simulation models can be challenging. Surrogate-based optimization (SBO) [25, 26, 27] aims at addressing those challenges. The basic idea behind SBO is to replace the direct optimization of the computationally expensive model with an iterative process that involves the construction, optimization, and updating of a fast surrogate model [21]. Surrogates models can be broadly divided into two categories: data-driven surrogates, and physics-based surrogates [21].

Data-driven surrogates (also called approximation surrogates) are constructed by approximating sampled high-fidelity model data using techniques such as polynomial approximation [26], kriging, [28] and neural networks [29]. In order to enhance the surrogate at global modeling accuracy or global optimum locating accuracy, numerous methods for allocating additional training points have been developed (see, e.g., Forrester and Keane [27]). According to the work of Forrester and Keane [27], SBO with data-driven surrogates can be used as an efficient global optimization technique.

A physics-based surrogate is constructed by correcting or enhancing a low-fidelity model [30], [31]. The low-fidelity model can be obtained by either using simplified governing equations or exploiting the high-fidelity model with a coarser computational description [30, 32] and relaxed convergence criteria [32]. The low-fidelity model is subsequently corrected or enhanced to become a reliable representation of the high-fidelity model by using certain response correction methods. In the data-driven surrogate approach, a large amount of data sample is needed to ensure a decent accuracy. The physics-based surrogates contain information on the underlying physics through the low-fidelity model. Consequently, physics-based surrogates require less high-fidelity information to set up. Typically, the high-fidelity model is evaluated only once per design iteration.

Over the last few decades, various correction techniques and related optimization algorithms have been developed, including the approximation and model management optimization (AMMO) [30], multi-point correction techniques [33, 34], several variations of output space mapping (SM) [33], as well as manifold mapping (MM) [1, 35]. Apart from the aforementioned ones, which are all so-called parametric methods [31] (where the correction functions are given explicitly with the parameters usually obtained by explicit calculations or solving auxiliary linear regression problems), a number of non-parametric technique have been developed, such as the shape-preserving response prediction (SPRP) [36], adaptive response correction (ARC) [37], and adaptive response prediction (ARP) [38].

Parametric methods are preferred by many due to their simplicity. Among these, MM [1, 35] seems to be one of the most interesting because of its capability to accommodate available high-fidelity model data accumulated during the optimization run, as well as having the potential to approximately satisfy the first-order consistency with the high-fidelity model upon the algorithm convergence. Moreover, MM is capable of handling vector responses [1], making it, potentially, a good candidate for multi-point and inverse design problems. Despite its potential, MM has not yet been applied to ASO.

CHAPTER 3. OPTIMIZATION METHODOLOGY

In this chapter, the PDE-constrained optimization problem formulation is given, and the techniques to solve it are described. In particular, direct and surrogate-based methods are described. The direct methods include derivative-free and gradient-based approaches. The surrogate-based methods include data-driven and physics-based approaches.

3.1 Basic Definitions and Terminology

We start by introducing the basic terminology used throughout the thesis:

Optimization problem: In this work, the optimization problem is looking for the best solution with respect to specific data given by a set of partial differential equation (PDE) simulations.

Design variables: The design variables considered in this work are a group of numerical arguments which are allowed to change during the design optimization. In general, design variables are often bounded, i.e., they have upper and lowers bounds. Typically, design variables are categorized into boo-lean, e.g., whether or not to have a fixed lift coefficient mode in a wing design; discrete, e.g., number of distribution of control points along the airfoil of the wing; or continuous, e.g., value of displacement of control points of the wing. The solution method of the optimization problem depends to some extent on the design variable types. In this work, all the design variables are continuous.

Objective function: The objective function represents the problem that is being solved over the design variables. Conventionally, minimization and maximization problems are the most popular ones. Usually, the minimization problem is always defined in the standard form, and the maximization problem can be treated by negating the objective function. This work considers determining single-objective optimization problems of the minimization type.

Constraints: Constrained optimization problems are problems where the minimization of the objective function are subject to constraints. In general, constraints can be categorized into two types: equality constraints, and inequality constraints. The optimization problems considered in this work use both types of constraints. In general, evaluations of the constraints require evaluations of the PDE simulator.

High-fidelity model: The high-fidelity model response function is an output of a PDE simulation model which is assumed to be accurate and expensive to evaluate.

Low-fidelity model: In contrast to the high-fidelity model, a low-fidelity model is a fast simplified version of the high-fidelity one. For example, a low-fidelity model can utilize simplified physics models, coarse mesh resolution and relaxed simulation convergence criteria [39]. In all three problems considered in this work, the low-fidelity models utilize a combination of a coarse mesh resolution and relaxed simulation convergence criteria. These are called multi-resolution models [36]. However, it should be noted that the optimization methodology does not depend on the low-fidelity model type. In this work, the design variables considered in low-fidelity model are the same as used in the high-fidelity model.

3.2 Optimization Problem Formulation

The aerodynamic design problems considered in this work involve nonlinear constrained optimization of airfoil shapes in two-dimensional transonic flow. The flow simu-

lations are performed using accurate, but computationally expensive, high-fidelity computational fluid dynamics (CFD) models. The high-fidelity simulations are denoted here by $\mathbf{f}$. In general, nonlinear constrained optimization problems can be formulated as

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} H(\mathbf{f}(\mathbf{x})), \tag{3.1}$$

$$s.t. \quad \mathbf{g}(\mathbf{x}) \leq 0, \quad \mathbf{h}(\mathbf{x}) = 0, \quad \mathbf{l} \leq \mathbf{x} \leq \mathbf{u},$$

where $\mathbf{x}$ is the design variable vector of size $n \times 1$, $\mathbf{x}^*$ is the optimized design vector of size $n \times 1$, H is a scalar valued objective function, $\mathbf{f}(\mathbf{x})$ is a $m \times 1$ vector with the figures of merit, $\mathbf{g}(\mathbf{x})$ is a $p \times 1$ vector with the inequality constraints, $\mathbf{h}(\mathbf{x})$ is a $q \times 1$ vector with the equality constraints, and $\mathbf{l}$ and $\mathbf{u}$ are the design variable lower and upper bounds, respectively, both vectors of the same size as $\mathbf{x}$. The vectors $\mathbf{f}(\mathbf{x})$, $\mathbf{g}(\mathbf{x})$, and $\mathbf{h}(\mathbf{x})$ are all obtained, or derived, from the computationally expensive PDE simulations.

In aerodynamic shape optimization, the high-fidelity simulation model $\mathbf{f}$ calculates the figures of merit. For example, in the two-dimensional case, $\mathbf{f}$ can be composed as follows

$$\mathbf{f}(\mathbf{x}) = [C_{l.f}(\mathbf{x}) \quad C_{d.f}(\mathbf{x}) \quad C_{m.f}(\mathbf{x}) \quad A(\mathbf{x})]^T, \tag{3.2}$$

where $C_{l.f}(\mathbf{x})$ is the high-fidelity non-dimensional lift coefficient, $C_{d.f}(\mathbf{x})$ is the high-fidelity non-dimensional drag coefficient, $C_{m.f}(\mathbf{x})$ is the pitching moment coefficient, and $A(\mathbf{x})$ is the airfoil cross-sectional area. The subscript f denotes the high-fidelity model. In the case of a drag minimization problem, the objective function in problem (3.1) is set as

$$H(\mathbf{f}(\mathbf{x})) = C_{d.f}(\mathbf{x}). \tag{3.3}$$

The inequality constraints are set as

$$g_1(\mathbf{x}) = C_{m.min} - C_{m.f}(\mathbf{x}) \leq 0, \qquad g_2(\mathbf{x}) = A_{min} - A(\mathbf{x}) \leq 0, \tag{3.4}$$

where $C_{m.min}$ is a minimum allowable pitching moment coefficient, and A_{min} is a minimum cross-sectional area. The equality constraint is

$$h_1(\mathbf{x}) = C_{l.t} - C_{l.f}(\mathbf{x}) = 0, \tag{3.5}$$

where $C_{l.t}$ is a target lift coefficient set by the designer. Here, the angle of attack, α, is used as a dummy variable to find the target lift coefficient value. All the constraints are handled directly in the optimization process.

3.3 Direct Optimization

Direct optimization can be broadly categorized into derivative-free and gradient-based search approaches [31]. Methods of the derivative-free type are typically more costly than their gradient-based counterparts, but have the benefit of being more immune to numerical noise. The latter may be present when using CFD simulation models, especially with coarse CFD model discretization. Due to the availability of adjoint technology [7], it is possible to perform direct gradient-based optimization of computationally expensive simulation models. Using this technology, the cost of obtaining the gradients is almost equivalent to one flow solution for any number of design variables.

The flowchart of a generic direct optimization algorithm is shown in Fig. 3.1 and has the following main steps

1. Consider the initial design and evaluate the objective function in terms of the high-fidelity simulation model at this design with the constraints.

2. Update the design variables and evaluate the objective function again using the high-fidelity model. Save the current design if the constraints are met. Try the next design if not satisfied.

3. Repeat the loop until the termination condition is satisfied.

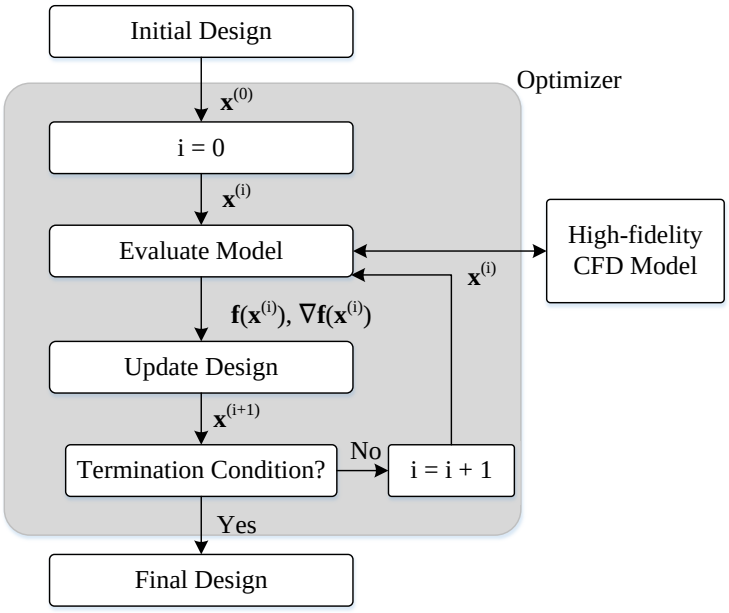


Figure 3.1: A flowchart of the direct optimization algorithm.

Note that in most cases, the high-fidelity simulation has to be evaluated in order to obtain the constraint terms. Also, obtaining the adjoint solution is an additional cost incurred during an evaluation of the high-fidelity PDE simulation.

3.3.1 Gradient-based search

Gradient-based search methods are widely used in ASO due to the availability of adjoint technology [7]. We use gradient-based search with adjoints for comparison with the optimization technique proposed in this work.

Direct gradient-based search is performed iteratively in this work as follows [21]

$$\mathbf{x}^{(i+1)} = \arg \min_{\mathbf{x}, ||\mathbf{x}-\mathbf{x}^{(i)}|| \leq \delta^{(i)}} H(\mathbf{s}^{(i)}), \tag{3.6}$$

where $\mathbf{x}^{(i)}$, $i = 0,1,...,$ is a sequence of approximate solutions to the problem (3.1), whereas $\mathbf{s}^{(i)}(\mathbf{x})$ is a linear expansion of $\mathbf{f(x)}$ at $\mathbf{x}^{(i)}$ defined as

$$\mathbf{s}^{(i)}(\mathbf{x}) = \mathbf{f}(\mathbf{x}^{(i)}) + \nabla \mathbf{f}(\mathbf{x}^{(i)}) \cdot (\mathbf{x} - \mathbf{x}^{(i)}), \tag{3.7}$$

where $\nabla \mathbf{f}(\mathbf{x}^{(i)})$ is the gradient of $\mathbf{f}$ at the reference design $x^{(i)}$. $\nabla \mathbf{f}(\mathbf{x}^{(i)})$ (applies separately for the drag and lift coefficient) is obtained by the adjoint equation [7]. The linear model (3.7) satisfies the zero- and first-order consistency conditions with the function $\mathbf{s}^{(\mathrm{i})}(\mathbf{x})$ at $\mathbf{x}^{(i)}$, i.e.,

$$\mathbf{s}^{(i)}(\mathbf{x}^{(i)}) = \mathbf{f}(\mathbf{x}^{(i)}), \tag{3.8}$$

and

$$\nabla \mathbf{s}^{(\mathrm{i})}(\mathbf{x}^{(i)}) = \nabla \mathbf{f}(\mathbf{x}^{(i)}). \tag{3.9}$$

Optimization of the linear model is constrained to the vicinity of the current design defined as $||\mathbf{x} - \mathbf{x}^{(i)}|| \leq \delta^{(i)}$, with the trust region radius $\delta^{(i)}$ adjusted adaptively using standard trust region rules [40]. The termination conditions for the algorithm (3.6) are: $(i)||\mathbf{x}^{(i)} - \mathbf{x}^{(i-1)}|| < \varepsilon_x$, $(ii)|H^{(i)} - H^{(i-1)}| < \varepsilon_H$, $(iii)\delta^{(i)} < \varepsilon_\delta$, where ε_x, ε_H, and ε_δ are user defined convergence tolerances. For direct optimization of the aerodynamics design benchmark problems given in Chapter 4, we use: $\varepsilon_x = 10^{-6}$, $\varepsilon_H = 10^{-7}$, and $\varepsilon_\delta = 10^{-6}$.

The solution of algorithm (3.6) with the model (3.7) is carried out using the MATLAB fmincon algorithm [41]. MATLAB fmincon [41] utilizes several optimization algorithms (depending on the user preference as well as the scale of the problem at hand). These include a trust region reflective algorithm where the objective function is represented using its linear or quadratic expansion models optimized using standard trust region rules [40], a sequential quadratic programming (SQP) algorithm, where the original problem is solved iteratively by replacing the original objective function (and nonlinear constraints) by their respective local quadratic models (linear for constraint functions), as well as an interior point algorithm, where the original constrained problem is replaced by a sequence of approximate minimization problems using a barrier method. In this work, we use the interior point algorithm.

3.3.2 Derivative-free search

Derivative-free search methods do not require gradient information, so it can be more immune to numerical noise than gradient-based approaches. Moreover, derivative-free methods can handle discontinuous function better. However, in most cases, derivative-free algorithms require greater amount of high-fidelity simulation evaluations to find the optimum than the gradient-based ones. Some widely used global search algorithms include genetic algorithms (GAs) [42], and particle swarm optimization [43, 44]. For local search algorithms, the pattern search algorithm [42] is widely used.

We use the pattern search algorithm [42] in this work for (i) comparison purposes, and (ii) to drive our proposed multi-fidelity algorithms (described in Section 3.6). The pattern search algorithm [42] is a stencil-based local optimization method that explores the neighborhood of the current design point. A rectangular grid (i.e., one point in each direction and in each dimension) is used in our implementation. The search process utilizes grid-constrained line search with the search direction determined using the objective function gradient estimated from perturbed designs. In case of a failure, the best perturbation (if better than the current design) is selected. Finally, the grid is refined in case the poll step does not lead to an improved design. The poll stage of the pattern search process is illustrated in Fig. 3.2.

3.4 Surrogate-based Optimization

This section describes the basis of surrogate-based optimization (SBO) [26], and approaches to construct surrogate models. The surrogates that are used in this work are emphasized.

The basic idea of SBO is that the computation of the expensive high-fidelity simulation model is replaced by a series of fast surrogate models. These surrogate models should be accurate enough to represent the high-fidelity model. During the optimization

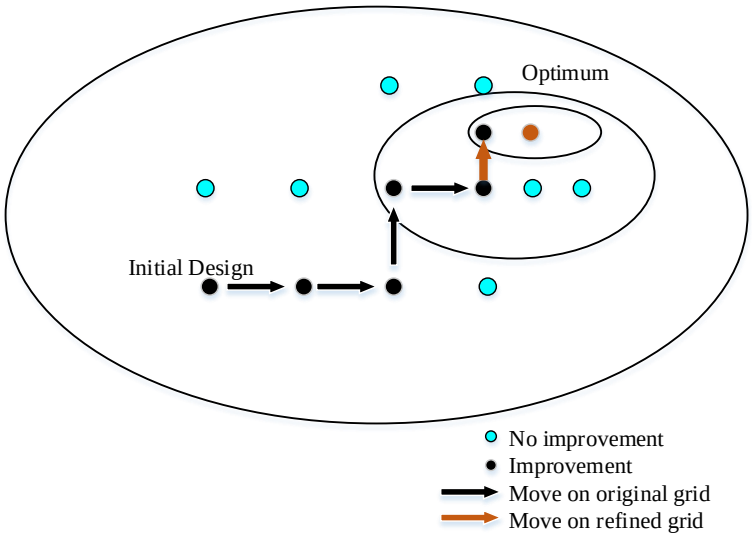


Figure 3.2: A schematic of the poll stage of the pattern search algorithm.

process, the surrogate models should be corrected by evaluating high-fidelity models. In this type of approach, most of the computation is performed by the surrogate models. As mentioned in Chapter 1, SBO can handle problems where the simulation costs are high and sensitivity information is not easily obtained. Even in some cases where it is easy to obtain derivative information cheaply, numerical noise still can still make the optimization process challenging [45]. SBO aims at solving these types of problems, as well as reducing the computational cost. A well-known approach for generating surrogate model is to sample the design space using design of experiments (DOE) [46, 47, 48] techniques, then evaluate the high-fidelity simulation at those designs and, subsequently, generate an approximation of the high-fidelity data. The SBO algorithm updates approximation by adding new high-fidelity information as it becomes available.

