AN EVOLUTIONARY GEOMETRY PARAMETRIZATION FOR AERODYNAMIC SHAPE OPTIMIZATION

by

Xiaocong Han

A thesis submitted in conformity with the requirements for the degree of Masters of Applied Science Graduate Department of Aerospace Engineering University of Toronto

Copyright © 2011 by Xiaocong Han
Abstract

An Evolutionary Geometry Parametrization for Aerodynamic Shape Optimization

Xiaocong Han
Masters of Applied Science
Graduate Department of Aerospace Engineering
University of Toronto
2011

An evolutionary geometry parametrization is established to represent aerodynamic configurations. This geometry parametrization technique is constructed by integrating the classical B-spline formulation with the knot insertion algorithm. It is capable of inserting control points to a given parametrization without modifying its geometry. Taking advantage of this technique, a shape design problem can be solved as a sequence of optimizations from the basic parametrization to more refined parametrizations. Owing to the nature of the B-spline formulation, feasible parametrization refinements are not unique; guidelines based on sensitivity analysis and geometry constraints are developed to assist the automation of the proposed optimization sequence. Test cases involving airfoil optimization and induced drag minimization are solved adopting this method. Its effectiveness is demonstrated through comparisons with optimizations using uniform refined parametrizations.
Acknowledgements

I would like to express my gratitude to my supervisor, Professor David Zingg, for his instructive advice and valuable suggestions on my study. He has walked me through all the stages of my research. Without his consistent encouragement and guidance, this thesis could not have reached its present form.

Secondly, I would like to thank my peers at UTIAS, especially senior researchers, Jason Hicken, Timothy Leung, and Howard Buckley. I have had many enjoyable discussions with CFD lab members, and a lot of concepts in this thesis are inspired during the daily conversations.

Finally, my thanks goes to my beloved family for their continuous support and considerations all through these years. I also owe my sincere gratitude to my friends who offered their help and time to resolve the difficulties I encountered during my study at Canada.

Financial support has been generously provided by the National Sciences and Engineering Research Council of Canada, Canada Research Chairs, MITACS, and the University of Toronto.
# Contents

List of Tables vii

List of Figures viii

1 Introduction 1
   1.1 Motivation ........................................... 1
   1.2 Aerodynamic Shape Optimization .......................... 2
   1.3 Parametrization Methods .................................. 4
   1.4 Evolutionary Parametrization ............................. 6
   1.5 Objective ............................................. 7

2 B-spline Parametrization 9
   2.1 Parametric Space and Knot Vector ......................... 9
   2.2 B-spline Curve ......................................... 10
      2.2.1 B-spline basis function .............................. 10
      2.2.2 B-spline control points and curves .................... 12
      2.2.3 The derivative of a B-spline curve .................... 14
      2.2.4 Knot insertion ...................................... 14
      2.2.5 B-spline curve approximation ......................... 16
   2.3 B-spline Volume ........................................ 17

3 Overview of Optimization Routines 21
   3.1 Grid Movement Algorithm ............................... 21
   3.2 Flow Solver ........................................... 22
   3.3 Optimizer ............................................. 23
      3.3.1 Objective function .................................. 24
      3.3.2 Gradient evaluation ................................ 24
## 3.3.3 Optimization algorithm ........................................ 26
## 3.3.4 Optimization sequence ........................................ 28

### 4 Airfoil Optimization

4.1 Implementation of Knot Insertion ................................ 31
4.2 Test Cases .......................................................... 33
  4.2.1 Case 1: Lift-constrained drag minimization (subsonic) ..... 33
  4.2.2 Case 2: Lift-to-drag ratio maximization .................... 35
  4.2.3 Case 3: Lift-constrained drag minimization (transonic) .... 37
  4.2.4 Case 4: Multipoint optimization .............................. 38

### 5 Induced Drag Minimization

5.1 Test Case Setup .................................................. 41
5.2 Test case .......................................................... 42
  5.2.1 Case 1: Planform optimization .............................. 42
  5.2.2 Case 2: Planform and twist optimization .................. 44
  5.2.3 Case 3: Winglet optimization ............................... 46
  5.2.4 Case 4: Box-wing optimization ............................. 48
  5.2.5 Case 5: Wing optimization using adaptive constraints .... 50

### 6 Conclusions

6.1 Future Work .................................................... 55
  6.1.1 Generalization ............................................... 56
  6.1.2 Evolutionary parametrization with adaptation .............. 56
  6.1.3 Multiple insertions ........................................ 56
  6.1.4 Alternative parametrizations ................................ 57
  6.1.5 Multilevel optimization .................................... 57

References .......................................................... 59

### A Convergence Histories

A.1 Airfoil Optimization ............................................. 65
A.2 Induced Drag Minimization ....................................... 69
List of Tables

4.1 Thickness constraints for Case 1 ........................................ 33
4.2 Thickness constraints for Case 2 ........................................ 36
4.3 Geometric constraints for Case 3 ........................................ 37
5.1 Grid parameters ............................................................... 42
5.2 Surface parametrization and drag coefficient of each optimization cycle 52
List of Figures

2.1 Schematic representation of the recursive relation .................. 10
2.2 B-spline basis function with different knot vectors .................. 11
2.3 An open B-spline curve and its basis functions .................... 12
2.4 Local modification .......................................... 13
2.5 Convex hull property ........................................ 13
2.6 Knot insertion ............................................... 15
2.7 B-spline parameterization of the NACA0012 airfoil ................. 18
2.8 B-spline volume parametrization ................................ 20
2.9 Parametrization refinement of B-spline volumes ................... 20

3.1 Optimization sequence with evolutionary parametrization .......... 28

4.1 NACA0012 airfoil parametrized using 15 control points ............. 32
4.2 Knot sequence ............................................... 32
4.3 Optimized geometry and pressure distribution of Case 1 .......... 34
4.4 Objective function comparison of Case 1 .......................... 35
4.5 Optimized geometry and pressure distribution of Case 2 .......... 36
4.6 Objective function comparison of Case 2 .......................... 36
4.7 Optimized geometry and pressure distribution of Case 3 .......... 38
4.8 Objective function comparison of Case 3 .......................... 38
4.9 Optimized geometry and objective function comparison of Case 4 . 39
4.10 Pressure distribution for each operating point ..................... 40

5.1 Planform deformation ........................................ 43
5.2 Planform optimization ........................................ 43
5.3 Shape changes of the planform and twist optimization ............. 45
5.4 Planform and twist optimization ................................ 46
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.5 Winglet formation</td>
<td>47</td>
</tr>
<tr>
<td>5.6 Winglet optimization</td>
<td>48</td>
</tr>
<tr>
<td>5.7 Shape changes of the box-wing optimization</td>
<td>49</td>
</tr>
<tr>
<td>5.8 Box-wing optimization</td>
<td>50</td>
</tr>
<tr>
<td>5.9 Geometry deformation as the parametrization is gradually refined and the constraints are progressively released</td>
<td>54</td>
</tr>
<tr>
<td>A.1 Convergence histories for airfoil optimization case 1</td>
<td>66</td>
</tr>
<tr>
<td>A.2 Convergence histories for airfoil optimization case 2</td>
<td>67</td>
</tr>
<tr>
<td>A.3 Convergence histories for airfoil optimization case 3</td>
<td>68</td>
</tr>
<tr>
<td>A.4 Convergence histories for airfoil optimization case 4</td>
<td>69</td>
</tr>
<tr>
<td>A.5 Convergence histories for planform optimization</td>
<td>69</td>
</tr>
<tr>
<td>A.6 Convergence histories for planform and twist optimization</td>
<td>70</td>
</tr>
<tr>
<td>A.7 Convergence histories for winglet optimization</td>
<td>70</td>
</tr>
<tr>
<td>A.8 Convergence histories for box-wing optimization</td>
<td>71</td>
</tr>
<tr>
<td>A.9 Convergence histories for wing optimization using adaptive constraints</td>
<td>71</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Motivation

Air traffic has grown at 5% - 9% annually since 1960s, and it is expected to maintain such growth rates for the next few decades. The expansion of air transport causes increases in jet fuel consumption and aviation emissions. According to a special report submitted to the Intergovernmental Panel on Climate Change (IPCC) [43], the anticipated annual growth rate of fuel consumption, \( CO_2 \) and \( NO_x \) emissions are 2.9%, 2.9% and 3.3% by 2015. These figures reveal two important consequences:

1. Increasing jet fuel consumption speeds up the depletion of oil, and the scarcity of oil ultimately increases its price. Since fuel cost has become the most significant expense for airline companies since 2006, it is expected to be a major obstacle of the sustainable development of aviation industry.

2. Continuous emissions of \( CO_2 \) and \( NO_x \) in the short term create meteorological hazards such as acid rain and unhealthy aerosol particulates and in the long term accelerate global climate change.