A flowchart of SBO in Fig. 3.3 shows that the SBO process can be represented as [31]

$$\mathbf{x}^{(i+1)} = \arg\min_{\mathbf{x}} \mathbf{s}^{(i)}(\mathbf{x}), \tag{3.10}$$

where $\mathbf{x}^{(i)}, i = 0, 1, ...,$ is a sequence of approximate solutions to the original problem (3.1), whereas $\mathbf{s}^{(i)}$ is the surrogate model at iteration i. The surrogate model should be

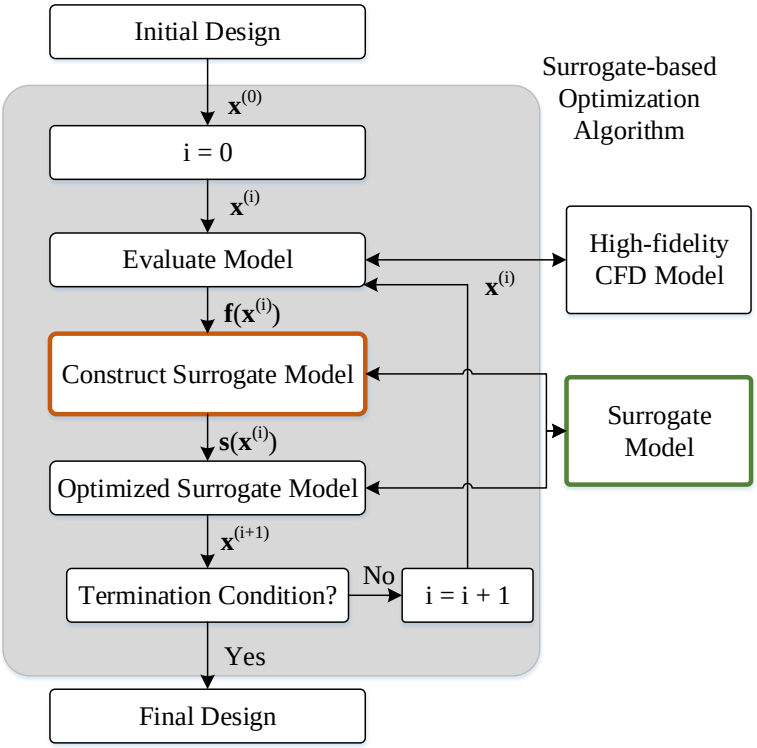


Figure 3.3: A flowchart of a generic surrogate-based optimization algorithm.

cheap and accurate enough to represent high-fidelity model, at least around the current design point, in the design space. Normally, the information from the high-fidelity model evaluation will be used in the next design iteration. Here, the computation time of the surrogate model is very short. This may make the SBO process faster than direct optimization methods.

The surrogate model is the most important part of SBO process as it replaces the high-fidelity model in the optimization process. Surrogate models can be constructed in two ways: using data-driven methods, or physics-based methods. We describe the two types in the next two sections.

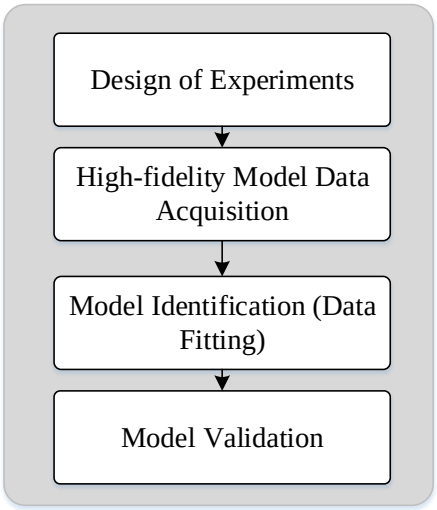


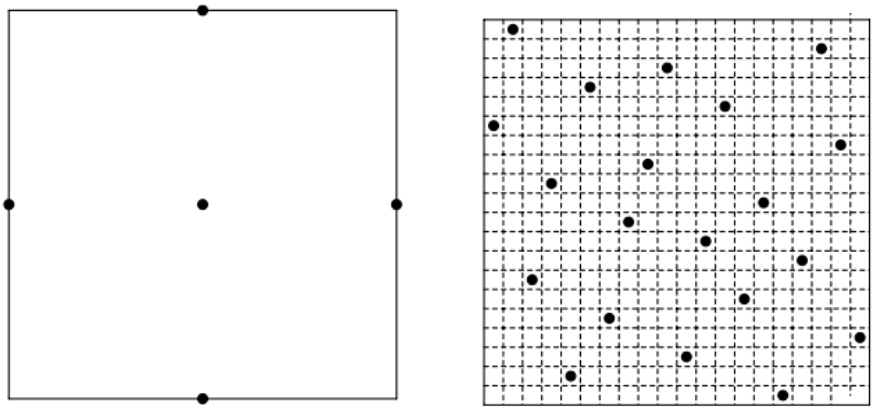
Figure 3.4: The general data-driven surrogate model construction process.

3.5 Data-driven Techniques

Data-driven surrogate models, also called approximation-based models, are constructed using the high-fidelity simulation model evaluated at selected sampling points in the design space. The process of constructing data-driven surrogate models is shown in Fig. 3.4. The first step is the DOE which is sampling training points in the design space. After sampling, the high-fidelity model is used to evaluate the training points. Fitting those points to a response surface can be achieved by two primary approaches: using explicit formulas by solving an appropriate regression problem [26], and solving a separate minimization problem [28]. To verify the accuracy of the surrogate model, different sets of data are sampled and evaluated to test the surrogate.

3.5.1 Design of experiment

DOE [46, 47, 48] are a set of techniques used to sample points in the design space. Generally, the more sampling points, the more information about system can be obtained. However, more sampling points require more evaluations of the high-fidelity model. Uniform sampling plan is the most popular one [49]. Factorial design [50] samples the points



(a) A star distribution is a simple factorial design

(b) Latin Hypercube Sampling

Figure 3.5: Sampling points in design space.

at the centers and corners (extreme points) in design space. A star distribution [51] is a simple version of factorial design. Figure 3.5(a) shows a star sampling distribution for the two dimensional case. A widely used DOE for sampling is the Latin Hypercube Sampling (LHS) [51]. The idea of LHS is to use bins to sample the points along each design variable dimension. For example, as shown in Fig. 3.5(b), if the range of each design variable is split into 20 bins, for the two-dimensional case, there are 20^2 cells in the design space. The samples are allocated randomly so that for each dimension bin there is only one sample inside. In some cases, LHS might not be uniformly sampled, for example, along the diagonal in Fig. 3.5(b). Ye [52], Palmer and Tsui [53], Beachkofski and Grandhi [54], Leary [55] developed some techniques to overcome the lack of uniformity of LHS. Other sampling plans include orthogonal array sampling [26], quasi-Monte Carlo sampling [50], and Hammersley sampling [50].

3.5.2 Approximation techniques

After sampling the points with DOE strategies, the next step is to use approximation methods to construct the surrogate models. Polynomial regression is a simple and widely used approach to generate surrogate model [26]. Other techniques available for setting up

data-driven surrogate models include Kriging [56], radial basis functions [57, 27], neural networks [58, 31]. Kriging, or Gaussian process regression might be the most widely used data-driven surrogate model in many different fields, especially, for ASO. Different from polynomial regression, Kriging [56] is to predict the value of the high-fidelity model at a given point by calculating a weighted average of the known values of the function in the neighborhood of the point.

3.5.3 Model validation

A reliable optimum can only be obtained if the surrogate model is constructed accurately enough. The split-sample method [26] is probably the most popular way to implement the model validation. Two subsets of the available data sample is utilized. The first one is the original set which is used to construct the surrogate model. The second subset contains the test points considered for the model validation purpose. The quality evaluation by this approach generally depends upon how the sample data is partitioned. It is not guaranteed the samples would provide the best use since it is only part of the samples. Cross-validation is another very popular accurate evaluation technique [59, 26]. In this approach, N subsets are used as test set for the surrogate model constructed by the rest of other $N-1$ subsets. The overall error can be estimated with all the N error evaluations obtained in this way. Compared with split-sample method, this approach is less biased. The drawback of this method is also very obvious: the surrogate model has to be generated multiple times, which brings another advantages of robustness since every point has been evaluated many times in both sampling and testing process. Bootstrapping seems to work better than cross-validation approach [60]. By using the easiest form, sampling with replacement from the full sample is considered. It takes the sample data set as a population so that samples can be drawn from it. The new random sample size is still the same with the original sample size. This approach can also be used in identifying confidence intervals [26].

3.6 Physics-based Techniques

Physics-based surrogates, also referred to as multi-fidelity surrogates, are constructed by correcting/enhancing physics-based low-fidelity models [31, 30, 61]. A low-fidelity model, or a simplified description of the system under consideration, can be obtained by neglecting certain physical or second-order effects, using simplified equations, or by exploiting the high-fidelity model with a coarser computational description [30]. The low-fidelity models are subsequently corrected or enhanced to become a reliable representation of the high-fidelity model by specific methods such as multiplicative or additive corrections [31], space mapping (SM) [33, 34], adaptive response correction (ARC) [37], adaptive response prediction (ARP) [62], or shape-preserving response prediction (SPRP) [36].

Data-driven surrogates can be the basis of efficient global optimization techniques [27]. However, to ensure decent accuracy, data-driven surrogates require a large number of data samples. Moreover, the number of samples grows quickly with the problem dimensionality. Although the physics-based surrogate models are not as versatile as the data-driven ones, they have the potential to offer significantly better efficiency in terms of the computational cost [31, 30, 61]. Many multi-fidelity algorithms require only a single high-fidelity model evaluation per design iteration [31]. Consequently, physic-based surrogates may exhibit better generalization capability than the approximation ones.

3.6.1 Multi-fidelity optimization algorithm

A generic multi-fidelity optimization algorithm with trust regions produces a sequence $\mathbf{x}^{(i)}, i = 0, 1, ...,$ of approximate solutions to the original problem (3.1) using a surrogate model $\mathbf{s}^{(i)}(\mathbf{x})$ at each iteration i of algorithm (3.10) [31]. The surrogate model $\mathbf{s}$ is a suitably corrected low-fidelity model $\mathbf{c}$. The key component of the multi-fidelity optimization algorithm is the physics-based low-fidelity (or coarse) model $\mathbf{c}$ that embeds

certain knowledge about the system under consideration, and allows us to construct a reliable surrogate using a limited amount of high-fidelity model data. In this work, the low-fidelity model is evaluated using the same CFD solver as the high-fidelity model **f**. Two (parametric) correction methods are considered: SM [34] (for the sake of comparison and validation) and manifold mapping (MM) [63, 35].

3.6.2 Space mapping

The SM technique was originally developed for microwave engineering design applications [25, 64], but is now widely used in many other engineering fields [65, 66]. In this work, multi-point output space mapping (OSM) is used to enhance the low-fidelity transonic airfoil model. Multi-point OSM constructs an aerodynamic surrogate model as follows [34]

$$\mathbf{s}^{(i)}(\mathbf{x}) = \mathbf{A}^{(i)} \circ \mathbf{c}(\mathbf{x}) + \mathbf{D}^{(i)} + \mathbf{q}^{(i)} = [a_l^{(i)} C_{l.c}(\mathbf{x}) + d_l^{(i)} + q_l^{(i)} \quad a_d^{(i)} C_{d.c}(\mathbf{x}) + d_d^{(i)} + q_d^{(i)} \quad A_c(\mathbf{x})]^T, \tag{3.11}$$

where $\circ$ denotes component-wise multiplication. Note that there is no need to map A_c because

$$A_c(\mathbf{x}) = A_f(\mathbf{x}) \tag{3.12}$$

for all $\mathbf{x}$. The response correction parameters $\mathbf{A}^{(i)}$ and $\mathbf{D}^{(i)}$ are obtained by solving

$$[\mathbf{A}^{(i)}, \mathbf{D}^{(i)}] = \arg\min_{[\mathbf{A},\mathbf{D}]} \sum_{k=0}^{i} \left\| \mathbf{f}(\mathbf{x}^{(k)}) - (\mathbf{A} \circ \mathbf{c}(\mathbf{x}^{(k)}) + \mathbf{D}) \right\|^2, \tag{3.13}$$

i.e., the response scaling is supposed to (globally) improve the matching for all previous iteration points. The additive response correction term $\mathbf{q}^{(i)}$ is defined as

$$\mathbf{q}^{(i)} = \mathbf{f}(\mathbf{x}^{(i)}) - [\mathbf{A}^{(i)} \circ \mathbf{c}(\mathbf{x}^{(i)}) + \mathbf{D}^{(i)}], \tag{3.14}$$

i.e., it ensures perfect matching (zero-order consistency) between the surrogate and the high-fidelity model at the current design $\mathbf{x}^{(i)}$, i.e., $\mathbf{s}^{(i)}(\mathbf{x}^{(i)}) = \mathbf{f}(\mathbf{x}^{(i)})$.

Fortunately, $\mathbf{A}^{(i)}$ and $\mathbf{D}^{(i)}$ can be obtained analytically [34] as

$$\begin{bmatrix} a_l^{(i)} \\ d_l^{(i)} \end{bmatrix} = (\mathbf{C}_l^T \mathbf{C}_l)^{-1} \mathbf{C}_l^T \mathbf{F}_l, \tag{3.15}$$

and

$$\begin{bmatrix} a_d^{(i)} \\ d_d^{(i)} \end{bmatrix} = (\mathbf{C}_d^T \mathbf{C}_d)^{-1} \mathbf{C}_d^T \mathbf{F}_d, \tag{3.16}$$

where

$$\mathbf{C}_l = \begin{bmatrix} C_{l.c}(\mathbf{x}^{(0)}) & C_{l.c}(\mathbf{x}^{(1)}) & ... & C_{l.c}(\mathbf{x}^{(i)}) \\ 1 & 1 & ... & 1 \end{bmatrix}, \tag{3.17}$$

$$\mathbf{F}_l = \begin{bmatrix} C_{l.f}(\mathbf{x}^{(0)}) & C_{l.f}(\mathbf{x}^{(1)}) & ... & C_{l.f}(\mathbf{x}^{(i)}) \\ 1 & 1 & ... & 1 \end{bmatrix}, \tag{3.18}$$

and

$$\mathbf{C}_d = \begin{bmatrix} C_{d.c}(\mathbf{x}^{(0)}) & C_{d.c}(\mathbf{x}^{(1)}) & ... & C_{d.c}(\mathbf{x}^{(i)}) \\ 1 & 1 & ... & 1 \end{bmatrix}, \tag{3.19}$$

$$\mathbf{F}_d = \begin{bmatrix} C_{d.f}(\mathbf{x}^{(0)}) & C_{d.f}(\mathbf{x}^{(1)}) & ... & C_{d.f}(\mathbf{x}^{(i)}) \\ 1 & 1 & ... & 1 \end{bmatrix}, \tag{3.20}$$

which is a least-square optimal solution to the linear regression problem

$$\mathbf{C}_l a_l^{(i)} + d_l^{(i)} = \mathbf{F}_l \tag{3.21}$$

and

$$\mathbf{C}_d a_d^{(i)} + d_d^{(i)} = \mathbf{F}_d, \tag{3.22}$$

which is equivalent to (3.13). Note that the matrices $\mathbf{C}_l^T \mathbf{C}_l$ and $\mathbf{C}_d^T \mathbf{C}_d$ are non-singular for $i > 1$. For $i = 1$, only the multiplicative correction $\mathbf{A}^{(i)}$ is used, which can be calculated in a similar way.

The solution of algorithm (3.6) with the surrogate model (3.11) is carried out using the pattern search algorithm [62] and the termination conditions are the same as for the direct optimization algorithm (Section 3.3), but with $\epsilon_x = 10^{-3}$, $\epsilon_H = 10^{-4}$, $\epsilon_\delta = 10^{-3}$.

3.6.3 Manifold mapping

MM [67, 1] is a response correction technique that is capable of comprehensive exploitation of available high-fidelity model data. In its basic version, the MM surrogate model is defined [1] as

$$\mathbf{s}^{(i)}(\mathbf{x}) = \mathbf{f}(\mathbf{x}^{(i)}) + \mathbf{S}^{(i)}(\mathbf{c}(\mathbf{x}) - \mathbf{c}(\mathbf{x}^{(i)}), \tag{3.23}$$

with $\mathbf{S}^{(i)}$ being a 3×3 correction matrix in our case (in general, the size of $\mathbf{S}$ is equal to the number of the components in $\mathbf{c}, \mathbf{f}$, and $\mathbf{s}$) defined as

$$\mathbf{S}^{(i)} = \Delta\mathbf{F} \cdot \Delta\mathbf{C}^{\dagger}, \tag{3.24}$$

where

$$\Delta\mathbf{F} = [\mathbf{f}(\mathbf{x}^{(i)}) - \mathbf{f}(\mathbf{x}^{(i-1)}) \quad \ldots \quad \mathbf{f}(\mathbf{x}^{(i)}) - \mathbf{f}(\mathbf{x}^{(max\{i-n,0\})})], \tag{3.25}$$

and

$$\Delta\mathbf{C} = [\mathbf{c}(\mathbf{x}^{(i)}) - \mathbf{c}(\mathbf{x}^{(i-1)}) \quad \ldots \quad \mathbf{c}(\mathbf{x}^{(i)}) - \mathbf{c}(\mathbf{x}^{(max\{i-n,0\})})]. \tag{3.26}$$

The pseudo-inverse, denoted by †, is defined as

$$\Delta\mathbf{C}^{\dagger} = \mathbf{V}_{\Delta\mathbf{C}} \sum\nolimits_{\Delta\mathbf{C}}^{\dagger} \mathbf{U}_{\Delta\mathbf{C}}^{T}, \tag{3.27}$$

where $\mathbf{U}_{\Delta\mathbf{C}}$, $\sum_{\Delta\mathbf{C}}$, and $\mathbf{V}_{\Delta\mathbf{C}}$ are the factors in the singular value decomposition of the matrix $\Delta\mathbf{C}$. The matrix $\sum_{\Delta\mathbf{C}}^{\dagger}$ is the result of inverting the nonzero entries in $\sum_{\Delta\mathbf{C}}$, leaving the zeros invariant [67].

The MM model alignment is illustrated in Fig. 3.6. Point $\mathbf{x}_c^*$ denotes the minimizer corresponding to the low-fidelity model response, and the point y is the vector of design specifications. Solid and dashed lines denote the tangent planes for the high- and low-fidelity model responses at their optimal designs. Upon convergence, the linear correction $\mathbf{S}^*$ (being the limit of $\mathbf{S}^{(i)}$ with $i \to \infty$) maps the point $\mathbf{c}(\mathbf{x}^*)$ to $\mathbf{f}(\mathbf{x}^*)$, and the tangent plane for $\mathbf{c}(\mathbf{x})$ at $\mathbf{c}(\mathbf{x}^*)$ to the tangent plane for $\mathbf{f}(\mathbf{x})$ at $\mathbf{f}(\mathbf{x}^*)$.

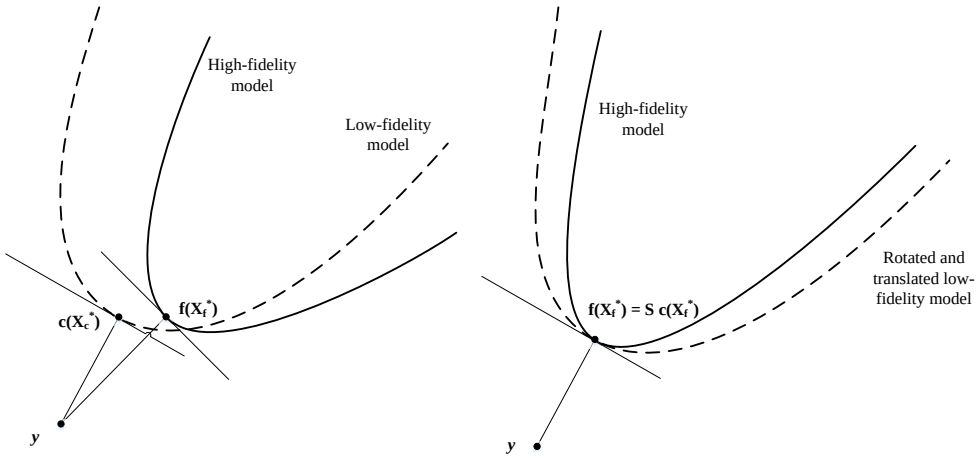


Figure 3.6: A conceptual illustration of the manifold mapping model alignment (adopted and reproduced from Echeverria [1]).

It should be noted that although MM does not explicitly use sensitivity information, the surrogate and the high-fidelity model Jacobians become more and more similar to each other towards the end of the MM optimization process (i.e., when $\|\mathbf{x}^{(i)}-\mathbf{x}^{(i-1)}\| \to 0$) so that the surrogate (approximately) satisfies both the zero- and first-order consistency conditions with $\mathbf{f}$. This allows for a more precise identification of the high-fidelity model optimum. On the other hand, the correction matrix $\mathbf{S}^{(i)}$ can be defined using exact Jacobians of the low- and high-fidelity models if available.

The solution of algorithm (3.6) with the surrogate model (3.23) is carried out using the pattern search algorithm [62], and the termination conditions are the same as SM algorithm. In the specific case of airfoils, where the model response vectors consist of three components (lift, drag, and cross-section area), the correction matrix $\mathbf{S}$ should be identity with respect to the third components (the area is identical for both the low- and high-fidelity model).

CHAPTER 4. NUMERICAL APPLICATIONS

In this chapter, the direct and multi-fidelity optimization algorithms are applied to two benchmark aerodynamic design problems involving inviscid and viscous transonic flow past airfoil shapes. These benchmark cases were developed by the AIAA Aerodynamic Design Optimization Discussion Group[1] (ADODG). A part of these results were presented at AIAA SciTech 2016 Conference [68, 69]. The first two-dimensional case, Benchmark Case I (BC I), is drag minimization of NACA 0012 in inviscid flow at zero lift, and the second case, Benchmark Case II (BC II), is the lift-constrained drag minimization of RAE 2822 in viscous flow. First, BC I, high-fidelity models are presented where Euler Equation of the flow is introduced. B-spline curves [70] is used to parameterize geometry. Then half region O-mesh has been generated in Pointwise [71] (since the airfoil is symmetric and the angle of attack is zero) and the compressible Euler equations has been solved by Stanford University Unstructured (SU^2) [72]. This improves the flow solutions and guarantees that the lift coefficient is equal to zero (which previously was hard to obtain due to highly nonlinear flow features). In BC II, a C-mesh has been constructed and the Reynolds-averaged Navier-Stokes (RANS) equations and the Spalart-Allmaras [73] turbulence model are solved by SU^2 solver. Grid studies of the benchmark cases are presented to verify that the computational grid satisfies the convergence criteria where the drag count should be converged within 0.1 and 1 counts for BC I and BC II, respectively. The RANS model has been validated by RAE 2822 airfoil using the RAE 2822 airfoil by comparing with experimental data. Both benchmark cases

[1] https://info.aiaa.org/tac/ASG/APATC/AeroDesignOpt-DG/default.aspx

are solved using multi-fidelity optimization with (space mapping) SM [34] and manifold mapping (MM) [63, 35]. Moreover, for comparison purposes, we solve both benchmark cases with a gradient-based technique with adjoints and trust regions [21]. Additionally, BC I is solved using pattern search [21, 62] (a derivative-free direct optimization technique). The optimization formulation are also explained and introduced as well. Results are compared in terms of the design quality as well as the number of model evaluations and total optimization time.

In order to improve robustness of optimized airfoil, multipoint design, such as done in [17], is explored by using MM. The optimization involves the minimization of the weighted drag coefficients at each operating points subject to constraints. Different cases of multipoint optimization are performed and analyzed. It turns out that MM is well suited for such cases as it can handle vector responses.

As mentioned in Chapter 2, inverse design is a different approach to perform aerodynamic design. Direct design approach requires the cost function (usually the drag coefficient) to be defined with constraints, and then finds the solution using mathematical algorithms with consideration of constraints sequentially [19]. In this work, we perform inverse design using MM to optimize airfoil geometry shape to reach to a target pressure distribution. Manifold mapping is also well suited for inverse design since it involves a vector responses.

4.1 Benchmark Case I: Drag Minimization of the NACA 0012 Airfoil in Transonic Inviscid Flow

The multi-fidelity optimization algorithms in Chapter 3 are applied to BC I involving inviscid transonic flow past NACA 0012 airfoil shape. The optimization results are compared with pattern search and gradient-based search in terms of the design quality as well as the amount of model evaluations and total optimization time.

4.1.1 Problem definition

The objective is to minimize the drag coefficient (C_d) of the modified NACA 0012 airfoil section at a free-stream Mach number of $M_\infty = 0.85$ and an angle of attack $\alpha = 0$ deg. subject to a minimum thickness constraint. The optimization problem is stated as

$$\min_{\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}} C_d, \tag{4.1}$$

where $\mathbf{x}$ is the vector of design variables, and $\mathbf{l}$ and $\mathbf{u}$ are the lower and upper bounds, respectively. The thickness constraint is stated as

$$z(x) \geq z(x)_{baseline}, \tag{4.2}$$

where $z(x)$ is the airfoil thickness, $x \in [0, 1]$ is the chord-wise location, and $z(x)_{baseline}$ is the thickness of the baseline airfoil, which is a modified version of the NACA 0012, defined as

$$z(x)_{baseline} = \pm 0.6(0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1036x^4). \tag{4.3}$$

In our implementation, the objective function is set as

$$H(\mathbf{f}(\mathbf{x})) = C_d(\mathbf{x})), \tag{4.4}$$

and the thickness constraint is handled directly. The design variables and their bounds are described as follows.

4.1.2 Design variables

The airfoil shape design variables are defined by the parametrization method. In this work, we use B-spline curves [70] for the shape parametrization. The airfoil surfaces are written in parametric form as

$$x(t) = \sum_{i=1}^{n+1} X_i N_{i,k}(t), \tag{4.5}$$

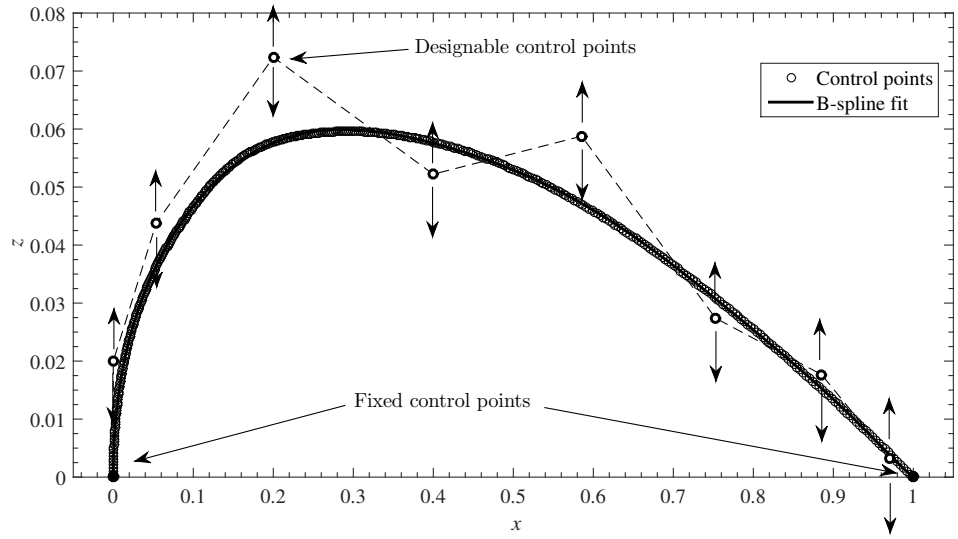


Figure 4.1: B-spline parameterization for the upper surface of the airfoil.

and

$$z(t) = \sum_{i=1}^{n+1} Z_i N_{i,k}(t), \tag{4.6}$$

where (x, z) are the Cartesian coordinates of the surface, $N_{i,k}$ is the B-spline basis function of order k, (X_i, Z_i) are the coordinates of the B-spline control polygon, and $n+1$ is the total number of control points. Note that the surface description with (4.5) and (4.6) is continuous. The control points are used as design variables and allowed only to move freely vertically as shown in Fig. 4.1 (in this figure we only show the upper surface of the airfoil). Each designable control point is free to move in the vertical direction only. Thus, we have $\mathbf{x} = [Z_1 \quad Z_2 \quad \dots \quad Z_{n+1}]^T$ and the corresponding X_i are fixed during the optimization.

In BC I, we use 10 control points, as shown in Fig. 4.1, where two are fixed at the leading- and trailing-edges, and the other ones can move in the vertical direction, yielding 8 design variables in total. Based on a fit to the modified NACA 0012 of (4.3), we fix the x-locations of the free control points as

$$X = [0 \quad 0.0536 \quad 0.2000 \quad 0.4000 \quad 0.5854 \quad 0.7527 \quad 0.8854 \quad 0.9706]^T.$$

The initial design variable vector is

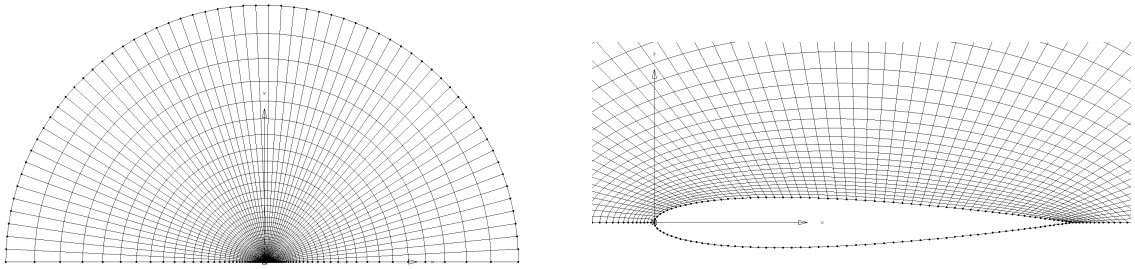
$$\mathbf{x} = [0.0185 \quad 0.0474 \quad 0.0654 \quad 0.0633 \quad 0.0486 \quad 0.0323 \quad 0.0161 \quad 0.0043]^T.$$

The lower bound of $\mathbf{x}$ is set as zero, i.e., $\mathbf{l} = \mathbf{0}$, and the upper bound is set as one, i.e., $\mathbf{u} = \mathbf{1}$.

4.1.3 High-fidelity CFD model

The Stanford University Unstructured (SU^2) computer code [72] is utilized for the inviscid fluid flow simulations. The steady compressible Euler equations are solved with an implicit density-based formulation. The convective fluxes are calculated using the second order Jameson-Schmidt-Turkel (JST) scheme [74]. Three multi-grid levels are used for solution acceleration. Asymptotic convergence to a steady state solution is obtained in each case. The flow solver convergence criterion is the one that occurs first of the two: (i) the change in the drag coefficient value over the last 100 iterations is less than 10^{-4}, or (ii) a maximum number of iterations of 1,000 is met.

An O-type computational mesh is generated using Pointwise [71] (see Fig. 4.2). Since the airfoil is symmetrical and the angle of attack is fixed at zero, only the half-plane is considered. The far-field boundary is set 55 chord lengths away from the airfoil surface. The mesh density is controlled by the number of cells on the airfoil surface and the number of cells normal to the surface. The results of a grid convergence study, given in Table 4.1, revealed that a 512×512 mesh (shown number 5 in the table) is required for convergence within 0.1 drag count (1 drag count is defined $\Delta C_d = 10^{-4}$) when compared with the next mesh. Distance to the first grid point is 0.0015c where c is the airfoil chord length. The flow simulation for Mesh 5 takes about 4.2 minutes. An adjoint solution for the drag coefficient take approximately the same amount of time. It should be noted that throughout an optimization run the airfoil shape may change significantly and the flow and adjoint simulation times may vary depending on the particular shape.