Given these emerging problems, fuel efficiency and emission alleviation are significant factors that should be accounted for in the shape design of the next generation of aircraft. As pointed out by Mistry, Smith and Fielding [34], the current conventional configurations are designed and optimized to achieve the highest level of performance, and substantial improvements can only be made by adopting novel concepts. Thus, the shift from the existing wing-fuselage shape to unconventional configurations has become a major trend in aerodynamic shape design.


1.2 Aerodynamic Shape Optimization

With the purpose of exploring novel aerodynamic configurations, numerical shape optimization exhibits distinct advantages compared to the traditional “cut and try” and inverse design approaches. One significant advantage of optimization is that it reduces intervention from designers. Both the “cut and try” and inverse design approaches require extensive experience from designers to either manually alter the geometry or specify the target pressure distribution, but when facing unconventional configurations or unfamiliar design objectives, the manipulation based on empirical data could mislead or restrict the design results. In contrast, shape optimization combines a geometry control technique and an optimization algorithm to numerically seek the optimal design. Thus an optimal configuration can be found through systematically and effectively searching a design space. Another prominent benefit is its capability of addressing complex design problems. A practical design is required to have robust performance under a range of operating conditions and constraints, and numerical shape optimization is ideal to handle the situation consisting of multiple design requirements and conditions.

A fully automated aerodynamic shape optimization contains several key components: a geometry parametrization which defines the design variables and governs shape changes; a mesh movement algorithm that perturbs the computational grid according to the geometry deformation; an objective function definition that typically includes lift, drag, and moment functionals; flow analysis tool (flow solver), and optimization algorithm (optimizer). Each component influences the implementation efficiency and the optimal solution of a particular optimization process. One of the major factors is the flow solver. A flow solver is repeatedly used to obtain flow solutions for objective function evaluations. Thus high accuracy and fast computational speed is demanded. Owing to the improvement in computational fluid dynamics (CFD) and the development of computing capabilities, high fidelity analysis codes are now available to handle three-dimensional Reynolds-averaged Navier-Stokes equations within a reasonable amount of time, but solving complex nonlinear flow features such as laminar-turbulence transition still remains a costly task. Besides the flow solver, the optimizer also critically affects the performance of an optimization method. In most aerodynamic applications, two types of optimizers are adopted. One only makes use of the objective function value, for instance, genetic algorithms [32, 42]. This type of optimizer has a large probability of finding a global optimum, but converges relatively slowly [24]. The other requires additional gradient
evaluation of the objective function with respect to the design variables. This type of gradient-based optimizer is capable of finding an optimum with relatively low computational cost. However, it inherently converges to a local optimum and is limited to problems with a smooth design space.

Due to the limitations of aerodynamic shape optimization methods, advanced optimization strategies are needed to improve the optimal solution based on the existing numerical tools. A possible method is the hybrid optimization strategy presented by Vicini and Quagliarella [52]. They develop an optimization algorithm by adding a gradient-based technique to a set of operators of a multiobjective genetic algorithm, and it turns out that the computational efficiency of the hybrid algorithm is increased, while the desirable properties of genetic optimizer are preserved. Another applicable approach is the multilevel optimization technique which decomposes a complex problem into several easier sub-problems and solves them sequentially. One example can be referred to Alexandrov et al. [1]. They propose an approximation model management framework that integrates a gradient-based optimizer with variable fidelity flow analysis tools, and the optimization is accelerated by adopting the low fidelity model at some intermediate iterations.

The optimization strategy proposed in this thesis focuses on the effectiveness of design variables in an optimization problem. As indicated by a number of authors [4, 3, 33], the presence of an excessive number of design variables can result in poor performance for most existing optimization algorithms. Thus, it is essential to only include a limited number of critical design variables in an optimization. However, for general optimization problems, the design space features are not obvious a priori, and there are no guidelines on choosing proper design variables. Therefore, prescribing design variables by a designer is not an ideal treatment for an automated optimization process. Especially when investigating unconventional configurations, a predetermined design space could limit the capability of an optimizer and should be minimized. The better alternative is to strategically introduce design variables during an optimization process based on the performance of the existing geometry. In other words, a design problem is organized as a succession of optimizations with an increased number of design variables.

To achieve such an optimization sequence, the set of design variables should be consistent, flexible and easy to manipulate. Since the definition of the design variables is provided by a geometry parametrization, an ideal parametrization method is required to progressively evolve as an optimization proceeds. This type of parametrization strategy is
referred to as evolutionary parametrization in this thesis. The following two sections are
devoted to a brief review of existing geometry parametrization techniques and previous
applications and constructions of evolutionary parametrization.

1.3 Parametrization Methods

The choice of geometry parametrization is a crucial procedure in shape optimization. A
number of authors have outlined the desirable characteristics of an ideal parametrization
method [47, 30, 11], and several essential criteria are worth additional emphasis.

- A parametrization should be able to provide fast, accurate and consistent repre-
sentations for complex geometries

- A parametrization should produce a compact and flexible set of design variables

- A parametrization should provide easy control and interpretation for geometry
deforation

A large variety of parametrization methods have been established for various appli-
cations [47, 6, 38]. An intuitive method is the discrete approach which uses surface grid
point coordinates as design variables. This approach is able to describe a large number
of dramatically different geometries and can also reflect subtle shape changes in a local
region. However, since the shape change is tied to individual grid points, it is difficult
to maintain a smooth surface profile, and there is a tendency to yield unrealistic designs
as indicated by Braibant and Fleury [5]. Another obstacle associated with this method
is the excessive number of design variables. Because the difficulty of converging an op-
timization increases quickly with the number of design variables, the efficiency of the
shape optimization is adversely impacted if a fine mesh is adopted. These problems are
well understood; some complementary procedures such as a smoothing routine have been
established in many works [36, 35].

On the other hand, some strategies, such as extended Joukowski transformation [28]
and PARSEC [42], can produce a compact set of design variables, but these methods are
restricted to represent the shape of an airfoil, and lack generality for arbitrary complex
configurations. In addition, the design variables used in such a parametrization must be
explicitly specified, which leaves little flexibility for a designer to adjust the design space
according to specific requirements and conditions.
The domain element approach [37] is a variation of the parametrization using grid point coordinates. It groups a set of grid points to a macro unit (domain element), and such grouping applies to the entire grid. Usually the vertices of the domain element are treated as design variables which controls the shape deformation, and the internal grid points belonging to the domain element move according to a certain inverse mapping that preserves the parametric coordinates of the grid points with respect to the domain element.

For typical airfoil and wing design, the geometry can also be represented by a linear combination of basis shape functions and the coefficients of these basis functions are regarded as the design variables. In this approach, a sufficient small number of design variables is needed to describe a geometry, but the design space is restricted by the choice of prescribed basis functions. Normally existing geometries with some perturbation functions are adopted as the basis functions, for instance, Hicks-Henne bump functions [23], but to reduce the linear dependence among the basis functions, some authors [49, 46] also use orthogonal basis functions. Nevertheless, the optimal geometry is highly dependent on the formulation of the basis functions.

Class / shape function transformation (CST) is another recently proposed parametrization by Kulfan and Bussoletti [30]. It represents a complex geometry such as an airfoil with a round-nose and a sharp aft-end using analytic and well-behaved mathematical functions. A considerable advantage of this approach is that conventional design parameters like leading edge radius, camber and trailing edge angle etc. can be specified through modifications of the local class function. However, the automation of this approach at the current stage is not ideal; experimentation and tuning are necessary for specific geometry representations.

A commonly used approach in computer aided geometric design is based on polynomial spline functions, specifically, Bezier and B-spline techniques [8, 26, 13]. These two types of spline approaches share some common favourable properties. For instance, both of them use the tensor product of polynomial basis functions and coefficient vectors to analytically represent a shape. Since the basis functions are invariant for a prescribed knot vector, the coefficient vectors, termed control points, are naturally chosen as design variables to generate shape changes. Also, for every parametrized shape, the control points form a convex hull which contains and mimics the geometry. This property is particularly useful for mesh movement. It allows costly mesh deformation techniques to move relatively few control points, and an inexpensive algebraic method to adjust a large
number of mesh points [22]. Besides these common properties, there exist unique features for individual approach. Regarding Bezier curves and surfaces, two characteristics of the basis functions limit their flexibility. First, the number of control points of a Bezier curve or surface is associated with the degree of the basis functions. In order to increase the number of control points, degree elevation of the basis functions is necessary [26, 13]. Second, a Bezier curve or surface does not provide local control. i.e. moving a control point changes the entire geometry.

The B-spline formulation [8] overcomes the mentioned disadvantages because its intrinsic basis function formulation provides local control of a parametrized shape, and its flexible knot vector structure allows additional control points to be introduced without increasing the degree of basis functions. Hence, the B-spline approach is an appropriate candidate for hierarchical parametrization. An additional advantage of the B-spline approach is that it is used in most CAD packages to represent the geometry. Therefore, it provides the most natural way to integrate the CAD geometry into the design process.

1.4 Evolutionary Parametrization

An evolutionary parametrization is a sequence of parametrization refinements that gradually enlarge the set of design variables. Such a procedure relies on the characteristics of the geometry representation method. Two essential conditions are required for such a procedure:

- Multiple parametrization refinements can be carried out in a consistent manner;
- The geometry should not be changed as its parametrization is refined.

Attempts at performing optimization with a changing parametrization have been carried out by several authors. Beux and Dervieux [4] describe a gradient based multilevel optimization using surface grid point coordinates as design variables. The hierarchical parametrization is defined by extracting different subsets of grid points from the complete surface points, forming a family of embedded parametrization levels (i.e. a coarse level corresponds to a small number of grid points, and a fine level refers to a large number of points). A linear prolongation operator defined by Hermitian interpolation is used to switch between different parametrization levels. The objective function and the gradient are evaluated at the finest level and projected to a particular coarse level where an optimization occurs. They also propose two optimization sequences: first, the optimizations
are conducted sequentially from the coarsest level to the finest level; second, multigrid strategies are adopted. The optimizations are performed at the various levels according to a full V-cycle sequence. This approach displays an increase of efficiency primarily due to the faster convergence by considering less design variables in coarse parametrization levels.

Further investigation is done by Martinelli and Beux [33]. They extend the described strategy to other kinds of parametrizations. instead of linear interpolation, affine operators are defined to project and prolong the gradient among different parametrization levels, thus the design variables are not restricted to the grid points. They present two hierarchical parametrizations: the first case is constructed using orthogonal basis functions, and each sub-level corresponds to a different number of basis functions; the second case is built on the degree elevation property of the Bezier curve, and every sub-level refers to a certain number of Bezier control points.

Similar research has also been accomplished by Desideri and colleagues [9, 3, 12]. Their evolutionary parametrization strategy is defined by Bezier curves and volumes with a degree elevation algorithm. In these works, optimizations are carried out independently on different parametrizations following a predetermined sequence. Therefore, gradient-based and gradient-free optimization algorithms are all applicable. In addition, to increase the geometric regularity of a refined parametrization, they also propose a self-adaptive procedure [31, 10] associated with the degree elevation mechanism.

Other geometry representation methods can also be modified to contain multiple sets of design variables. For instance, Morris et al. [37] formulate a hierarchical domain element method which combines conventional planform variables of a wing with different level of parametrization. Despite the different features associated with various geometry representations, the primary idea of all evolutionary parametrizations is the same: the optimization results can be improved using flexible design variables.

1.5 Objective

The objective of the current work is to construct an evolutionary geometry parametrization based on an existing B-spline curve formulation for an airfoil [40] and B-spline volume parametrization for general three-dimensional objects [22]. This process consists of a B-spline approximation and a series of knot insertions. The difference between this approach and the above methods is that the knot insertion procedure is not unique due to
Chapter 1. Introduction

the intrinsic properties of B-spline formulation. Therefore, parametrization refinements cannot be predetermined but can be selected during the optimization process. As a result, the proposed optimization is carried out sequentially from the initial parametrization to more refined parametrizations as long as the objective function continues to improve.

The proposed process is applied to different aerodynamic shape optimization problems, and its implementation is adjusted by considering different design problems. For airfoil optimizations, special considerations are devoted to geometric constraints; while for three-dimensional wing optimizations, different control point coupling methods are employed for different design purposes. The expected benefits of this process are mainly twofold. First, it eliminates the assumptions of the number and locations of design variables before an optimization, and reduces the need for intervention from a designer during an optimization. Thus it may yield geometries not anticipated. Second, this process has the potential to achieve a more optimal design with fewer design variables compared to optimizations using a regular parametrization strategy.
Chapter 2

B-spline Parametrization

When a B-spline parametrization is employed in a shape optimization, the coordinates of its control points are normally used as design variables. Therefore, constructing an evolutionary parametrization requires a geometry to be represented by different numbers of control points. This can be achieved through an initial B-spline approximation, and multiple knot insertions. This chapter is devoted to the explanation of the mathematical formulation and the implementation of the evolutionary B-spline parametrization in aerodynamic shape optimization problems.

2.1 Parametric Space and Knot Vector

The B-spline formulation is a type of parametric representation. Thus, in simple words, it defines a mapping from a parametric space to a physical space, \( \{ f : \xi \rightarrow X \} \); both the parametric and physical space can be multidimensional, and the mapping function consists of a series of polynomials. Take \( \{ \xi \in \mathbb{R} \} \) as an example. The domain of the parametric space can be set to any arbitrary values, and by convention \([0, 1]\) is normally assumed. To properly construct the mapping functions, a partition of the parametric domain is defined by dividing the domain into several intervals, and the vector containing this partition, \( t = \{ t_1 \leq t_2 \leq \ldots \leq t_{m-1} \leq t_m \} \), is referred to as the knot vector. The half-open interval, \([t_i, t_{i+1})\) is the i-th knot interval. Since some \( t_i \)'s may be equal, some knot intervals may not exist. If \( t_i \) appears \( l \) times, then it is a multiple knot of multiplicity \( l \). In general, for a multidimensional parametric space, there exist different knot vectors for each individual dimension.
2.2 B-spline Curve

A planar B-spline curve, \( \{ \xi \in \mathbb{R} \rightarrow X \in \mathbb{R}^2 \} \), is the simplest geometry representation using B-spline approach, but it is effective to describe a two-dimensional object such as an airfoil. In this section, some critical B-spline properties are introduced in the context of a planar B-spline curve, but they also hold in a more general situation.

2.2.1 B-spline basis function

B-spline basis functions are polynomials of order \( k \) defined in a parametric space by the following recursive relation:

\[
N_{i,1}(\xi) = \begin{cases} 
1 & \text{if } t_i \leq \xi \leq t_{i+1} \\
0 & \text{otherwise} 
\end{cases} \quad (2.1)
\]

\[
N_{i,k}(\xi) = \frac{\xi - t_i}{t_{i+k-1} - t_i} N_{i,k-1}(\xi) + \frac{t_{i+k} - \xi}{t_{i+k} - t_{i+1}} N_{i+1,k-1}(\xi) \quad (2.2)
\]

The above is usually referred to as the Cox-de Boor recursion formula [8]. To visualize this recursive relation, a triangle scheme is made in Figure 2.1. From the above figure, two important observations can be made. First, if one traces down from a particular basis function to the lowest level, then it can be seen that a basis function, \( B_{i,k} \), is non-zero at most on the interval, \([t_i, t_{i+k}]\); second, if one starts with an interval, \([t_i, t_{i+1}]\), and search for the basis functions it contributes to, then it is easy to identify that at most \( k \) non-zero basis functions of order \( k \) are non-zero at this interval. The term “at most” is used because the presence of multiple knots would reduce the numbers, and these statements describe the upper limits.
To illustrate the impact of multiple knots, 4th-order basis functions computed based on three knot vectors are compared in a schematic diagram, Figure 2.2. Figure 2.2(a) shows a vector consisting of only simple knots, Figure 2.2(b) displays the effect of moving $t_5$ to $t_6$, and Figure 2.2(c) shows the result if both $t_5$ and $t_7$ are moved to $t_6$. The above statements that one basis function spans a finite number of knot intervals and one knot interval contains a finite number of non-zero basis functions are clearly depicted. Because of them, B-spline parametrization is locally supported.

Besides the local support property, a few characteristics of B-spline basis functions also can be deduced from these plots.

- The knot intervals close to the boundary are not fully supported (e.g. only one non-zero basis function exists for interval, $[t_1, t_2]$). Thus, to overcome this problem, one could put $k$ repeated knots at both ends of the knot vector, where $k$ is the order of the basis functions. This could result in an open B-spline curve to be discussed in the next section. On the other hand, one can make a periodic knot vector to avoid boundary problems, and this will produce a closed B-spline curve \[26\].

- The numbers of non-zero basis function at a knot is $k - l$, where $l$ is the multiplicity.

- At a knot of multiplicity $l$, the non-zero basis functions are $C^{k-l-1}$ continuous.

- At any point in the parametric domain, the sum of all non-zero basis functions is unity. This is usually referred to as the partition of unity, and a rigorous mathematical proof is given by de Boor [8].