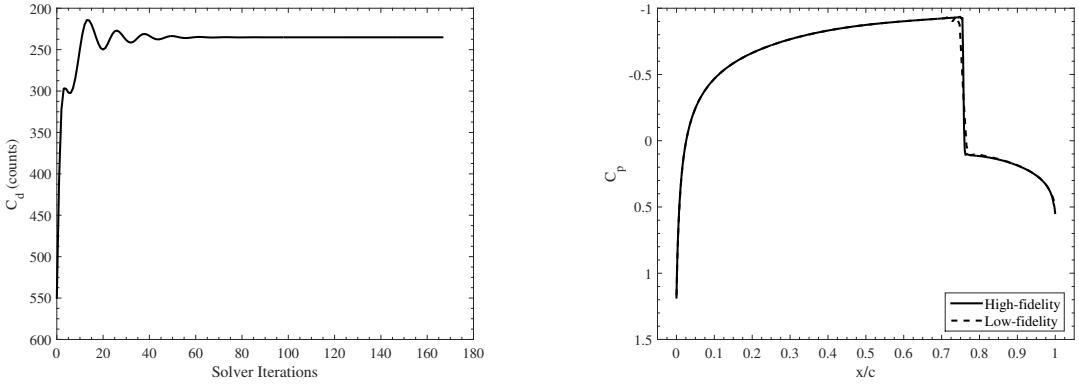
(a) Far-field (b) Surface

Figure 4.2: Mesh views of half-plane O-mesh used in the inviscid model.

Table 4.1: Grid convergence study for the baseline shape of BC I at $M_\infty = 0.85$ and $\alpha = 0.0^o$.

Mesh	**Grid Size**	$\mathbf{C}_l$	$\mathbf{C}_l$	**Simulation Time** * **(min)**
1	32×32 (961)	0.0	496.7194	0.29
2	64×64 (3969)	0.0	480.7083	0.41
3	128×128 (16129)	0.0	470.2201	0.71
4	256×256 (65025)	0.0	469.3722	1.72
5	512×512 (261121)	0.0	468.4470	4.24
6	1024×1024 (1046529)	0.0	468.3714	24.60

*Computed on a high-performance cluster with 32 processors. Flow solution only.



(a) Low-fidelity model

(b) High- and low-fidelity models

Figure 4.3: Solver convergence history of the high- and low-fidelity model.

For the optimization studies, Mesh 5 is used as the high-fidelity model $\mathbf{f}$, and Mesh 3 as the low-fidelity model $\mathbf{c}$ (used only for SM and MM). For the low-fidelity model, the maximum number of solver iterations is set to 300. Figure 4.3 shows the solver convergence of the low-fidelity model, and it also gives a comparison of the low- and the high-fidelity models. The comparison indicates that the low-fidelity model is a relatively good representation of the high-fidelity one.

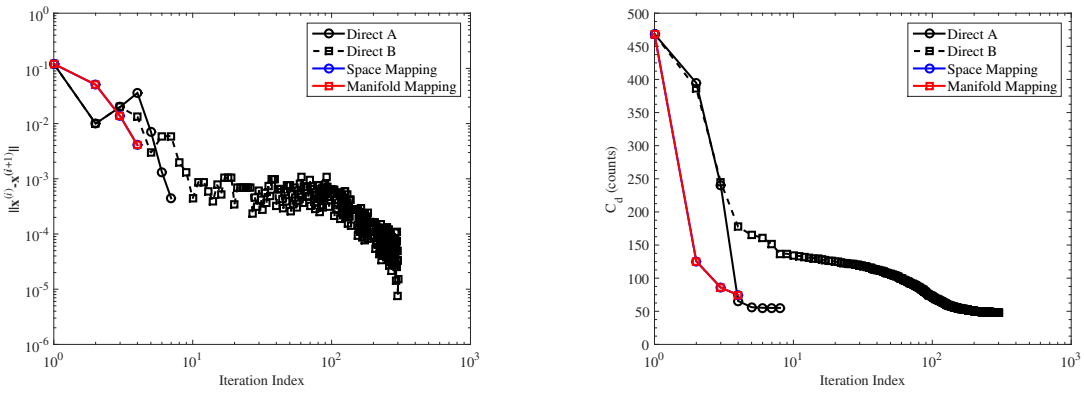
4.1.4 Results

BC I is solved using the optimization algorithms listed in Table 4.2. Figure 4.4 shows the convergence of the argument $\mathbf{x}$, and it also shows the evolution of the objective function H (the drag coefficient in this case). The direct algorithms A and B terminate on the objective function, whereas the SM and MM algorithms terminate on the argument.

In terms of design quality, Direct B obtains the lowest drag coefficient value of 47.7 counts (Table 4.3). Direct A obtains a drag coefficient value of 55.1 counts, whereas the SM and MM algorithms converge to the same design with a drag coefficient value of 74.5 counts. Although, there is a significant difference in the drag coefficient values,

Table 4.2: Details of the optimization algorithms used for BC I.

Algorithm	**Driver**	**Mesh for f**	**Mesh for c**	**Adjoints**	**Trust Region**
Direct A	Pattern Search	5	N/A	N/A	N/A
Direct B	MATLAB's fmincon	5	N/A	Yes	Yes
Space Mapping	Pattern search	5	3	No	Yes
Manifold Mapping	Pattern Search	5	3	No	Yes



(a) Argument $\mathbf{x}$ (b) Objective function H

Figure 4.4: BC I convergence history.

Table 4.3: Optimization results for BC I.

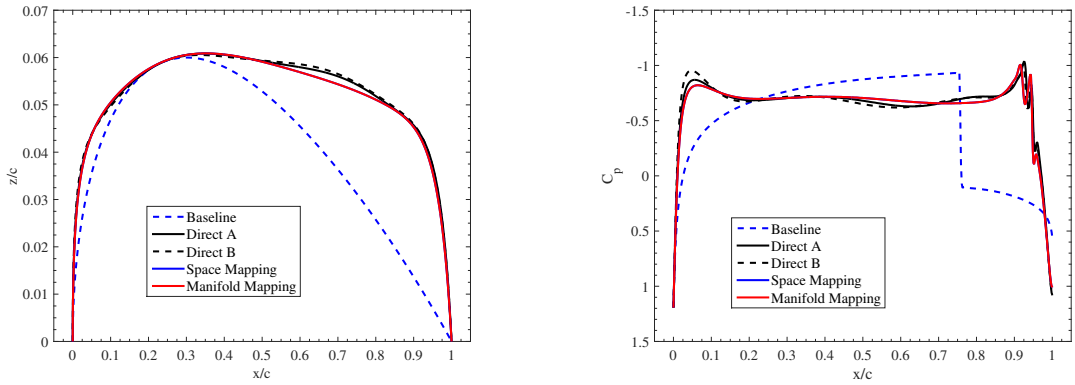
Parameter/Method	**Baseline**	**Direct A**	**Direct B**	**SM**	**MM**
$C_l(l.c.)$	0.0	0.0	0.0	0.0	0.0
$C_d(d.c.)$	468.45	55.10	47.68	74.53	74.53
N_c	-	-	-	778	778
N_f	-	1,383	391*	4	4
N_{equ}	-	1,383	391*	≈ 150	≈ 150
$t_c(min)$	-	-	-	447.0	445.8
$t_f(min)$	-	6,939	4,494	12.3	13.1
$t_{tot}(min)$	-	6,939	4,494	459.3	458.9

* Primary flow solutions as well as adjoint solutions.

the overall shapes look very similar (see Fig. 4.5). However, there is a slight difference between the shapes obtained by the direct algorithms and the multi-fidelity algorithms. In particular, the shapes obtained by the direct algorithms have a slightly fuller shape between $x/c = 0.6$ to $x/c = 0.8$ than the shapes obtained by the multi-fidelity algorithms. Other parts of the shapes are comparable.

In terms of the pressure coefficient distributions, shown in Fig. 4.5, the shape obtained by Direct B has a distinctly lower peak near the leading-edge of the airfoil compared to the others. All the shapes have reduced the shock strength significantly when compared to the baseline pressure distribution. Figure 4.6 shows the pressure coefficient contours for all the shapes, including the baseline. A grid convergence study of the optimized shape obtained by MM indicates that the high-fidelity grid used in the optimization run (the 512×512 grid) is converged within 0.3 drag counts. This indicates that a finer mesh may have to be used for the high-fidelity model in order to ensure a drag coefficient resolution of less than 0.1 drag counts.

In terms of computational cost, the multi-fidelity algorithms need the least amount of time. In particular, they need 4 high-fidelity model evaluations (N_f), and 778 low-fidelity model evaluations (N_c); which, in total, is equivalent to less than 150 high-fidelity model evaluations (N_{tot}). The total optimization time is about 459 minutes. Direct B needs 391 high-fidelity model evaluations (includes both flow and adjoint solutions) and the total



(a) Shapes (b) Pressure coefficient distributions

Figure 4.5: Comparison of BC I baseline and optimized characteristics.

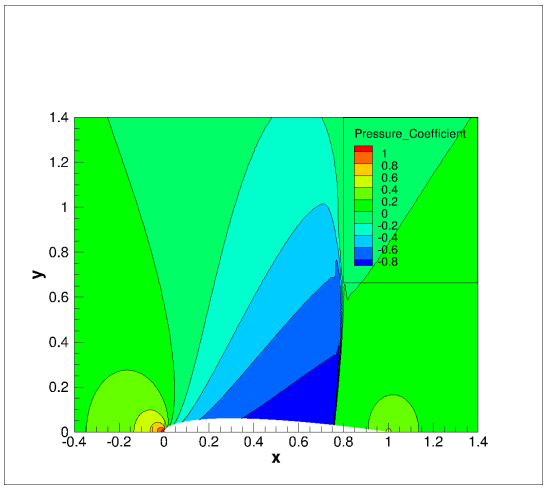
Table 4.4: Grid convergence study for the optimized shape of BC I at $M_\infty = 0.85$ and $\alpha = 0.0^o$.

Mesh	**Grid Size**	$\mathbf{C}_l$	$\mathbf{C}_d$	**Simulation Time** * **(min)**
1	32×32 (961)	0.0	272.7192	0.25
2	64×64 (3,969)	0.0	129.5981	0.42
3	128×128 (16,129)	0.0	83.0096	0.76
4	256×256 (65,025)	0.0	77.2443	1.59
5	512×512 (261,121)	0.0	74.5209	4.41
6	$1,024 \times 1,024$ (1,046,529)	0.0	74.1940	23.34

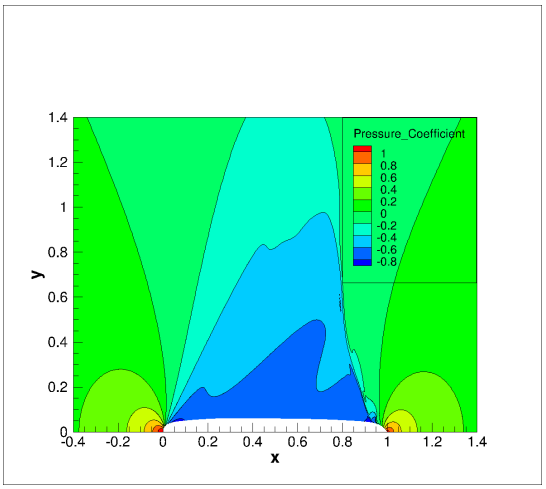
*Computed on a high-performance cluster with 32 processors. Flow solution only.

optimization time is about 4,949 minutes. Note that both of the multi-fidelity methods used in this study only require the flow solutions but not the adjoint solutions like the gradient-based algorithm. Consequently, the SM and MM algorithms are more efficient than the Direct B algorithm. Direct A needs 1,383 high-fidelity model evaluations (only flow solutions) and the total optimization time is about 6,939 minutes.

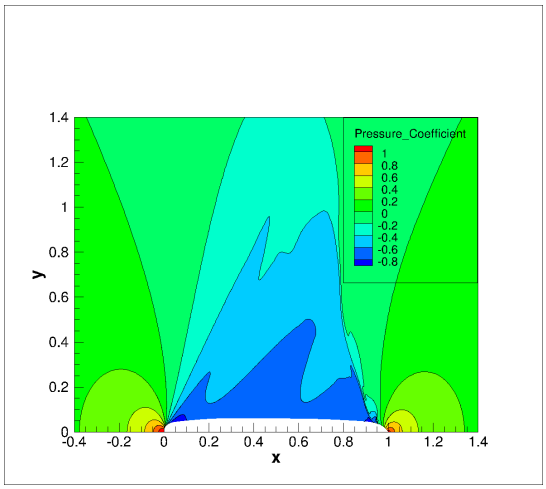
The problem (4.1) is solved using the pattern search algorithm and the low-fidelity model $\mathbf{c}$ only (mesh 2 of Table 4.1 without the SM or MM corrections). The results are shown in Fig. 4.7. We can clearly see that the low-fidelity optimum is much worse than the optimized shapes obtained by MM and pattern search with the high-fidelity model since it has a very strong double shock near the trailing-edge. In face the drag



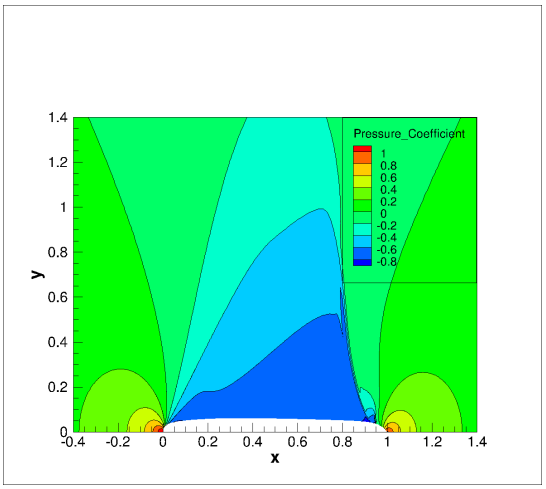
(a) Baseline



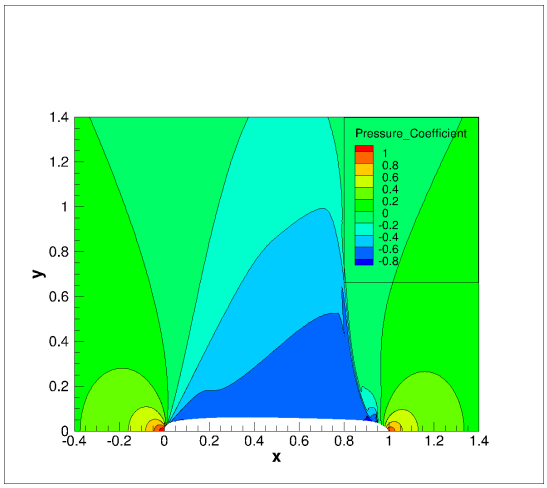
(b) Direct A



(c) Direct B

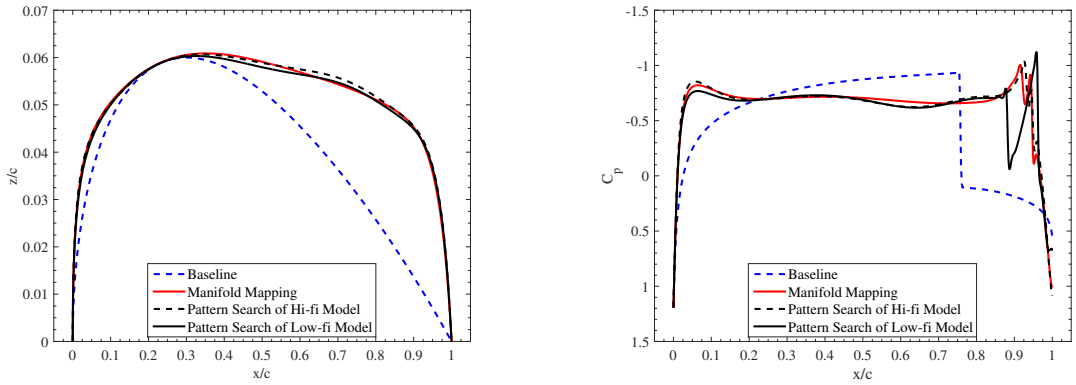


(d) Space mapping



(e) Manifold mapping

Figure 4.6: Comparison of BC I baseline and optimized pressure coefficient contours.



(a) Shapes (b) Pressure coefficient distributions

Figure 4.7: Comparison of BC I baseline and optimized pressure coefficient of low-fidelity model.

coefficient value of the low-fidelity optimum is around 114 counts, where as the high-fidelity optimum is around 55 counts.

4.2 Benchmark Case II: Lift-constrained Drag Minimization of the RAE 2822 in Transonic Viscous Flow

The multi-fidelity optimization algorithms in Chapter 3 are applied to BC II involving viscous transonic flow past the RAE 2822 airfoil shape. The optimization results are compared with gradient-based search in terms of the design quality as well as the number of model evaluations and total optimization time.

4.2.1 Problem definition

The objective is to minimize the drag coefficient (C_d) of the RAE 2822 airfoil at a free-stream Mach number of $M_\infty = 0.734$, lift coefficient of 0.824, and Reynolds number of 6.5×10^6, subject to an area and pitching moment constraints. The task is to solve the following constrained optimization problem

$$\min_{\mathbf{l} \leq \mathbf{x} \leq \mathbf{u}} C_d, \tag{4.7}$$

subject to the following constraints:

$$C_l = 0.824, \tag{4.8}$$

$$C_m \geq -0.092, \tag{4.9}$$

$$A \geq A_{baseline}, \tag{4.10}$$

where C_m is the moment coefficient and A is the airfoil cross-sectional area nondimensionalized with the chord length squared.

The constant lift coefficient constraint (4.8) is implicitly satisfied in the flow solver by using the angle of attack as a dummy parameter. In the implementation of the multi-fidelity algorithms, the pitching moment and cross-sectional area constraints (4.9) and (4.10), respectively, are handled through a penalty function. In the direct optimization algorithms, those constraints are handled directly.

4.2.2 Design variables

The B-spline parameterization approach, described in Section 4.1, is used in BC II for the upper and lower surfaces. We use 10 control points, as shown in Fig. 4.8, where two are fixed at the leading- and trailing-edges, and the other ones, 4 for each surface, can move in the vertical direction. This yields 8 design variables. Based on a fit to the RAE 2822, we set the x-locations of the free control points as

$$\mathbf{X} = [\mathbf{X}_u; \mathbf{X}_l]^T = [0.0 \quad 0.15 \quad 0.45 \quad 0.80; \quad 0.0 \quad 0.35 \quad 0.60 \quad 0.90]^T.$$

The initial design variable vector is

$$\mathbf{x} = [\mathbf{x}_u; \mathbf{x}_l]^T = [0.0175 \quad 0.0498 \quad 0.0688 \quad 0.0406; \quad -0.0291 \quad -0.0679 \quad -0.0384 \quad 0.0054]^T.$$

The lower bound of $\mathbf{x}$ is set as $\mathbf{l} = [0 \quad 0 \quad 0 \quad 0; \quad -1 \quad -1 \quad -1 \quad -1]^T$, and the upper bound is set as $\mathbf{u} = [1 \quad 1 \quad 1 \quad 1; \quad 0 \quad 0 \quad 0 \quad 0]^T$.

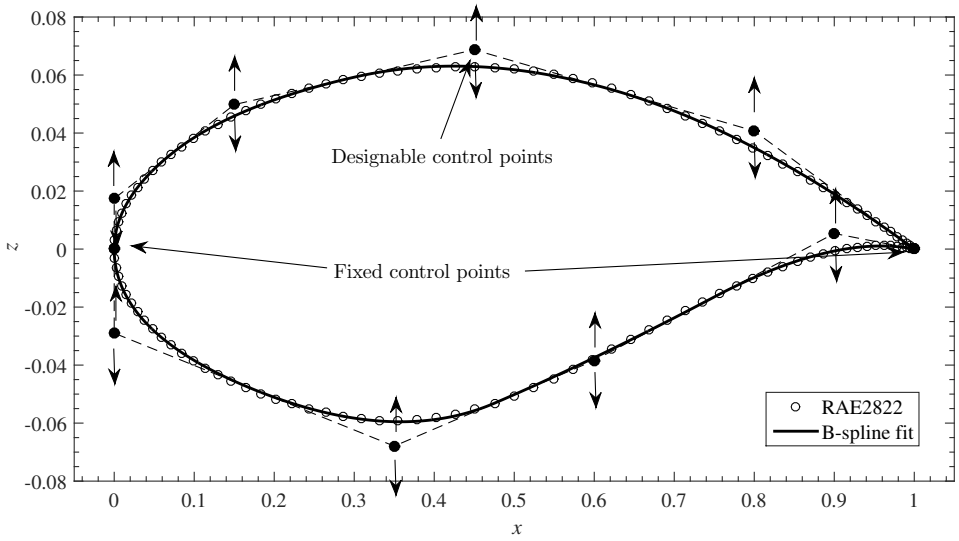
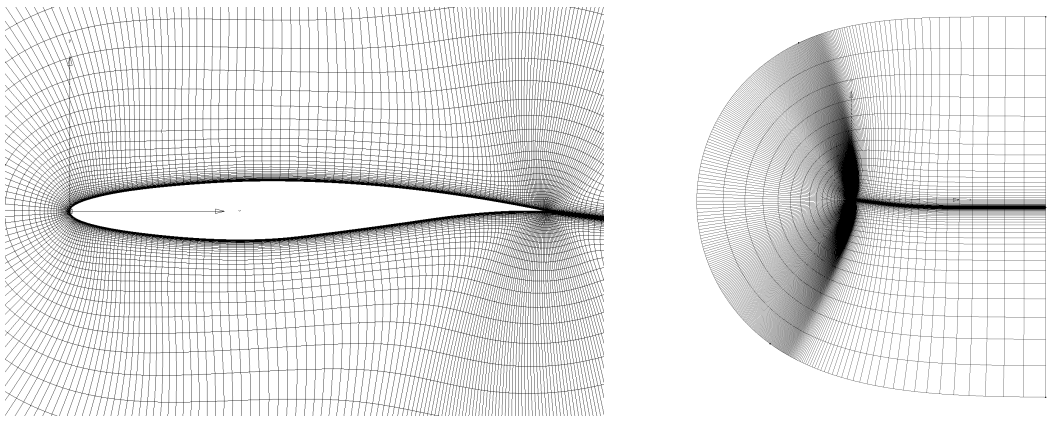


Figure 4.8: B-spline parameterization for the airfoil surface.

4.2.3 High-fidelity CFD model

The SU^2 implicit density-based flow solver [72] is used for the viscous case, solving the steady compressible Reynolds-averaged Navier-Stokes (RANS) Equations with the Spalart-Allmaras turbulent model [73]. The convective flux will be calculated using the second order JST scheme [74]. One multi-grid level is used for solution acceleration. The turbulent variables are convected using a first-order scalar upwind method. The flow solver convergence criterion is the one that occurs first of the two: (i) the change in the drag coefficient value over the last 100 iterations is less than 10^{-5}, or (ii) a maximum number of iterations of 5,000 is met.

The grids are generated using a hyperbolic C-mesh, [75] (see Fig. 4.9). The far-field is set 100 chords away from the airfoil surface. The grid points are clustered at the trailing edge and the leading edge of the airfoil to give a minimum streamwise spacing of $0.001c$, and the distance from the airfoil surface to the fist node is $4 \times 10^{-6}c$. The grid density is controlled by the number of points in the streamwise direction (N_s), and the number of points in the direction normal to airfoil surface (N_n). We set the number of



(a) Surface (b) Far-field

Figure 4.9: Mesh views of hyperbolic C-mesh used in viscous model.

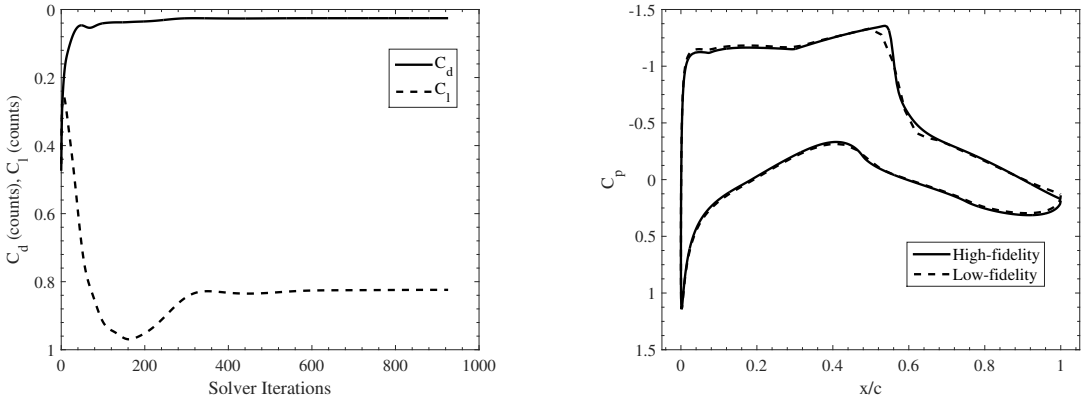
Table 4.5: Grid convergence study for the baseline shape of BC II.

Mesh	**Grid Size**	$\mathbf{C}_l(cts)$	$\mathbf{C}_d(cts)$	**Simulation Time** * **(min)**
1	160×40 (9,836)	82.40	255.9	5.5
2	320×80 (38,876)	82.40	215.0	20.8
3	640×160 (154,556)	82.39	202.4	66.0
4	$1,280 \times 320$ (616,316)	82.41	201.9	275.7

*Computed on a high-performance cluster with 32 processors. Flow solution only.

points in the wake region equal to the number in the normal direction. The grid sizes are denoted by $N_s \times N_n$. Table 4.5 gives the results of a grid convergence study using the RAE 2822 airfoil at $M_\infty = 0.734$ and $C_l = 0.824$. The constant lift condition is determined by externally changing the angle of attack using a numerical optimization algorithm. Typically, 3 to 4 iterations are needed. The simulation times presented in Table 4.5 give the overall time to compute the constant lift condition. The times do not include the time it takes to calculate the adjoint solutions.

For the optimization studies, we use Mesh 3 for the high-fidelity model **f** and Mesh 1 for the low-fidelity model **c**. The low-fidelity model convergence criteria is the same as the high-fidelity one, but with the following values: drag coefficient convergence set to 10^{-4}, and the maximum number of iterations is set to 2,000. Figure 4.10 shows that the low-fidelity solver is converged well within the 1,000 iteration limit. Also the low-fidelity



(a) Low-fidelity model (b) High- and low-fidelity models

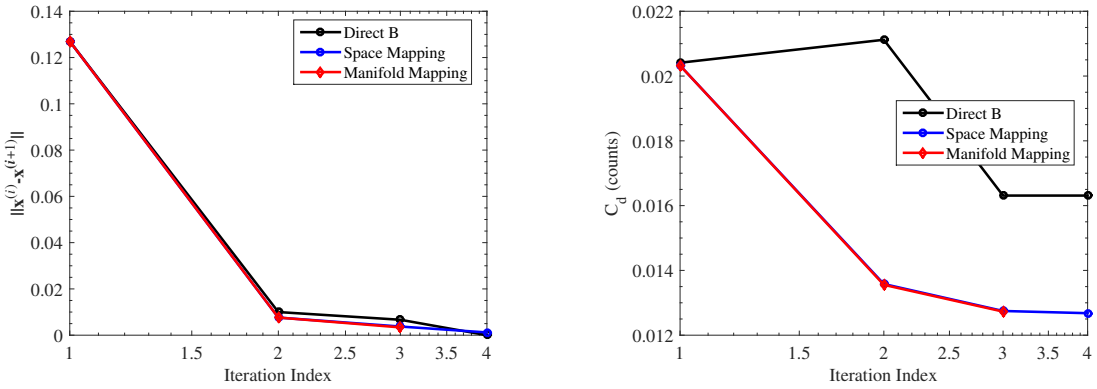
Figure 4.10: Solver convergence history of the high- and low-fidelity model.

model is a good representation of high-fidelity one in terms of the pressure coefficient distributions.

4.2.4 Results

BC II is solved using the Direct B, SM, and MM algorithms as described in Table 4.2, but using Meshes 3 and 1 of Table 4.5 for the high- and low-fidelity models, respectively. Figure 4.11 shows the convergence history of the algorithms. The Direct B algorithm was terminating after 4 design iterations based on the argument. The multi-fidelity algorithms are, however, able to make significant reduction in the objective function. SM terminates after 4 design iterations based on the design variables, and MM terminates after 3 design iterations.

As can be seen by Table 4.6, the SM algorithm reduces the drag coefficient value from 202.4 counts to 126.8 counts (or by 75.6 counts). The MM algorithm reduces the drag coefficient value from 202.4 counts to 127.3 counts (or by 75.1 counts). Direct B reduces the drag coefficient value to 163.1 counts (or by 39.3 counts). Figure 4.12 shows the optimized shapes and pressure coefficient distributions. Figure 4.13 shows the pressure coefficient contours. Clearly, the SM and MM reduce the shock strength significantly.



(a) Argument (b) Objective function

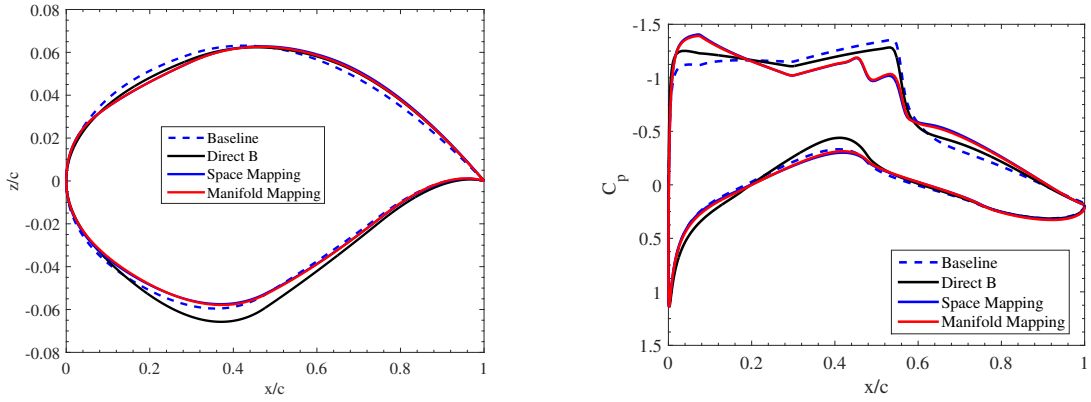
Figure 4.11: BC II convergence history.

Table 4.6: Optimization results for BC II.

Parameter/Method	**Baseline**	**Direct B**	**SM**	**MM**
$C_l(l.c.)$	82.39	82.40	82.45	82.39
$C_d(d.c.)$	202.4	163.10	126.80	127.30
$C_{m,c/4}$	-0.0961	-0.0883	-0.0920	-0.0915
A	0.0779	0.0779	0.0772	0.0769
N_c	-	-	1,878	1,621
N_f	-	55	17	13
N_{equ}	-	55	≈ 133	≈ 114
$t_c(min)$	-	-	1,807	1,566
t_f (min)	-	535.9	265.6	202.0
t_{tot} (min)	-	535.9	2,073.0	1,768.0

However, the cross-sectional area constraints are slightly violated for both SM and MM. In particular, the cross-sectional areas of SM is 0.90% below the baseline value of 0.0779, and MM is 1.28% below. These violations are minor. The Direct B algorithm satisfies the pitching moment coefficient constraint and cross-sectional area properly.

The optimization cost of the SM and MM algorithms is comparable. SM needs approximately 133 equivalent high-fidelity model evaluations (17 high-fidelity and 1,878 low-fidelity). MM needs approximately 114 equivalent high-fidelity model evaluations (13 high-fidelity and 1,621 low-fidelity). In terms of time, SM takes 34.6 hours, and MM takes 29.5 hours. Table 4.7 indicates that the meshes used in the studies here are



(a) Shapes (b) Pressure coefficient distributions

Figure 4.12: BC II baseline and optimized shapes and pressure distributions.

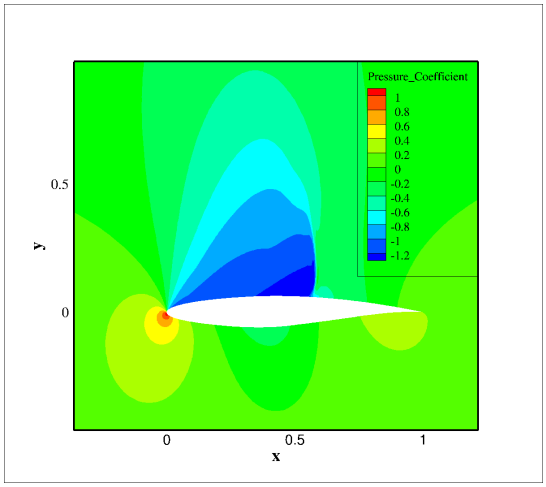
Table 4.7: Grid convergence study for the optimized shape of BC II.