These properties hold for multidimensional mappings, and all of them will play a part when constructing a B-spline representation.
2.2.2 B-spline control points and curves

A B-spline curve can be written as a linear combination of basis functions weighted by proper control points:

\[ X(\xi) = \sum_{i=0}^{1} d_i N_{i,k}(\xi) \]  

where \( X \) represents the B-spline curve, and the coefficient vectors, \( \{d_i : i = 1, \ldots, n\} \), are the coordinates of the control points. Since the parametrization in aerodynamic shape optimization extensively uses open B-spline curves, therefore, the following discussion focuses on open B-spline curves, and the boundary problem of the knot vector is resolved by placing repeated knots at both ends. Figure 2.3(a) displays a 4-th order open B-spline curve and Figure 2.3(b) shows the basis functions and parametric domain partition. The associated knot vector is \( \{0, 0, 0, 0, 0.05, 0.19, 0.39, 0.61, 0.81, 0.95, 1, 1, 1, 1\} \).

B-spline curves possess many favourable properties that can be used to construct a desired geometry. Some of them are a direct consequence of the characteristics of the basis functions. These properties are summarized as follows:

- An open B-spline curve passes through the two end control points \( d_1 \) and \( d_n \). This is due to the fact that the two basis functions at each end of the parametric domain have the value of unity, thus Eq. 2.3 is reduced to \( X(\xi = 0, 1) = d_i, i = 1, n \).

- Changing the position of a particular control point affects the shape of the B-spline curve locally. Since a control point is the coefficient of its corresponding basis
function, it will only affect the knot intervals where the basis function is non-zero. Figure 2.4 gives a clear illustration of this property.

- If the order of the basis function is $k$, each segment of the B-spline curve lies in the convex hull of the associated $k$ control points. Globally, the entire B-spline curve lies in the convex hull of all the control points. Figure 2.5(a) shows the convex hull for the curve segment associated with the first 4 control points, and Figure 2.5(b) indicates the convex hull for the entire curve.

- For a planar B-spline curve, no straight line intersects a B-spline curve more times than it intersects the control polygon which is formed by connecting control points.
sequentially. This reveals an important property that the B-spline curve is no more complex than the control polygon. If the variation of the control polygon is reduced, then the regularity of the curve is improved. This property also holds for Bezier parametrization, and a couple of authors [50, 7, 11] have considered a self-adaptive parametrization in optimization problems based on this property.

### 2.2.3 The derivative of a B-spline curve

To determine the tangent vectors of a B-spline curve, it is necessary to compute its derivative. Since the basis functions are merely polynomials, the derivatives can be computed as follows:

$$\frac{d}{d\xi} N_{i,k} = \frac{k-1}{t_{i+k-1} - t_i} N_{i,k-1} - \frac{k-1}{t_{i+k} - t_{i+1}} N_{i+1,k-1} \quad (2.4)$$

Substituting these derivatives into the curve equation, the derivative of a B-spline curve is formulated by

$$\frac{d}{d\xi} X(\xi) = \sum_{i=1}^{n-1} N_{i+1,k-1} \frac{k-1}{t_{i+k} - t_{i+1}} (d_{i+1} - d_{i}) \quad (2.5)$$

Observing the above equation, a direct consequence can be drawn for a open B-spline curve: the derivatives of the starting and end points can written as $$\frac{k-1}{t_{k+1} - t_2} (d_2 - d_1)$$ and $$\frac{k-1}{t_{n+k-1} - t_n} (d_n - d_{n-1})$$. Considering the fact that these two end points coincide with the first and last control points, thus the B-spline curve is tangent to $$(d_2 - d_1)$$ and $$(d_n - d_{n-1})$$. This property is useful if one considers connecting two B-spline curves. As long as the the joint control point and its adjacent two points are kept aligned, a smooth transition between the B-spline curves is achieved.

### 2.2.4 Knot insertion

So far, a B-spline curve is defined, and its control points are normally chosen as design variables in an optimization problem to govern the shape deformation. In order to form a evolutionary parametrization suitable for sequential optimization, a B-spline curve should be defined with a flexible number of control points, and its shape should be preserved if the number of control points changes. This requirement is accomplished through inserting an additional knot into the knot vector.

Inserting a new knot obeys the following procedure. Denote the new set of control points with a superscript $\ast$, if a knot $t^\ast$ is added to $$(t_r, t_{r+1})$$, the new control points
Chapter 2. B-spline Parametrization

\{d_i^*: i = 1, \ldots, r - k + 1, r + 1, \ldots, n + 1\} retain the positions of the old control points based on the local support property of B-spline formulation:

\[ d_i^* = d_i \quad 1 \leq i \leq r - k + 1 \]  \hspace{1cm} (2.6)

\[ d_i^* = d_{i-1} \quad r + 1 \leq i \leq n + 1 \]  \hspace{1cm} (2.7)

The new control points, \{d_i^*: i = r - k + 2, \ldots, r\}, are placed on the control polygon formed by the old control points, \{d_i^*: i = r - k + 1, \ldots, r\}. The quantitative relation can be deduced using de Boor algorithm [8]:

\[ d_i^* = (1 - \alpha_i) d_{i-1} + \alpha_i d_i \]  \hspace{1cm} (2.8)

\[ \alpha_i = \frac{t^* - t_i}{t_{i+k-1} - t_i} \]  \hspace{1cm} (2.9)

Figure 2.6(a) shows the effect of inserting a knot, 0.5, into the knot vector stated in Section 2.2.2. The resulted new control points are located at the old control polygon, and the B-spline curve is maintained. Figure 2.6(b) displays the variation of the basis functions. Compared to Figure 2.3(b), the modified basis functions are coloured. The above procedure is able to create new control points located at the old control polygon, but their exact coordinates are unknown in advance. If a user specified control point \( d^* \) between \((d_r, d_{r+1})\) is required, this procedure can be reversed to calculate the required
knot [44]:

\[ s = \frac{d^* - d_r}{d_{r+1} - d_r} \quad (2.10) \]

\[ t^* = t_{r+1} + s(t_{r+k} - t_{r+1}) \quad (2.11) \]

### 2.2.5 B-spline curve approximation

In the context of aerodynamic shape optimization, the baseline geometry is usually not defined by a B-spline formulation, but in terms of the coordinates of surface grid points. Therefore, converting this representation into B-spline form is a preliminary step. In this section, based on the work of Nemec [39], the procedure of representing a given airfoil with a 4th-order open B-spline curve is outlined.

The first essential factor is the choice of the parameter that maps each surface point on the airfoil to the parametric domain. As discussed in the work of Kulfan [30], for an airfoil with a round nose, the following centripetal parametrization [13] gives desirable results:

\[ \xi_1 = 0 \quad (2.12) \]

\[ \xi_j = \frac{n - k - 1}{L_T} \sum_{m=1}^{j-1} \sqrt{L_m} \quad j = 2, \ldots, N \quad (2.13) \]

where \( N \) is the total number of surface points on the airfoil, \( L_m \) is the segment length between successive points, and the normalization factor, \( L_T \) is given by

\[ L_T = \sum_{m=1}^{N-1} \sqrt{L_m} \quad (2.14) \]

The above mapping defines a parametric domain, \( \xi \in [0, n - k - 1] \). The construction of the knot vector also significantly impacts the quality of an approximation. Many authors [50, 26] have experimented with different choices. The cosine function adopted here turns out be a good compromise between robustness and accuracy:

\[ t_i = \begin{cases} 
0 & 1 \leq i \leq k \\
n - k - 1 & 1 \leq i \leq k \\
n - k - 1 \left[ 1 - \cos \left( \frac{i-k}{n-k-1} \pi \right) \right] & k + 1 \leq i \leq \frac{n+k-3}{2} \\
n - k - 1 & \frac{n+k-1}{2} \leq i \leq \frac{n+k+3}{2} \\
n - k - 1 \left[ 1 - \cos \left( \frac{i-k-2}{n-k-1} \pi \right) \right] & n + 1 \leq i \leq n \\
n - k - 1 & n + 1 \leq i \leq n + k 
\end{cases} \quad (2.15) \]
Note the multiple knots appearing at the middle of the knot vector; they ensure one control point is placed at the leading edge of the airfoil. However, because the multiplicity of a knot decreases the continuity of the associated curve, the continuity reduction at the leading edge is unwanted. To overcome this problem, two adjacent control points are placed colinearly with the leading edge point, so that $G^1$ continuity is restored. This amendment is performed at the end of the approximation.

Once the parametric domain is fully defined, the basis functions can be subsequently computed according to Eq. 2.2, and the last step is to determine the control points. Assume the airfoil surface is defined by a set of points, \( \{P_j, j = 1, \ldots, N\} \). The control points can be computed by solving the least squares problem, \( \min \sum_{j=1}^{N} ||P_j - X_j||_2 \). The obtained approximation curve \( \{X_j, j = 1, \ldots, N\} \) through this procedure has a well-known deficiency that the error vector, \( P_j - X_j \), is usually not perpendicular to the tangent. To resolve this problem, Hoschek [25] proposes the following iterative parameter correction algorithm:

\[
\bar{\xi}_j = \xi_j + \Delta c_j \frac{t_n - t_1}{L} \\
\Delta c_j = (P_j - X_j) \cdot Y_j
\] (2.16) (2.17)

where \( L \) is the total length of the control polygon defined by connecting the control points, and \( Y_j \) is the normalized tangent vector that can be found by computing the derivative of the B-spline curve.