Mesh	**Grid Size**	$\mathbf{C}_l(cts)$	$\mathbf{C}_d(cts)$	**Simulation Time** * **(min)**
1	160×40 (9,836)	82.38	179.90	4.1
2	320×80 (38,876)	82.38	134.51	25.9
3	640×160 (154,556)	82.40	126.80	66.1
4	$1,280 \times 320$ (616,316)	82.38	124.74	498.1

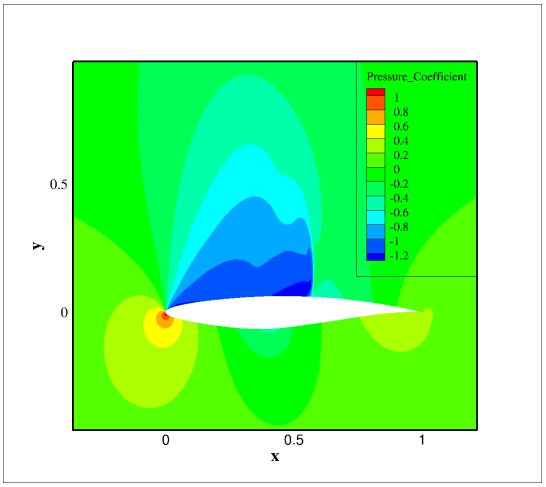
*Computed on a high-performance cluster with 32 processors. Flow solution only.

converged within 2 drag count. Direct B algorithm shows low cost in this case in terms of time, however, from Fig. 4.11, the objective function was not able to approach further optimization iteration after the third one.

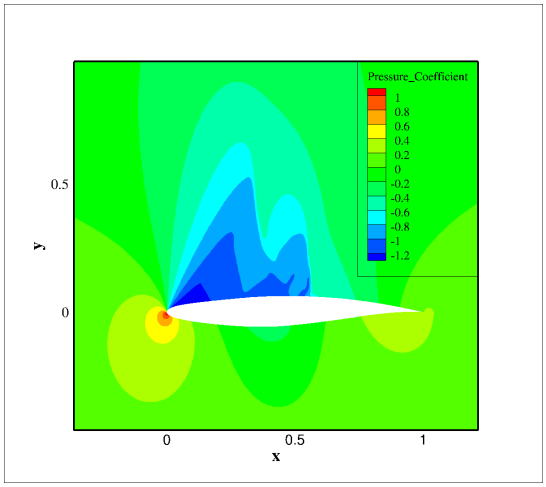
From the two benchmark cases above, it is noticed that MM performs very similarly with SM. This was expected since since high-fidelity model responses are scalars rather than vector responses, so the following two applications will introduce the advantages of MM when the responses are vectors.



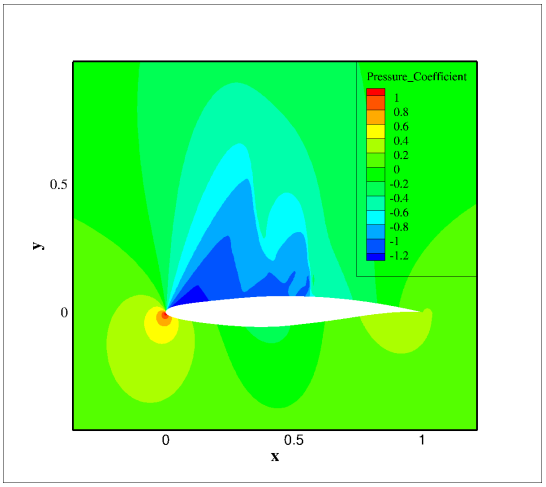
(a) Baseline



(b) Direct B



(c) Space mapping



(d) Manifold mapping

Figure 4.13: BC II baseline and optimized pressure coefficient contours.

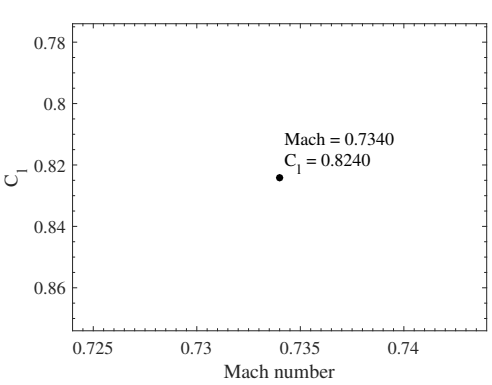
4.3 Case III: Multipoint Lift-constrained Drag Minimization in Transonic Inviscid Flow

Transport aircraft operate at multiple conditions due to various flight missions and air traffic control restrictions [16]. Small changes in the airfoil shape may effect the fuel burn efficiency significantly in the off-design points, which may increase the operating cost and greenhouse gas emission [16]. This is the specific motivation for multipoint design.

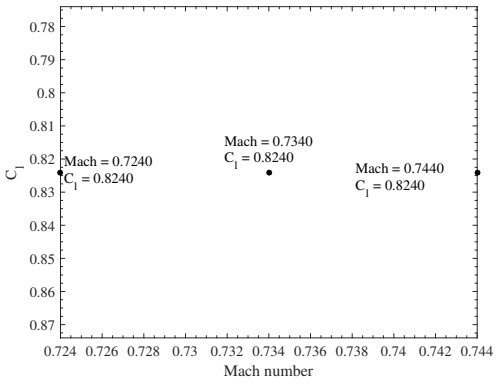
4.3.1 Problem definition

In this section, a total of 3 cases are investigated. Like BC I and II, the idea of these come from the Aerodynamic Design Optimization Discussion Group (ADODG) Benchmark Case IV where the problem investigates the optimum solution for the 3-D CRM wing. Martins et al. [4, 16] have performed a lot of work on this benchmark case. In this section, in order to focus on the MM algorithm, the model has been simplified, so that geometry being used is from BC II, i.e., RAE 2822 airfoil. The objective is to optimize the weighted sum of drag coefficients for several operation conditions. Comparison between multipoint design and single point design is performed. Case IIIb is the baseline single point optimization. The single point case and the new multipoint cases are summarized in Table 4.8. Case IIIa and Case IIIc are multipoint problems considered here for different Mach numbers and lift conefficients. The weight factors τ_i for the cases are uniformly distributed over the Mach number and C_l space. Note that the weights and the problem formulations are based on the AIAA ADODG BC II. The general form of each optimization problem can be described as follows.

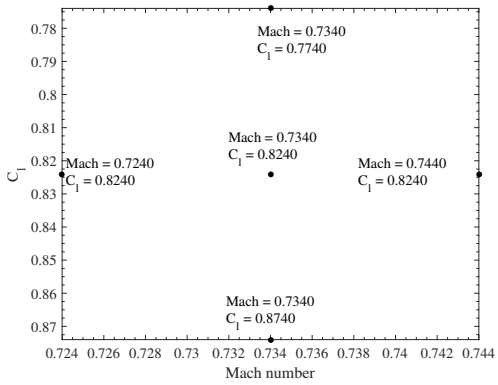
The objective is to minimize the weighted average drag coefficient $\sum_{i=1}^{N} \tau_i C_{di}$ subject to an area and pitching moment constraint. The task is to solve the following constrained



(a) Case IIIa



(b) Case IIIb



(c) Case IIIc

Figure 4.14: Operation conditions of the singlepoint and multipoint designs.

Table 4.8: Operating conditions for each optimization cases.

Case	Point	Weights(τ_i)	Mach	C_l
IIIa	1	1	0.734	0.824
IIIb	1	$\frac{1}{4}$	0.724	0.824
	2	$\frac{1}{2}$	0.734	0.824
	3	$\frac{1}{4}$	0.744	0.824
IIIc	1	$\frac{1}{5}$	0.724	0.824
	2	$\frac{1}{5}$	0.734	0.774
	3	$\frac{1}{5}$	0.734	0.824
	4	$\frac{1}{5}$	0.734	0.874
	5	$\frac{1}{5}$	0.744	0.824

optimization problem:

$$\min_{\mathbf{l}\leq\mathbf{x}\leq\mathbf{u}} \sum_{i=1}^{N} \tau_i C_{di}, \tag{4.11}$$

subject to the following constraints:

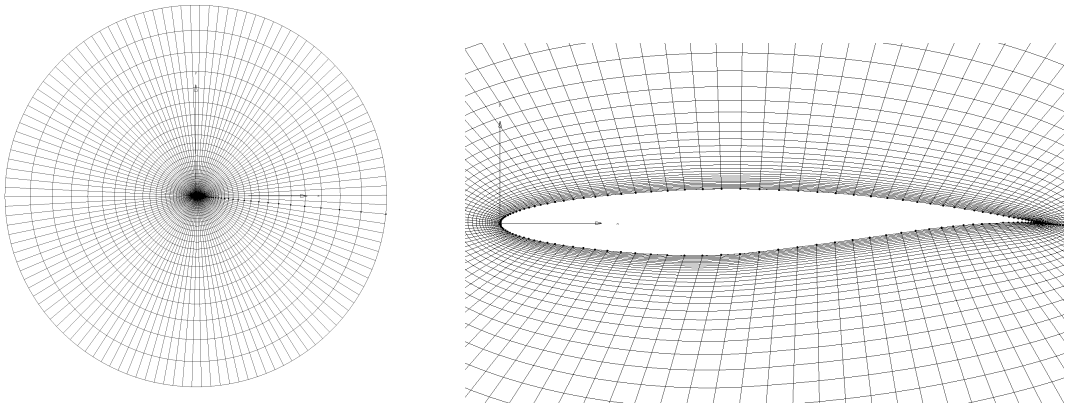
$$C_l = 0.824, \tag{4.12}$$

$$A \geq A_{baseline}, \tag{4.13}$$

The constant lift coefficient constraint (4.12) is implicitly satisfied in the flow solver by using the angle of attack as a dummy parameter. In the implementation of the multi-fidelity algorithms, the cross-sectional area constraint (4.13) is handled through a penalty function. The B-spline parameterization approach, described in Section 4.1, is used this application for the upper and lower surfaces. We still use 10 control points, as shown in Fig. 4.8.

4.3.2 High-fidelity CFD model

An O-type computational mesh is generated using Pointwise [71] (see Fig. 4.15. The far-field boundary is set 55 chord lengths away from the airfoil surface. The results of a



(a) Far-field (b) Surface

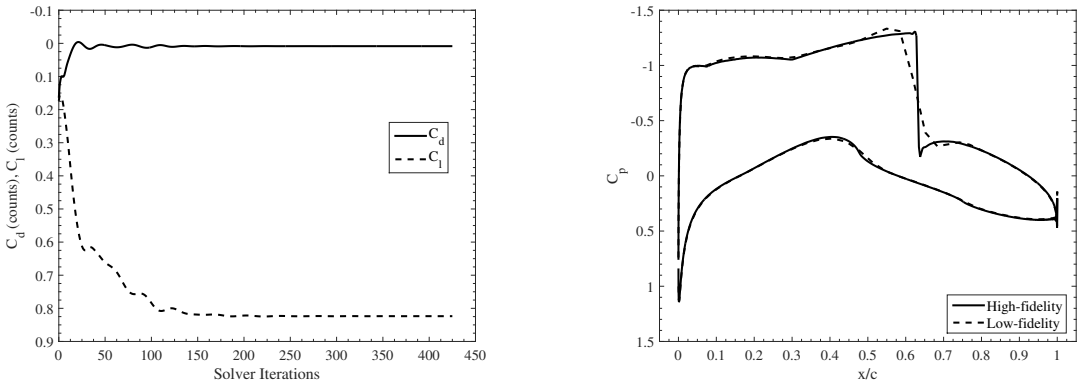
Figure 4.15: O-mesh views of RAE 2822 airfoil.

Table 4.9: Grid convergence study for the baseline shape (RAE 2822).

Mesh	**Grid Size**	$\mathbf{C}_d$	$\mathbf{C}_l$	**Alpha**	**Simulation Time** * **(min)**
1	32×32 (1,922)	127.36	82.42	2.087	0.9
2	64×64 (7,938)	84.60	82.37	1.922	1.1
3	128×128 (32,258)	78.06	82.40	1.877	2.4
4	256×256 (130,050)	76.60	82.40	1.852	6.7
5	512×512 (522,242)	76.37	82.40	1.847	20.7
6	1024×1024 (2,093,058)	76.30	82.41	1.843	114.0

*Computed on a high-performance cluster with 32 processors. Flow solution only.

grid convergence study, given in Table 4.9, revealed that a 512×512 mesh (shown number 5 in the table) is required for convergence within 0.1 drag count when compared with the next mesh. Distance to the first grid point is 0.0004c. The flow simulation for Mesh 5 takes about 20.7 minutes. SU^2 [72] is utilized for the inviscid fluid flow simulations. The flow solver convergence criterion is the same with BC I solver convergence criterion in Section 4.1. As the same with BC II, the constant lift condition is determined by externally changing the angle of attack using a numerical optimization algorithm. The simulation times presented in Table 4.9 give the overall time to compute the constant lift condition.



(a) Low-fidelity model

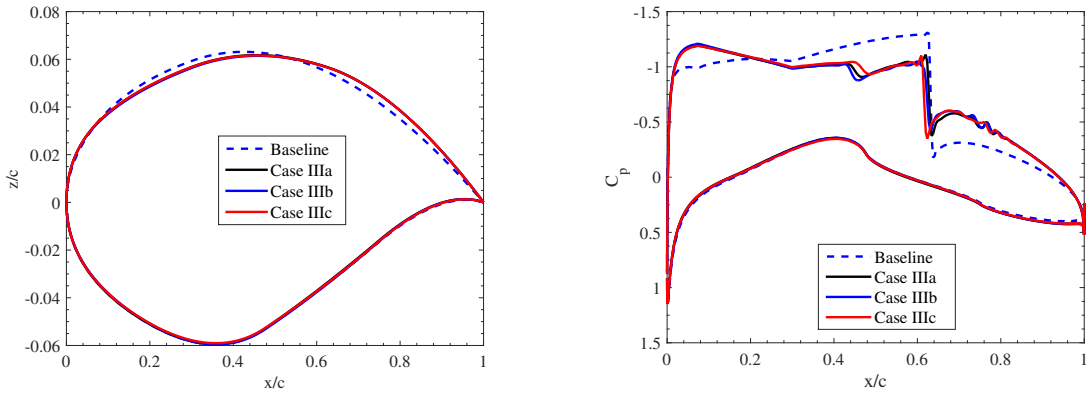
(b) High- and low-fidelity models

Figure 4.16: Solver convergence history of high- and low-fidelity simulation.

For the optimization studies, we use Mesh 5 for the high-fidelity model **f** and Mesh 2 for the low-fidelity model **c**. The low-fidelity model convergence criteria is the same as the high-fidelity one, but with the following values: drag coefficient convergence set to 10^{-4}, and the maximum number of iterations is set to 500. Figure 4.16 shows that the low-fidelity solver is converged well within the 500 iteration limit which allows the deformed mesh to converge properly. Fig. 4.16 shows that the low-fidelity model has a slight mismatch at the edge of shock on the upper surface with high-fidelity model. However, from the optimized results and consideration of time ratio between high- and low- fidelity model, it is still a preferred representation of high-fidelity model.

4.3.3 Results

This multipoint design is solved by using MM algorithms as described in Table 4.2, but using Meshes 5 and 2 of Table 4.9 for the high- and low-fidelity models, respectively. Figure 4.17 shows the optimized shapes and pressure distributions of different design cases. The multi-fidelity algorithms are able to make a significant reduction in the objective function.

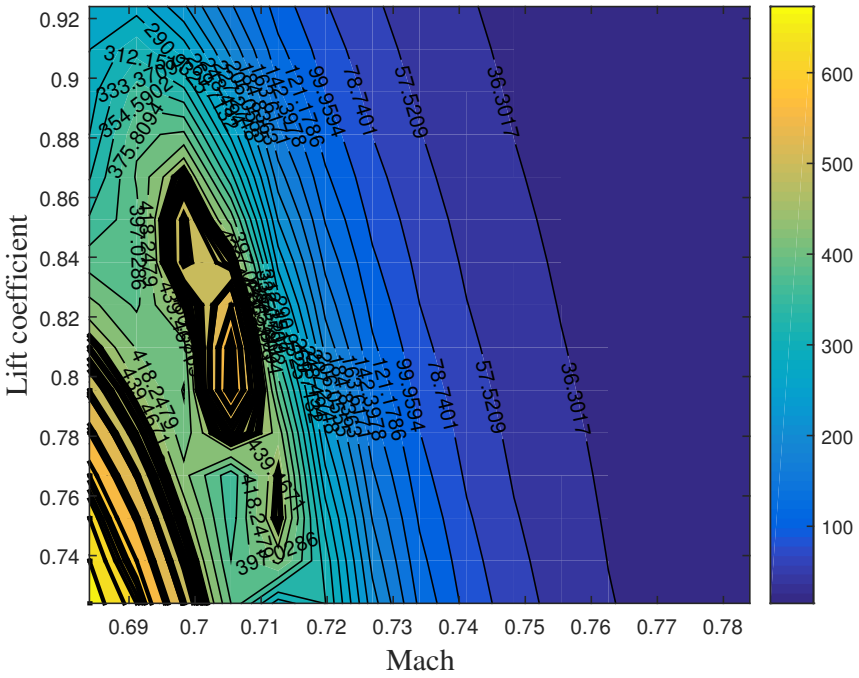


(a) Shapes (b) Pressure coefficient distributions

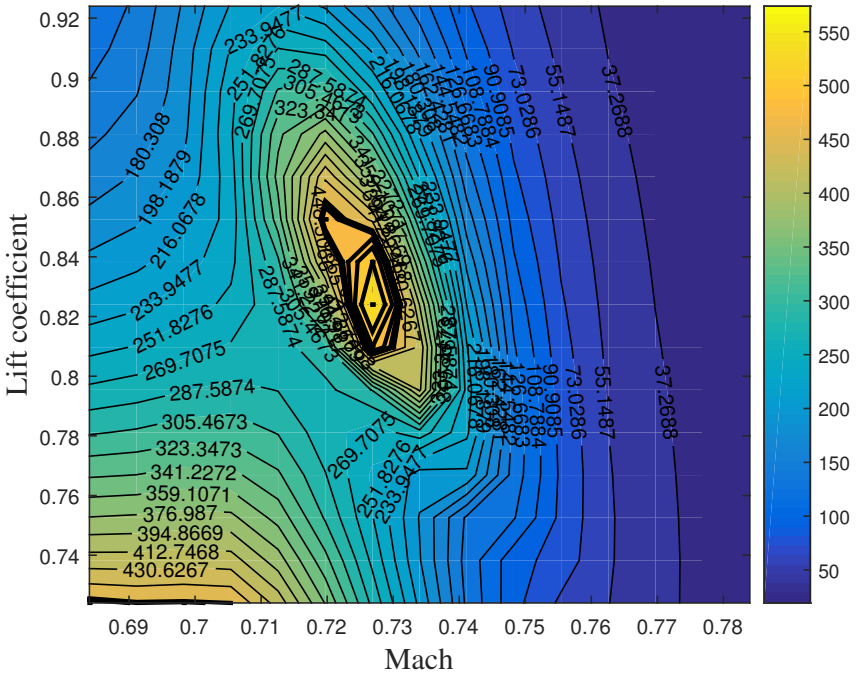
Figure 4.17: Multipoint design baseline and optimized characteristics.

To explain the robustness of the multipoint design properly, we compare the ML/D contours with respect to C_l and Mach number for the baseline, single point design (Case IIIa), and multipoint designs (Cases IIIa and IIIb). ML/D provides a metric for quantifying aircraft range based on the Breguet range equation with constant thrust-specific fuel consumption [16]. It is assumed the thrust-specific fuel consumption is constant in this case when performance falls into limited range of Mach number [76]. From Figs. 4.18 and 4.19, the maximum of the baseline airfoil falls on a lower Mach number and a higher C_l than the nominal flight condition. The single-point optimization in Fig. 4.18 shows a movement of the optimum range of ML/D toward the nominal cruise condition. The multipoint design in Fig. 4.19 shows a flatter area of ML/D contours near the maximum of the nominal condition, which means the aircraft can cruise at a wider robust range in terms of Mach number and C_l. Also, the maximum ML/D of Case IIIb of multipoint design is 585.6 which is slightly higher than the single-point design (Case IIIa) maximum of $ML/D = 573.7$.

In aircraft design, the 99% value of maximum ML/D is the often used to examine the robustness of the design [77], however, this comparison is not totally fair since the maximum of each case varies. To compare more accurately, the area where ML/D is

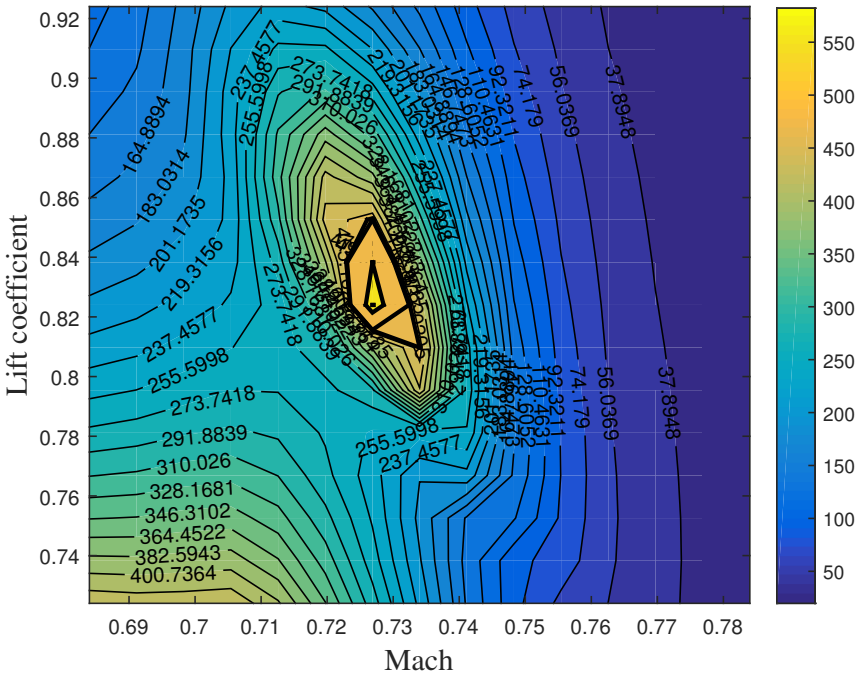


(a) Baseline

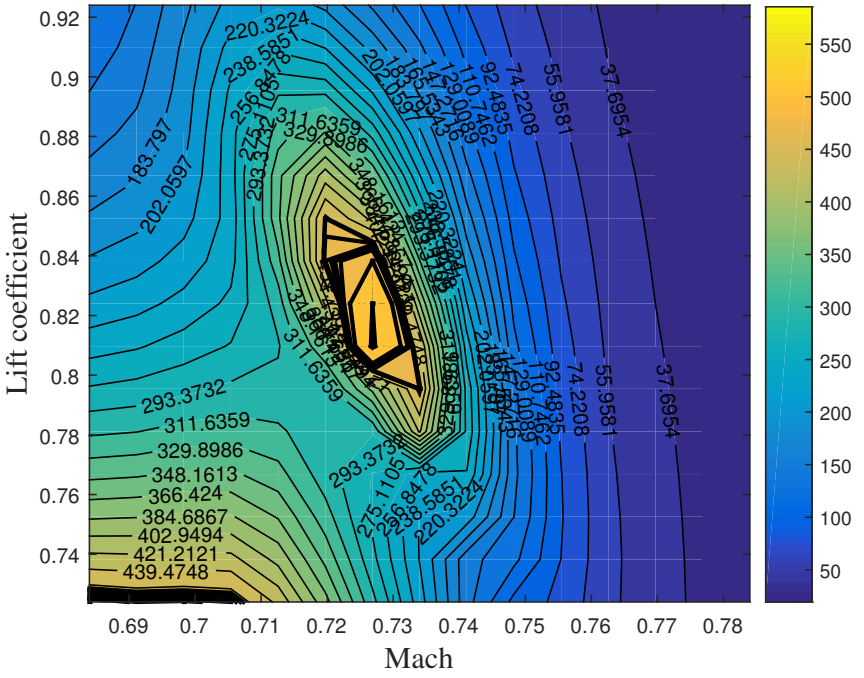


(b) Case IIIa: single point

Figure 4.18: Baseline and single point design ML/D contours.



(a) Case IIIb: 3 multipoints



(b) Case IIIc: 5 multipoints

Figure 4.19: Multipoint design ML/D contours.

greater than 450 is selected and compared for all designs in Figs. 4.18 and 4.19. As can be seen in Case IIIc, the bold line closed area is greater than the bold area in Case IIIa, this means the robust range of optimized airfoil in Case IIIc can perform more robustly under wider cruise conditions.

4.4 Case IV: Inverse Design in Transonic Inviscid Flow

Inverse design is another case where the MM shows the capability of dealing with vector responses. Instead of optimizing the figure of merit directly, a target pressure distribution is considered as the objective response that can provide a desired aerodynamic feature. The target pressure distribution has to be obtained beforehand [78] however. Figure 4.22 shows the general algorithm for inverse design.

4.4.1 Problem definition

The initial geometry is the RAE 2822 airfoil. The objective is to optimize the norm of the current pressure distribution and a target pressure distribution. Here, we use the optimized shape from Case IIIa to generate the target pressure distribution.

We solve the following constrained minimization problem

$$\min_{\mathbf{l}\leq\mathbf{x}\leq\mathbf{u}} \left\| C_p(\mathbf{x}) - C_p^* \right\| \tag{4.14}$$

subject to the following constraints

$$C_l = 0.824, \tag{4.15}$$

$$A \geq A_{baseline}, \tag{4.16}$$

where $C_p(\mathbf{x}$ is the pressure distribution of the current design $\mathbf{x}$, and C_p^* is the target pressure distribution.

The constant lift coefficient constraint (4.15) is implicitly satisfied in the flow solver by using the angle of attack as a dummy parameter. In the implementation of the

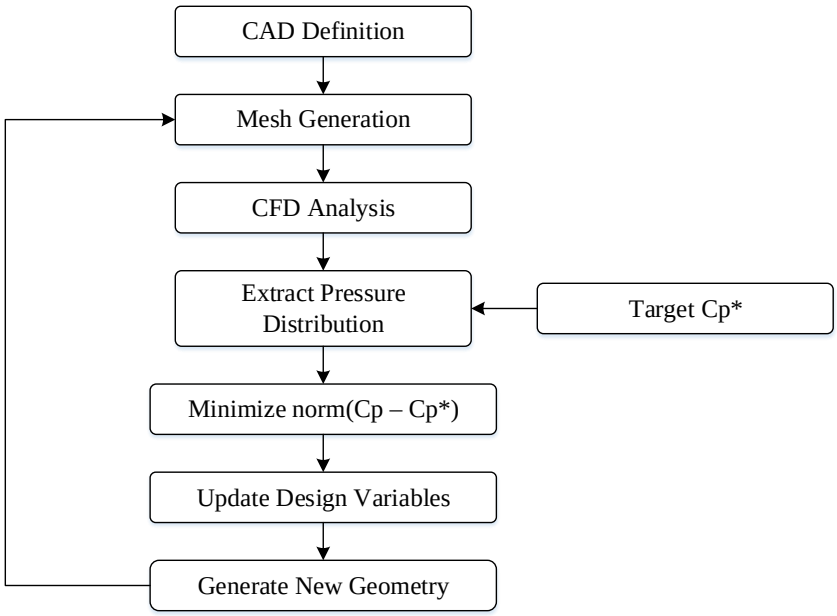


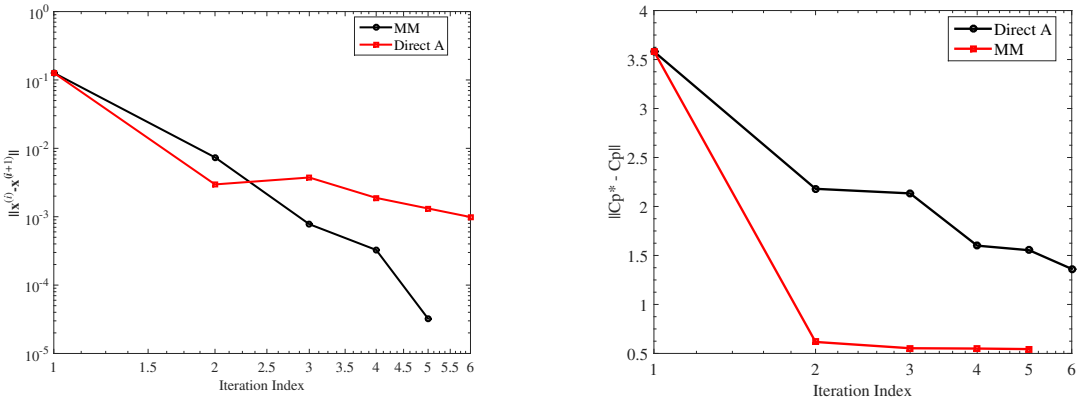
Figure 4.20: Inverse design algorithm flowchart.

multi-fidelity algorithms, the cross-sectional area constraints (4.16) is handled through a penalty function. The B-spline parameterization approach, described in Section 4.1, is used in this application for the upper and lower surfaces. We still use 10 control points, as shown in Fig. 4.8. The CFD solver is described in Section 4.1.

4.4.2 Results

This inverse design problem is solved by using Direct A and MM algorithm as described in Table 4.2, but using Meshes 5 and 2 of Table 4.9 for the high- and low-fidelity models, respectively. Figure 4.21 shows the convergence history of the algorithms. The multi-fidelity algorithm makes a significant reduction in the objective function and terminates after 4 design iterations based on the objective function. The direct algorithm terminates after 5 iterations on objective function like MM.

The overall optimized shapes look very similar (see Fig. 4.22). Since the highly nonlinear PDE simulation, a small shape change may generate big difference on the physics and effect the characteristics. There is a slight difference between the shapes

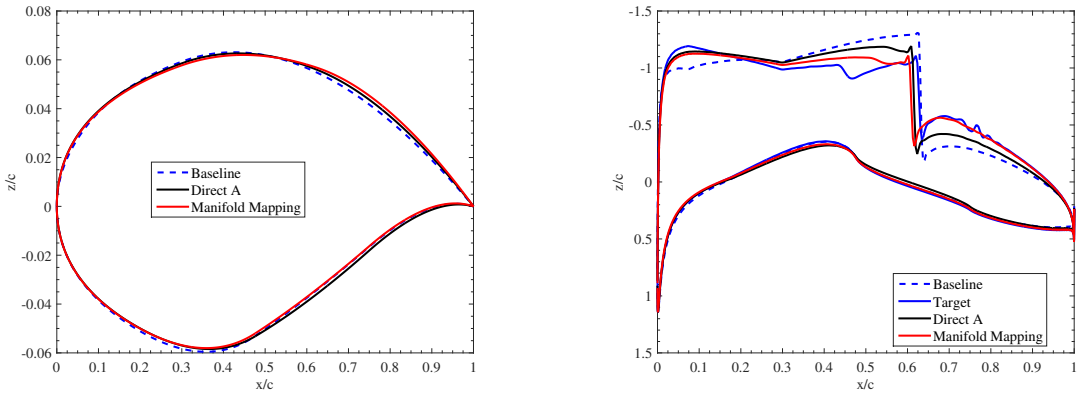


(a) Argument (b) Objective function

Figure 4.21: Inverse design convergence history.

Table 4.10: Optimization results for inverse design.

Parameter/Method	**Baseline**	**Target**	**Direct A**	**MM**
$C_l(l.c.)$	82.38	82.42	82.41	82.41
$C_d(d.c.)$	76.27	15.15	39.77	22.33
N_c	-	-	-	741.0
N_f	-	-	1,302	24.0
N_{equ}	-	-	1,302	≈ 145
t_c (min)	-	-	-	816.2
t_f (min)	-	-	9,766	161.4
t_{tot} (min)	-	-	9,766	977.6



(a) Shapes (b) Pressure coefficient distributions

Figure 4.22: Inverse design baseline and optimized characteristics.

obtained by the direct algorithm and the multi-fidelity algorithm, however, in terms of the pressure coefficient distributions, shown in Fig. 4.22, at $x/c = 0.6$ to 0.9 of the pressure coefficient distribution, MM reaches to the edge of target C_p more closely than Direct A. Specifically at portion from 0.7 of x/c, MM is able to overlap with target design, and Direct A shows C_p distribution between baseline and target design. Other than that, the distributions obtained by both Direct A and MM near the leading-edge of the airfoil have distinctly lower peak than the target pressure distribution, but higher than the baseline. Both of the shapes are approaching to the target pressure distribution though.

In terms of computational cost in Table 4.10, the multi-fidelity algorithm is faster than the direct one. In particular, it needs 24 high-fidelity model evaluations (N_f), and 741 low-fidelity model evaluations (N_c); which, in total, is equivalent to less than 150 high-fidelity model evaluations (N_{tot}). The total optimization time is about 977.6 minutes. Direct A needed 1,302 high-fidelity model evaluations (includes flow simulation only) and the total optimization time is about 9,766 minutes. Consequently, the MM algorithms are more efficient than the Direct A algorithm in terms of computational cost and optimization quality.

CHAPTER 5. CONCLUSION

A robust and computationally efficient optimization methodology for aerodynamic shape optimization is presented. The approach uses low-fidelity models corrected by the manifold mapping (MM) technique to create fast and reliable surrogates which are utilized to search for an approximate optimum of an expensive high-fidelity model at a low CPU cost. The MM correction is applied both to the objectives and constraints to ensure a good alignment between the surrogate and the high-fidelity model. The MM approach developed here does not require gradient information. The approach is applied to benchmark cases involving direct design of airfoil shapes in two-dimensional inviscid and viscous transonic flows. The results of these benchmark cases show that the optimized designs are obtained at a significantly lower computational cost compared to the direct high-fidelity model optimization with adjoint sensitivity information when using eight design variables. In those cases, which involve scalar objective functions, the cost is comparable with the space mapping algorithm (also based on multi-fidelity models). In terms of vector responses, the MM algorithm proves to be efficient. To this end, multipoint design problems were solved using MM with good results. Also, the MM algorithm was demonstrated on an inverse design problem. The MM algorithm gets close to the target pressure distribution at a fraction of the cost of using a derivative-free approach.