One example of the approximation using 4th-order B-spline curve is shown in Figure 2.7(a). The NACA0012 airfoil is described using 15 control points, and the leading edge is handled by keeping three adjacent control points aligned. The figure beside it illustrates the application of the knot insertion algorithm. One knot insertion takes place at the interval \( (t_{13}, t_{14}) \) on the upper surface, splitting this knot interval into two parts that contain approximately same number of parameters. Another inverse knot insertion occurs on the lower surface, generating a new control point located at \( x = 0.5 \).

### 2.3 B-spline Volume

The B-spline volume parametrization developed by Hicken and Zingg [17] is employed in the three-dimensional shape optimization problems. Instead of parametrizing only the surface of the object, this approach represents the entire computational grid with
B-spline parametrization

Figure 2.7: B-spline parameterization of the NACA0012 airfoil. Plot (a) depicts the parametrized surface and control points; plot (b) shows the refined parametrization with the additional control points.

B-spline volumes and control points. Thus, the shape deformation is acquired through the movement of the B-spline surface patches, and the corresponding grid perturbation is driven by the adjustment of the volume control points to conform to the surface changes. The B-spline volume method is formulated by extending planar B-spline curves into multi-dimensional space and it inherits all the geometric properties from B-spline curves. In the discussion of the B-spline volume formulation, emphasis is given to the approximation of the computational grid and the implementation of the knot insertion algorithm.

The B-spline volume parametrization defines a mapping from a parametric space, \( \{ \xi = (\xi, \eta, \zeta) \in \mathbb{R}^3 : (\xi, \eta, \zeta) \in [0, 1] \} \) to the physical space \( \{ X(\xi) \in \mathbb{R}^3 \} \). The tensor product representation is given by

\[
X(\xi) = \sum_{i=1}^{n_i} \sum_{j=1}^{n_j} \sum_{k=1}^{n_k} d_{ijk} \mathcal{N}_{i,p_i}(\xi) \mathcal{N}_{j,p_j}(\eta) \mathcal{N}_{k,p_k}(\zeta)
\]

Here, the B-spline volume \( X(\xi) \), and the control points, \( d_{ijk} \), have analogous characteristics to the B-spline curve and control points. However, the mapping parameter \( \xi \) is defined using a traditional chord length parametrization for the purpose of approximating structured multi-block grids. The basis functions are defined separately for each parameter. Theoretically, the orders of the basis functions are independent for different parameters, but in practical applications, the order of polynomial basis functions is usu-
ally set to be the same for each parameter to maintain the same continuity condition. Hence, the order of the basis functions is denoted by $p$ for the rest of this work. The computation of the basis functions still refers to the recursive definition but with spatially varying knot vectors. Taking basis functions in the $\xi$ direction as an example, they are given by

$$
\begin{align*}
N_{i,1}(\xi) &= \begin{cases} 
1 & t_i(\eta, \zeta) \leq \xi \leq t_{i+1}(\eta, \zeta) \\
0 & \text{otherwise}
\end{cases} \\
N_{i,p}(\xi) &= \left( \frac{\xi - t_i(\eta, \zeta)}{t_{i+p-1}(\eta, \zeta) - t_i(\eta, \zeta)} \right) N_{i,p-1}(\xi) \\
&\quad + \left( \frac{t_{i+p}(\eta, \zeta) - \xi}{t_{i+p}(\eta, \zeta) - t_{i+1}(\eta, \zeta)} \right) N_{i+1,p-1}(\xi)
\end{align*}
$$

where the knot vector, $t_i(\eta, \zeta)$, is a spatially varying function of parameters, $\eta$ and $\zeta$. Its form can be arbitrarily set to accommodate the different geometries, as long as it remains non-decreasing. For the grids made of hexahedra, a simple bilinear form is used [17]:

$$
t(\eta, \zeta) = t(0,0)(1-\eta)(1-\zeta) + t(1,0)\eta(1-\zeta) + t(0,1)(1-\eta)\zeta + t(1,1)\eta\zeta
$$

Here $t(0,0)$, $t(1,0)$, $t(0,1)$ and $t(1,1)$ are four edge knot vectors in the $\xi$ direction, and they are constructed to have roughly the same number of parameters in each knot interval.

The B-spline volume control points are determined by solving Eq. 2.18 through a least squares procedure. For multi-block grids, consistent positions of control points at interfaces are mandatory. Thus the least squares problem is solved sequentially for block edges, surfaces, and volumes. The resulting B-spline volume control points are also referred to as a volume control mesh. Figure 2.8 shows a B-spline volume grid and its corresponding control mesh.

To form an evolutionary parametrization, B-spline volumes should be constructed with different numbers of control points. From Eq. 2.21, one can observe that the edge knot vectors are spatially invariant; this implies that the knot insertion algorithm for B-spline curves is still applicable to these edge knot vectors. Since the grids used in this work consist of hexahedra, once the four edge knot vectors in the same direction are refined simultaneously, all the other knot vectors in this direction can be subsequently refined using Eq. 2.21. Hence, by re-solving the least squares problem, a new B-spline volume can be established with a refined control mesh. Referring to the local support property of the B-spline formulation, only some of the control points will be altered.
Chapter 2. B-spline Parametrization

Figure 2.8: B-spline volume parametrization

Figure 2.9: Parametrization refinement of B-spline volumes. The white spheres in plot (a) are the original control points and the red spheres in plot (b) are the refined control points.

on the edges, which implies that the change to the control mesh will be limited to the corresponding sections. Figure 2.9 illustrates a control mesh refinement of a rectangular wing in the spanwise direction (one more section is added).
Chapter 3

Overview of Optimization Routines

In order to perform an automated aerodynamic shape optimization, a set of analysis and optimization routines are integrated together. The present optimization codes in two and three dimensions contain the following essential components:

1. Geometry parametrization
2. Grid perturbation algorithm
3. Flow solver
4. Optimizer

Among them, the geometry parametrization using the B-spline formulation is described in detail in the previous chapter. An overview of the remaining numerical routines is given in the following sections.

3.1 Grid Movement Algorithm

Once a geometry has been modified by manipulation of the surface control points, the surrounding computational grid has to conform to the new shape. In two-dimensional airfoil optimization problems, the movement of the airfoil surface is usually moderate and does not involve translation and rotation, so an algebraic grid perturbation method is sufficient and effective. Nemec et al. [41] propose the following algorithm based on the normalized arclength from the surface.

\[
\begin{bmatrix}
\Delta y \\
\Delta x
\end{bmatrix}
= \begin{bmatrix}
y \\
x
\end{bmatrix} + [1 + \cos(\pi S)] \begin{bmatrix}
\Delta y \\
\Delta x
\end{bmatrix}
\]  (3.1)
where \( \Delta y \Delta x \) represents the shape change, and \( S \) is the normalized arclength from the airfoil surface. The normalization factor is the total arclength from the airfoil surface to the outer boundary, so the grid boundary is preserved during a perturbation.

In three-dimensional wing optimization problems, the shape change is more drastic. Hicken and Zingg [22] develop a semi-algebraic method which first moves the B-spline control mesh using an incremental linear elasticity method [51], and regenerates the computational grid based on the perturbed control mesh afterwards. The linear elasticity equations are discretized using a finite element method, the resulting mesh movement residual equation is given by

\[
\mathbf{r}^{(i)} = \mathbf{r}^{(i)}(\mathbf{b}^{(i)}, \mathbf{b}^{(i-1)}) = K^{(i)}(\mathbf{b}^{(i)} - \mathbf{b}^{(i-1)}) - \mathbf{f}^{(i)}, \quad i = 1, \ldots, m
\]  

where \( \mathbf{r}^{(i)} \) is the residual, \( K^{(i)} \) is the stiffness matrix whose elements are defined by the spatially varying Young’s modulus, which is determined by the volume and distortion of each element, \( \mathbf{b}^{(i)} \) are the control points vector, and \( \mathbf{f}^{(i)} \) is the discrete force defined implicitly by the displacements of the surface and boundary control points. This entire procedure is done in \( m \) increments; for the optimization examples presented in this thesis, \( m \) is fixed to be 5.