Numerical applications of benchmark cases, multipoint design and inverse design using the manifold framework illustrate the following:

- In the benchmark cases considered here, the objective functions are all scalars. In such cases, the MM algorithm yields comparable results to the space mapping algorithm.

- MM typically works well for design cases with vector-valued responses. Multipoint problems are an example of aerodynamic design cases using vector responses. Here, the aerodynamic surface is optimized at multi operating conditions, e.g., at several sets of Mach number and lift coefficients. MM was easy to apply to such a case, and gave satisfactory results at a low cost.

- Inverse design is another example of an aerodynamic design case requiring vector-valued responses. MM was applied to the shape optimization for matching a target pressure distribution. In this case, MM outperforms a derivative-free approach in terms of both cost and matching of the distributions.

Future work in this research area should consider the following:

- Since our implementation uses pattern search to optimize the surrogate model, the cost with increasing number of design variables grows quickly. Future work will investigate alternatives to efficiently optimize the surrogate model. In particular, extension of MM using adjoint sensitivity information will be investigated, as well as adding techniques for the reduction of the design space dimensionality. This will enable the use of MM for large-scale cases, e.g., the aerodynamic shape optimization of three-dimensional wings which require several hundred design variables.

- Another possible future investigation is to combine space mapping and MM with space mapping used as a preconditioner for the low-fidelity model. In such as case, the multi-point space mapping surrogate may be established using, e.g., star-distribution design of experiments in order to improve the global matching between the models (in which case only the global parameters $\mathbf{A}$ and $\mathbf{D}$ need to be calculated), and the MM correction may be applied on the top of SM.

BIBLIOGRAPHY

[1] D. Echeverria. *Multi-level Optimization: Space Mapping and Manifold Mapping*. PhD thesis, Thomas Stieltjes Institute for Mathematics, 2007.

[2] G. Kenway and J. R. R. A. Martins. Aerostructural shape optimization of wind turbine blades considering site-specific winds. In *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Victoria, British Columbia, Canada, September 10-12, 2008.

[3] A. Mousavi and S. Nadarajah. Heat transfer optimization of gas turbine blades using an adjoint approach. In *13th AIAA/ISSMO Multidisciplinary Analysis Optimizaiton Conference*, Fort Worth, Texas, Sep. 13-15, 2010.

[4] G. K. W. Kenway and J. R. R. A. Martins. Multipoint aerodynamic shape optimization investigations of the common research model wing. In *53rd AIAA Aerospace Sciences Meeting*, Kissimmee, FL, January 5-9, 2015.

[5] T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, Dallas, August 18–22, 1986.

[6] E. Polak. *Optimization Algorithms and Consistent Approximations*. Springer Science Business Media, New York, 1997.

[7] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3:233–260, 1988.

[8] C. A. Mader and J. R. R. A. Martins. Derivatives for time-spectral computational fluid dynamics using an automatic differentiation adjoint. *AIAA Journal*, 50(12):2809–2819, 2012.

[9] T. M. Leung and D. W. Zingg. Aerodynamic shape optimization of wings using a parallel newton-krylov approach. *AIAA Journal*, 50(3):540–550, 2005.

[10] B. Epstein and S. Peigin. Constrained aerodynamic optimization of three-dimensional wings driven by navier-stokes computations. *AIAA Journal*, 43(9):1946–1957, 2005.

[11] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, Berlin Heidelberg, 2006.

[12] S. Kim, K. Hosseini, K. Leoviriyakit, and A. Jameson. Enhancement of class of adjoint design methods via optimization of parameters. *AIAA Journal*, 48(6):1072–1076, 2010.

[13] S. Schmidt, N. Gauger, C. Ilic, and V. Schulz. Three dimensional large scale aerodynamic shape optimization based on shape calculus. In *41st AIAA Fluid Dynamics Conference and Exhibit*, Honolulu, Hawaii, Jun. 27-30, 2011.

[14] C. P. van Dam. The aerodynamic design of multi-element high-lift systems for transport airplanes. *Progress in Aerospace Sciences*, 38:101–144, 2002.

[15] A. Jameson, L. Martinelli, J. Alonso, J. Vassberg, and J. Reuther. Simulation based aerodynamic design. In *IEEE Aerospace Conference*, Big Sky, Montana, March 2000.

[16] Z. Lyu, G. K. W. Kenway, and J. R. R. A. Martins. Aerodynamic shape optimization investigations of the common research model wing benchmark. *AIAA Journal*, 53(4):968–984, 2015.

[17] M. Nemec, D. W. Zingg, and T. H. Pulliam. Multipoint and multi-objective aerodynamic shape optimization. *AIAA Journal*, 42(6):1057–1066, June, 2004.

[18] A. Jameson, K. Leoviriyakit, and S. Shankaran. Multi-point aero-structural optimization of wings including planform variations. In *45th Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 811, 2007.

[19] M. Zhang, A. Rizzi, and R. Nangia. Transonic airfoil and wing design using inverse and direct methods. In *53rd AIAA Aerospace Sciences Meeting*, Kissimmee, Florida, Jan. 5-9, 2015.

[20] K. Fujii and G. S. Dulikravich. *Recent development of aerodynamic design methodologies*, volume 65. Vieweg+Teubner Veriag, 1999.

[21] L. Leifsson and S. Koziel. *Simulation-Driven Aerodynamic Design Using Variable-Fidelity models*. Imperial College Press, London, UK, 2015.

[22] J. Jacobs, L. Etman, F. Keulen, and J. Rooda. Framework for sequential approximate optimization. *Structural and Multidisciplinary Optimization*, 27:384–400, 2004.

[23] F. N. Foster and Dulikravich S. G. Three-dimensional aerodynamic shape optimization using genetic and gradient search algorithms. *Journal of Spacecraft and Rockets*, 34(1):36–42, January, 1997.

[24] R. M. Hicks and P. A. Henne. Wing design by numberical optimization. *Journal of Aircraft*, 15(7):407–412, 1978.

[25] J. W. Bandler, Q. S. Cheng, S. A. Dakroury, A. S. Mohamed, M. H. Bakr, K. Madsen, and J. Sondergaard. Space mapping: the state of art. *Microwave Theory and Techniques, IEEE Transactions*, 52(1):337–361, 2004.

[26] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidynathan, and P. K. Tucker. Surrogate-based analysis and optimization. *Progress in Aerospace Sciences*, 41:1–28, 2005.

[27] A. I. J. Forrester and A. J. Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 41(1–3):50–79, 2009.

[28] J. P. C. Kleijnen. Kriging metamodeling in simulation: a review. *European Journal of Operational Research*, 192:707–716, 2009.

[29] J. E. Rayas-Sanchez. Em-based optimization of microwave circuits using artificial neural networks: the state-of-the-art. *IEEE Trans. Microwave Theory Tech*, 52:420–435, 2004.

[30] N. M. Alexandrov, R. M. Lewis, C. R. Gumbert, L. L. Green, and P. A. Newman. Optimization with variable-fidelity models applied to wing design. In *38th Aerospace Science Meeting and Exibit*, Reno, NV, Jan. 2000.

[31] S. Koziel, D. Echeverria-Ciaurri, and L. Leifsson. "Surrogate-based methods" in S. Koziel and X. S. Yang (Eds.) *Computational Optimization, Methods and Algorithms, Series: Studies in Computational Intelligence*. Springer-Verlag, Berlin Heidelberg, 2011.

[32] L. Leifsson and S. Koziel. Multi-fidelity design optimization of transonic airfoils using shape-preserving response prediction. *Procedia Computer Science*, 1(1):1311–1320, 2010.

[33] S. Koziel, Q. S. Cheng, and J. W. Bandler. Space mapping. *IEEE Microwave Magazine*, 9(6):105–122, Dec. 2008.

[34] S. Koziel and L. Leifsson. Knowledge-based airfoil shape optimization using space mapping. In *30th AIAA Applied Aerodynamics Conference*, New Orleans, Louisiana, Jun. 25-28, 2012.

[35] S. Koziel and D. Echeverria-Ciaurri. Reliable simulation-driven design optimization of microwave structures using manifold mapping. *Progress in Electromagnetic Research B (PIER B)*, 26:361–382, 2010.

[36] S. Koziel and L. Leifsson. Surrogate-based aerodynamic shape optimization by variable-resolution models. *AIAA Journal*, 51(1):94–106, 2013.

[37] S. Koziel and L. Leifsson. Adaptive response correction for surrogate-based airfoil shape optimization. In *30th AIAA Applied Aerodynamics Conference*, New Orleans, Louisiana, Jun. 25-28, 2012.

[38] S. Koziel and L. Leifsson. Multi-fidelity airfoil optimization with adaptive response prediction. In *14th AIAA/ISSMO/ Multidisciplinary Analysis and Optimization Conference*, Indinanapolis, Indiana, Sep. 17-19, 2012.

[39] R. M. Lewis N. M. Alexandrov and. An overview of first-order model management for engineering optimization. *Optimization and Engineering*, 2(4):413–430, 2001.

[40] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust-Region Methods*. MPS-SIAM Series on Optimization, Philadelphia, PA, 2000.

[41] Mathworks. *Matlab primer*. The Mathworks Inc., Corporate Headquarters, 3 Apple Hill Drive, Natick, MA 01760-2098, USA, Sep. 2015.

[42] D. E. Goldberg. *Generic algorithms in search, optimization, and machine learning*. Addison-wesley, 1989.

[43] M. Clerc and J. Kennedy. The particle swarm-explosion, stability and convergence in a multidimensional complex space. *Evolutionary Computation, IEEE Transactions*, 6(1):58–73, 1995.

[44] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings, IEEE international conference*, volume 4, pages 1942–1948, 1995.

[45] O. Pironneau. On optimum design in fluid mechanics. *J. Fluid Mech.*, 64:97–110, 1974.

[46] J. R. Koehler and A. B. Owen. *Computer experiments, in Ghosh, S. and Rao, C. R. (eds), Handbook of Statistics*. Elsevier Science B.V., Amsterdam, 1996.

[47] A. A. Giunta. *Aircraft multidisciplinary design optimization using design of experiments theory and response surface modeling methods*. PhD thesis, Virginia Polytechnic Institute and State University, 1997.

[48] T. J. Santner, B. Williams, and W. Notz. *The Design and Analysis of Computer Experiments*. Springer-Verlag, Berlin, 2003.

[49] T. W. Simpson, J. Peplinski, P. N. Koch, and J. K. Allen. Metamodels for computer-based engineering design: survey and recommendations. *Engineering with Computers*, 17:129–150, 2001.

[50] A. A. Giunta, S. F. Wojtkiewicz, and M. S. Eldred. Overveiw of modern design of experiments methods for computational simulations. In *41st AIAA Aerospace Sciences Meeting and Exhibit*, volume AIAA 2003-0649, Reno, NV, January 6-9, 2003.

[51] M. McKay, W. Conover, and R. Beckman. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21:239–245, 1979.

[52] K. Q. Ye. Orthogonal column latin hypercubes and their application in computer experiments. *Journal of the American Statistical Association*, 93:1430–1439, 1998.

[53] K. Palmer and K. L. Tsui. A minimum bias latin hypercube design. *IIIE Transactions*, 33:793–808, 2001.

[54] B. Beachkofski and R. Grandhi. Improved distributed hypercube sampling. In *43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Denver, CO, April 22-25, 2002.

[55] S. Leary, A. Bhaskar, and A. Keane. Optimal orthogonal-array-based latin hypercubes. *Journal of Applied Statistics*, 30:585–598, 2003.

[56] D. G. Krige. A statistical approach to some basic mine valuations problems on the witwatersrand. *Journal of the Chemical, Metallurgical and Mining Engineering Society of South Africa*, 52(6):119–139, 1951.

[57] S. M. Wild, R. G. Regis, and C.A. Shoemaker. Optimization by radial basis function interpolation in trust-regions. *SIAM Journal on Scientific Computing*, 30:3197–3219, 2008.

[58] S. Haykin. *Neural Networks: A Comprehensive Foundation, 2nd ed.* Prentice Hall, Upper Saddle Rive, NJ, 1998.

[59] S. Geisser. *Predictive Inference*. Springer Science Business Media, B.V. Dordrecht, 1993.

[60] B. Efron. Estimating the error rate of a prediction rule: improvement on cross-validation. *Journal of American Statistical Association*, 78:316, 1983.

[61] T. D. Robinson, M. S. Eldred, K. E. Willcon, and R. Haimes. Surrogate-based optimization using multifidelity models with variable parameterization and corrected space mapping. *AIAA Journal*, 46(11):2316–2326, 2008.

[62] S. Koziel. Multi-fidelity multi-grid design optimization of planar microwave structures with sonnet. In *International Review of Progress in Applied Computational Electromagnetics*, pages 719–724, Tampere, Finland, 2010.

[63] D. Echeverria and P.W. Hemker. Manifold mapping: a two-level optimization technique. *Computing and Visualization in Science*, 11:193–206, 2008.

[64] S. Koziel, J. W. Bandler, and K. Madsen. A space mapping framework for engineering optimization: theory and implementation. *IEEE Trans. Microwave Theory Tech*, 54:3721–3730, 2006.

[65] S. J. Leary, A. Bhaskar, and A. J. Keane. A constraint mapping approac to the structural optimization of an expensive model using surrogates. *Optimization and Engineering*, 2:385–398, 2001.

[66] M. Redhe and L. Nilsson. Optimization of the new saab 9-3 exposed to impace load using a space mapping technique. *Structural and Mltidisciplinary Optimization*, 27:411–420, 2004.

[67] D. Echeverria and P. W. Hemker. Space mapping and defect correction. *Comp. Methods in Appl. Math*, 5(2):107–136, 2005.

[68] J. Ren, L. Leifsson, S. Koziel, and Y.A. Tesfahunegn. Multi-fidelity aerodynamic shape optimization using manifold mapping. In *54th AIAA Aerospace Sciences Meeting, Science and Technology Forum*, San Diego, CA, Jan 4-8, 2016.

[69] J. Ren, A. Thelen, A. Amrit, X. Du, L. Leifsson, Y.A. Tesfahunegn, and S. Koziel. Application of multi-fidelity optimization techniques to benchmark aerodynamic design problems. In *54th AIAA Aerospace Sciences Meeting, Science and Technology Forum*, San Diego, CA, Jan 4-8, 2016.

[70] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design*. Academic Press, Boston, MA, 1993.

[71] Pointwise. *Pointwise V17.3R4*, volume 76104–1107. Pointwise Inc., 213 South Jennings Avenue, Fort Worth, Texas, USA, 2015.

[72] F. Palacios, M. R. Colonno, A. C. Aranake, A. Campos, S. R. Copeland, T. D. Economon, A. K. Lonkar, T. W. Lukaczyk, T. W. R. Taylor, and J. J. Alonso. Stanford university unstructured (su2): An open-source integrated computational environment for multi-physics simulation and design. In *51st AIAA Aerospace Sciences Meeting and Exhibit*, volume 2013–0287, Grapevine, Texas, USA, 2013.

[73] P. R. Spalart and S. R. Allmaras. A one equation turbulence model for aerodynamic flows. In *38th AIAA Aerospace Sciences Meeting and Exhibit*, volume 92-0439, Reno, NV, January 6-9, 1992.

[74] A. Jameson, W. Schmidt, and E. Turkel. Numerical solution of the euler equations by finite volume methods using runge-kutta time-stepping schemes. In *AIAA 14th Fluid and Plasma Dynamic Conference*, volume 1981–1259, Palo Alto, CA, June 23-25, 1981.

[75] D. W. Kinsey and T. J. Barth. *Description of a Hyperbolic Grid Generation Procedure for Arbitrary Two-Dimensional Bodies*. Air force wright aeronautical labs wright-patterson AFB, OH, 1984.

[76] E. Torenbeek. *Advanced Aircraft Design: Conceptual Design, Technology and Optimization of Subsonic Civil Airplanes, Aerospace Series*. Wiley, Chichester, England, 2013.

[77] J. Vassberg, M. Dehaan, M. Rivers, and R. Wahls. Development of a common research model for applied cfd validation studies. In *26th AIAA Applied Aerodynamics Conference*, Honolulu, HI, Aug. 2008.

[78] E. Jonsson. *Aerodynamic Optimization by Variable-Resolution Modeling and Space Mapping*. Reykjavik University, Reykjavik, Iceland, 2012.