### 3.2 Flow Solver

A high-fidelity flow solver is used to evaluate the aerodynamic performance of the existing geometry. For a two-dimensional airfoil, an efficient Newton-Krylov flow solver was developed by Nemec and Zingg [40] to solve the discretized Reynolds-averaged Navier-Stokes equations with Spalart-Allmaras turbulence model. The governing equations in conservative form are given by

\[
\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{E}}{\partial x} + \frac{\partial \mathbf{F}}{\partial y} = \frac{\partial \mathbf{E}_v}{\partial x} + \frac{\partial \mathbf{F}_v}{\partial y} 
\]  

where \( \mathbf{Q} = [\rho, \rho u, \rho v, e]^T \) are the conservative variables of the flow field, \( \mathbf{E} \) and \( \mathbf{F} \) are the inviscid fluxes, \( \mathbf{E}_v \) and \( \mathbf{F}_v \) are the viscous fluxes. The governing equations are simplified using the thin-layer approximation and transformed to computational space by incorporating metric terms (see the manuscript by Pulliam and Zingg [45] for details). The second-order centred finite-difference scheme is used for the spatial discretization,
and the temporal derivative is neglected for steady flows. Thus the governing equations are reduced to a set of nonlinear algebraic equations, \( \mathbf{R} = 0 \). Here, \( \mathbf{R} \) is referred to as the flow residual.

The residual equations are solved using a Newton-Krylov method. For each Newton iteration, an update of the conservative variables is found by the following equation:

\[
\mathbf{A}^{(n)} \Delta \mathbf{Q}^{(n)} = -\mathbf{R}^{(n)}
\]

(3.4)

where \( \mathbf{A}^{(n)} \) is the flow Jacobian matrix. Because convergence of the Newton method depends on the initial values of the flow variables at all nodes, a start-up algorithm is necessary to provide a proper initial iterate. In this solver, an implicit Euler time marching method with approximate factorization is used as the start-up algorithm. Once the initial values are obtained, Eq. 3.4 is solved inexactly using the generalized minimal residual (GMRES) linear solver.

For the optimization code for a three-dimensional wing, a Newton-Krylov-Schur flow solver was developed for the three-dimensional Euler equations by Hicken and Zingg [21]. The governing equations are given by

\[
\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{E}_i}{\partial x_i} = 0
\]

(3.5)

where \( \mathbf{Q} = [\rho, \rho u_1, \rho u_2, \rho u_3, e]^T \) are the conservative variables, and \( \mathbf{E}_i = [\rho u_i, \rho u_1 u_i + p \delta_{1i}, \rho u_2 u_i + p \delta_{2i}, \rho u_3 u_i + p \delta_{3i}, (e + p) u_i]^T \) are the inviscid fluxes. The Euler equations are discretized using a second-order accurate summation-by-parts operator and simultaneous approximation terms (details can be found in the paper by Hicken and Zingg [21]). If the temporal term is neglected, the Euler equations are reduced to nonlinear algebraic equations of the form, \( \mathbf{R} = 0 \). Solving the residual equation adopts the same Newton-Krylov strategy, but the start-up algorithm is changed to the dissipation based homotopy continuation [19], and the Krylov solver is changed to flexible GMRES with an approximate-Schur preconditioner.

### 3.3 Optimizer

Once the design variables are defined by a geometry parametrization, and the flow analysis is obtained by a flow solver, an efficient optimizer is necessary to determine the way to modify the design variables, such that the design objective is improved. Although there exist many optimization algorithms that only require the objective function value,
the most computationally efficient algorithms are based on the gradient of the objective function. In this section, the formulations of objective function, gradient evaluation, optimization algorithm and sequence with an evolutionary parametrization are discussed.

### 3.3.1 Objective function

To evaluate the performance of a particular geometry, aerodynamic coefficients are usually referred to as a quantitative measure. Thus, the objective function, $J$, is commonly formed by two basic coefficients, lift and drag coefficients. For instance, $J = C_D$ or $J = C_D/C_L$. Because the coefficients are obtained by integrating the pressure and stresses around the geometry, the objective function clearly depends on the flow solution. Moreover, the current treatment considers the angle of attack as a design variable, and the angle of attack determines the direction in which the lift and drag components are resolved. Therefore, the objective function also explicitly depends on design variables. Finally, the coordinates of grid points affect the calculation of integrals, so they also appear in the objective function. In general, an objective function can be represented as $J = J(Q, X, G)$, where $X$ and $G$ are design variables and grid point coordinates.

### 3.3.2 Gradient evaluation

To derive the proper form of the gradient, two approaches are considered. The first approach is developed by Nemec and Zingg [40], it adopts the well-known discrete adjoint method [27], and treats the sensitivity of the grid points implicitly. The final form of the gradient with respect to the design variables is given by

$$
\mathcal{G} = \frac{dJ}{dX} = \frac{\partial J}{\partial X} + \psi^T \frac{\partial R}{\partial X} 
$$

(3.6)

$$
\begin{bmatrix}
\frac{\partial R}{\partial Q}^T \\
\partial Q
\end{bmatrix} \psi = \begin{bmatrix}
\frac{\partial J}{\partial Q}^T \\
\frac{\partial J}{\partial Q}
\end{bmatrix} 
$$

(3.7)

where, $\frac{\partial J}{\partial X}$ and $\frac{\partial J}{\partial Q}$ are easy to compute either analytically or using finite differences. $\frac{\partial R}{\partial X}$ implicitly contains the sensitivity of grid points and is also evaluated by finite differences. However, as pointed by Truong et al. [51], if a sophisticated grid perturbation algorithm is employed, computing this partial derivative is expensive. Therefore, this approach is only used with the algebraic grid movement. Finally, $\frac{\partial R}{\partial Q}$ is the flow Jacobian, and Eq. 3.7 is solved using the GMRES linear solver [40].
The second approach is developed by Truong et al. [51] and Hicken and Zingg [18]. It extends the previous discrete adjoint approach and explicitly expresses the grid sensitivity through mesh adjoint equations. Since in three-dimensional optimization problems, the computational grid is represented by B-spline volumes, the dependence of the objective function can be written as $J = J(Q, X, b^{(m)})$, and the dependence of the flow residual equations can be explicitly expressed as $R = R(Q, X, b^{(m)})$, where $b^{(m)}$ are the volume control points and $X$ are the design variables. The derivation of the gradient is done through Lagrange multipliers, which requires an optimal point to satisfy the flow residual equations and grid movement equations:

$$\min \ J(Q, X, b^{(m)})$$

w.r.t $Q, X, b^{(m)}$

s.t. $R(Q, X, b^{(m)}) = 0$

$$r^{(i)}(b^{(i)}, b^{(i-1)}) = 0, \quad i \in \{1, \ldots, m\}$$

The Lagrangian is defined by

$$L = J + \sum_{i=1}^{m} \lambda^{(i)} r^{(i)} + \psi^T R$$ (3.8)

where, $\lambda^{(i)}$ and $\psi$ are the mesh and flow adjoint variables. The optimality condition sets the partial derivatives of the Lagrangian to zero:

$$\frac{\partial L}{\partial \lambda^{(i)}} = 0 = r^{(i)}, \quad i \in \{1, \ldots, m\}$$ (3.9)

$$\frac{\partial L}{\partial \psi} = 0 = R$$ (3.10)

$$\frac{\partial L}{\partial Q} = 0 = \frac{\partial J}{\partial Q} + \psi^T \frac{\partial R}{\partial Q}$$ (3.11)

$$\frac{\partial L}{\partial b^{(m)}} = 0 = \frac{\partial J}{\partial b^{(m)}} + \lambda^{(m)} r^{(m)} + \psi^T \frac{\partial R}{\partial b^{(m)}}$$ (3.12)

$$\frac{\partial L}{\partial b^{(i)}} = 0 = \lambda^{(i)} r^{(i)} + \lambda^{(i+1)} r^{(i+1)} + \psi^T \frac{\partial R}{\partial b^{(i)}} \quad i \in \{m-1, \ldots, 1\}$$ (3.13)

$$\frac{\partial L}{\partial X} = 0 = \frac{\partial J}{\partial X} + \sum_{i=1}^{m} \left( \lambda^{(i)} r^{(i)} \frac{\partial b^{(i)}}{\partial X} \right) + \psi^T \frac{\partial R}{\partial X}$$ (3.14)

Following the strategy proposed by Truong et al. [51], Eqs. 3.9 to 3.13 are solved sequentially, providing the flow and mesh adjoint variables to form the right-hand side of the last equation (details are presented by Hicken and Zingg [22]), which produces the
gradient:
\[ G = \frac{\partial J}{\partial X} + \sum_{i=1}^{m} \left( \lambda^{(i)T} \frac{\partial r^{(i)}}{\partial b^{(i)}} \frac{\partial b^{(i)}}{\partial X} \right) + \psi^T \frac{\partial R}{\partial X} \] (3.15)

### 3.3.3 Optimization algorithm

Once an objective function and its gradient are properly defined, an optimization problem of the following form can be posed:

\[
\begin{align*}
\min & \quad J(X) \\
\text{w.r.t} & \quad X \\
\text{subject to} & \quad C_i(X) \leq 0, \quad i = 1, \ldots, N_c
\end{align*}
\]

where \( C_i(X), i = 1, \ldots, N_c \) represent the imposed constraints. The presence of constraints has a significant impact on the optimization problem. Here, a summary of commonly used constraints in aerodynamic shape optimization is given.

For two-dimensional airfoil optimization:

- **Lift constraint**: the airfoil should maintain a specified lift coefficient during an optimization

- **Thickness constraint**: the airfoil thickness should exceed specified minimum thicknesses at some stations

- **Range thickness constraint**: the maximum thickness of an airfoil should exceed a specified value among a group of chordwise stations

- **Area constraint**: the area enclosed by the airfoil should exceed a specified minimum value

For three-dimensional wing optimization:

- **Lift constraint**: the wing should preserve a specified lift coefficient during an optimization

- **Area constraint**: the wing should maintain a specified planform or wetted area

- **Volume constraint**: the volume enclosed by the wing should exceed a specified minimum value
- Linear constraints: the linear relations that couple the surface control points should be satisfied

Depending on the treatment of the constraints, two optimization algorithms are implemented. One of them is the BFGS quasi-Newton algorithm combined with a backtracking line search method. This optimizer is designed for unconstrained optimization problems; therefore, the imposed constraints have to be handled by penalty terms and incorporated into the objective function. Currently, the following quadratic penalty method is used:

\[
P.T. = \begin{cases} \frac{1}{2} (1 - \frac{C_i}{C_i^*})^2 & \text{if } C_i < C_i^* \\ 0 & \text{otherwise} \end{cases}
\]

where \( C_i^* \) is the specified target value. The above form works for inequality constraints; when an equality constraint is encountered such as a lift constraint, only the first equation is needed. Additionally, Zingg and Billing [53] develop a strategy that adjusts the angle of attack in the flow solver to satisfy the lift constraint, and this is used as an alternative treatment for the lift constraint. The major components of this BFGS algorithm, i.e. the Hessian approximation and backtracking line search strategies are documented in many references (e.g. Fletcher [14]), but two additional implementation regulations are imposed. First, the line search is limited to a maximum of 20 steps to avoid unnecessary iterations when the objective function values become indistinguishable due to small changes in the step size. When this occurs, the optimization is restarted using a steepest-decent search direction. Second, after each iteration, large violations of constraints are checked. If such violations are detected, the step size is reduced by a factor of two.

The other optimization algorithm is the computational package, SNOPT, developed by Gill et al. [15]. This algorithm handles constraints by forming a modified Lagrangian and solves for the optimal point which satisfies the KKT optimality condition. SNOPT adopts a sequential quadratic programming (SQP) method. The search directions are obtained from quadratic subproblems that approximate the Lagrangian subject to linearized constraints (include initially imposed linear constraints and locally linearized constraints). Since these quadratic subproblems are formed based on the approximated Hessian of the Lagrangian, the full memory BFGS Hessian matrix approximation is currently used. Each subproblem is solved using an active-set method [16], while during this process the satisfaction of the linearized constraints is required. Thus SNOPT always keeps the initially imposed linear constraints satisfied, and this property is used to estab-
lisch the necessary coupling among control points. Unless otherwise stated, the SNOPT package is used as the default optimizer.

### 3.3.4 Optimization sequence

Having all the numerical routines available, an automated optimization process can be established with the aid of an evolutionary B-spline parametrization. Since this evolutionary parametrization is able to produce consistent geometry representations as the number of control points gradually increases, the proposed process is executed in a progressive manner (Figure 3.1): initially, an optimization is started with relatively few control points; once it converges or is close to convergence, the geometry parametrization is refined and the next optimization begins with more control points based on the obtained geometry. This procedure repeats continuously until the final termination. To clearly present the results, each time, finding an optimal geometry is regarded to be a completion of an optimization cycle.

Although this optimization process is carried out following a straightforward sequence,
there are still some questions to be addressed during a practical implementation. First, the knot insertion algorithm can be performed at different knot intervals, so the additional control points can be placed at various locations. To choose the effective control points among all the candidates, we have established the following selection criteria:

1. The number of parameters in each knot interval should remain above a certain threshold. Since the B-spline control points are locally supported, clustering excessive control points at some region would not lead to significant improvement.

2. Local non-linear constraints such as minimum thickness, critically impact the movement of control points. If a constraint is inactive, it implies that there exists a feasible region in the design space, and adding control points at this location has a large probability to outperform adding control points in a region with active local constraints. Therefore preference is given to refinements which add control points to the regions containing inactive local non-linear constraints.

3. The satisfaction of the linear constraints is mandatory. Any parametrization refinement violating the linear constraints is dropped from consideration.

4. If there are multiple candidates remaining after considering the above requirements, the magnitude of the gradient with respect to the proposed new control points is used as a measure of the potential improvement. The prospective control points with the highest sensitivity are selected.

Second, parametrization refinement (i.e. increasing the number of design variables) occurs at regular intervals during the optimization process, but it is hard to identify a clear signal that triggers this procedure. Some authors [33, 4] argue that when the number of design variables is small, relatively few optimization iterations are enough and full convergence is unnecessary since these optimization cycles are only intermediate steps. However others [48, 11] point out that the optimizations at the first few cycles should be driven sufficiently close to the optimal shape, so that the current shape can provide a good start for subsequent optimizations. At the current stage, we choose to perform well converged optimization at each cycle for the following two reasons:

1. It is desirable to have a small number of critical design variables. If a significant improvement is made by existing design variables, the optimization could be terminated without attempting the next cycle. Thus, optimization at each cycle is driven toward convergence to make full use of every design variable.
2. One selection criterion uses the magnitude of the sensitivity as a test of different parametrization refinements. This is only valid if the optimization is fairly well converged.

Moreover, terminating the optimization process also requires a criterion. As pointed in the work of Zingg et al. [54], the benefit of introducing control points after a certain threshold is marginal. Thus, this process is terminated if significant improvements are not achieved by adding further design variables. Other termination conditions are also posed; they will be stated in the test cases.
Chapter 4

Airfoil Optimization

The evolutionary parametrization is applied to two-dimensional airfoil shape optimization. In this chapter, I present the implementation and performance of this approach through a number of test cases. To demonstrate its effectiveness, the same problems are also solved using varying numbers of control points that are uniformly placed. Here “uniform” means that for a two-dimensional airfoil, its knot vector obeys the stated cosine function, and for a three-dimensional wing, its knot vectors evenly partitions the parametric domain. The results from these two approaches are compared and discussed.

4.1 Implementation of Knot Insertion

The knot insertion algorithm is discussed thoroughly in previous chapters. Specifically for airfoil parametrization, some complementary discussions and elaborations are necessary. The first consideration is attributed to the frozen control points at the leading and trailing edges. Figure 4.1 depicts a cubic B-spline parametrization of the NACA0012 airfoil using 15 control points, and its corresponding knot sequence is schematically displayed in Figure 4.2. In order to prevent translation and an unrealistic leading edge radius, three control points at the leading edge and two points at the trailing edge are frozen. In Figure 4.1 the frozen control points are labelled by the unshaded squares and the flexible control points by the shaded squares. As demonstrated in Chapter 2, if a new term \( t^* \) is added between \( t_r \) and \( t_{r+1} \), the new control points, \( \{d_i : i = r - 2, \ldots, r\} \), would be relocated. To ensure the appearance of new control points does not affect the frozen control points, \( \{d_i : i = 1, 7, 8, 9, 15\} \), the additional knot can only be inserted at the dashed knot intervals. In other words, the intervals adjacent to the multiple knot in the
middle are not permitted for knot insertions.

Apart from the restriction caused by the fixed control points, the thickness constraints imposed at certain chordwise locations play a significant role in the knot insertion algorithm. Take the minimum thickness constraint displayed in Figure 4.1 as an example. This particular constraint defines the minimum distance between two surface grid points, $s_1$ and $s_2$. Projecting these two points on the parametric domain using Eq. 2.13, one can identify the knot intervals in which they reside (Figure 4.2). If the thickness constraint is inactive, it implies that there exists a feasible region in the neighbourhood of this constraint, and additional degrees of freedom may result in a better optimal solution. Thus
Table 4.1: Thickness constraints for Case 1

<table>
<thead>
<tr>
<th>Location (c)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thickness (c)</td>
<td>0.04</td>
<td>0.11</td>
<td>0.04</td>
<td>0.026</td>
<td>0.012</td>
<td>0.002</td>
</tr>
</tbody>
</table>

in the following test cases, if an inactive thickness constraint is encountered, inserting an additional knot at the interval where this constraint resides and its adjacent intervals is preferred. Here, the inclusion of the neighbouring intervals is effective since a $k$-th order B-spline control points supports $k$ knot intervals.

Finally, as mentioned in the selection criteria, to avoid accumulation of control points, the size of the knot intervals should be restricted. In the two-dimensional airfoil optimization problems, the minimum size of a knot interval is set to be 5% of the parametric domain. This number has been tested in numerous experiments, and the results indicate that it works well for airfoil optimization problems.

### 4.2 Test Cases

In the following test cases, the RAE2822 and NACA0012 airfoils are used as the baseline geometries for subsonic and transonic problems respectively. The computational grids have a “C” topology with 289 nodes in the streamwise direction and 65 nodes in the normal direction. The normal off-wall spacing is $2 \times 10^{-6}$ chord, the surface spacing at the leading edge and trailing edge are $5 \times 10^{-4}$ chord and $1 \times 10^{-3}$ chord, respectively, and the distance to the far-field boundary is 24 chords.

#### 4.2.1 Case 1: Lift-constrained drag minimization (subsonic)

The design objective is to minimize the drag coefficient ($J = C_d$) at a Mach number of 0.25 and a Reynolds number of 2.88 million in a fully turbulent flow. The lift coefficient is constrained at 0.33, and the thickness constraints listed in Table 4.1 define the minimum thickness at specified chordwise locations. The initial airfoil is parametrized using 15 uniformly distributed control points. Besides the frozen points, the ordinates of the remaining points are used as design variables. The angle of attack is initially 3.0 degrees and it is not a design variable but adjustable to satisfy the lift constraint [53].
The optimization is terminated after 7 optimization cycles (i.e. 6 control points are added), because the subsequent parametrization cycles fail to provide an improvement in the objective function. Figure 4.3(a) displays the surface modification and the inserted control points upon the completion of the optimization. Figure 4.3(b) shows the comparison of the pressure distribution.

To compare the performance of the optimal solution using evolutionary parametrization with the results obtained from uniformly refined parametrizations, Figure 4.4 plots the scaled objective function values versus the number of control points. The scaling factor is the drag coefficient of the NACA0012 airfoil. From this figure, a few typical features of the proposed optimization sequence can be identified. First of all, it is efficient at finding critical control points. In this problem, the optimization sequence achieves a reduction of 5.96% in drag coefficient using 21 control points, but the same amount of improvement requires 33 control points if they are uniformly distributed. Secondly, the sensitivity analysis only evaluates the gradient of the objective function with respect to the design variables, but the optimality measurement incorporates both the objective function and the constraints, thus a relatively large sensitivity does not guarantee a significant optimality value. Hence, the additional degree of freedom may not result in desirable improvements. Evidence can be found from the convergence plots listed in Figure A.1, the initial optimality values of cycle 4 and 6 are small, and correspondingly, the reductions of the drag coefficient are not obvious in these two cycles. Nonetheless, the optimization sequence with evolutionary parametrization gives a systematic way of selecting...
4.2.2 Case 2: Lift-to-drag ratio maximization

This test case adopts the BFGS algorithm, and its objective is to maximize the lift-to-drag ratio. The baseline geometry and operating conditions are identical to Case 1. The thickness constraints are summarized in Table 4.2. Initially, the airfoil is represented by 15 control points, and the angle of attack is considered as a design variable. Since BFGS addresses unconstrained minimization problems, the thickness constraints are included in the objective function as a quadratic penalty term, and the objective function is constructed as follows:

\[ J = \frac{C_d}{C_l} + \omega_T T \]

where \( t_i \) is the thickness at a specified chordwise station, \( t_i^* \) is the target value, and the applied weight, \( \omega_T \), is 1.0.

Figures 4.5(a) and 4.5(b) show the change of the airfoil shape and pressure distribution. Figure 4.6 displays the scaled objective function values from the proposed sequence and the optimizations with uniform refinements. The scaling factor is the lift-to-drag
Table 4.2: Thickness constraints for Case 2

<table>
<thead>
<tr>
<th>Thickness constraints</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location ((c))</td>
<td>0.05</td>
<td>0.35</td>
<td>0.65</td>
<td>0.75</td>
<td>0.85</td>
<td>0.95</td>
<td>0.99</td>
</tr>
<tr>
<td>Thickness ((c))</td>
<td>0.04</td>
<td>0.11</td>
<td>0.04</td>
<td>0.03</td>
<td>0.026</td>
<td>0.012</td>
<td>0.002</td>
</tr>
</tbody>
</table>

Figure 4.5: Optimized geometry and pressure distribution of Case 2

Figure 4.6: Objective function comparison of Case 2

The optimization with evolutionary parametrization terminates at the end of cycle 6, due to the fact that all the thickness constraints are considered. The performance of
the optimization sequence confirms its effectiveness at introducing critical control points. The optimal lift-to-drag ratio is obtained after adding 5 design variables. The gradient histories are included in Figure A.2, and it can be seen that BFGS converges more slowly than SNOPT, so the primary optimizer used in this work is SNOPT.

### 4.2.3 Case 3: Lift-constrained drag minimization (transonic)

In this test case, the drag coefficient is minimized in a transonic turbulent flow. The freestream flow has a Mach number of 0.74 and a Reynolds number of 2.7 million. The lift coefficient is constrained to be 0.733, and the geometric constraints contain area and thickness constraints (Table 4.3). The baseline shape is the RAE2822 airfoil, which is parametrized by 15 control points initially. The treatment of angle of attack is the same as in Case 1.

In this problem, the area constraint affects every design variable; thus it behaves as a global constraint. The thickness constraints only influence the control points close to them, so they are referred to as local constraints. As discussed in Section 4.1, the parametrization refinement takes into account whether or not these two thickness constraints are active. The optimization sequence is terminated when two successive parametrization refinements result in negligible improvement. Figure 4.7(a) displays the surface change and the inserted control points upon the completion of the optimization. Figure 4.7(b) shows a comparison of the pressure distributions. The shock is completely eliminated by the optimized airfoil.

The optimal solution further is compared with a series of optimization results with varying numbers of control points uniformly placed around the airfoil. Figure 4.8 depicts the objective function (drag coefficient) scaled by its initial value. In this figure, a consistent reduction of the objective function is observed from the optimization sequence with evolutionary parametrization. Using 21 control points, the proposed optimization sequence is able to reach an optimal solution that is comparable to the result obtained.

<table>
<thead>
<tr>
<th>Constraints</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location ((c))</td>
<td>0.96</td>
<td>0.99</td>
<td>-</td>
</tr>
<tr>
<td>Thickness ((c))</td>
<td>0.006</td>
<td>0.0012</td>
<td>-</td>
</tr>
<tr>
<td>Area ((c^2))</td>
<td>-</td>
<td>-</td>
<td>0.07790</td>
</tr>
</tbody>
</table>
Figure 4.7: Optimized geometry and pressure distribution of Case 3

Figure 4.8: Objective function comparison of Case 3

using 27 design variables evenly around the airfoil.

4.2.4 Case 4: Multipoint optimization

The last test case taken from the paper by Nemec and Zingg [41] applies evolutionary parametrization to a multipoint optimization problem. This test case begins with the RAE2822 airfoil and minimizes the drag coefficient over a range of Mach numbers: 0.68, 0.71, 0.74, 0.76 with associated weights of 1.0, 1.0, 2.0, 3.0. The lift coefficient is constrained at 0.733, and the freestream flow has a Reynolds number of $9 \times 10^6$. The angle
of attack for each operating point is adjusted to meet the lift requirement. The geometric constraints are identical to Case 3.

The optimization sequence starts with 15 control points around the airfoil and terminates after 4 additional control points are introduced. Because of the presence of multiple operating points, the sensitivity analysis is modified to use a composite gradient magnitude to measure the potential of the prospective control points. The optimized geometry is displayed in Figure 4.9(a), and the pressure distributions for each operating point are depicted in Figure 4.10. To illustrate the capability of the optimization sequence, the optimal solution is compared with the results using various uniform parametrizations. From Figure 4.9(b), one can be seen that the sequence with evolutionary parametrization effectively introduces critical design variables, and it outperforms the conventional uniform parametrizations by using fewer design variables.
Chapter 4. Airfoil Optimization

Figure 4.10: Pressure distribution for each operating point
Chapter 5

Induced Drag Minimization

The evolutionary parametrization using B-spline volumes is employed in the studies of induced drag minimization for finite span wings. The purpose of this chapter to demonstrate that the proposed optimization sequence effectively finds the optimal aerodynamic shape through a few test cases.

5.1 Test Case Setup

When investigating induced drag reduction, the classical Prandtl lifting line theory is considered as a benchmark for optimization results. According to the lifting line theory [2], the minimum induced drag for a planar wake occurs when the lift is elliptically distributed, and the minimum drag has the following expression:

$$C_{D,\text{min}} = \frac{C_L^2}{\pi \Lambda}$$

where $C_D$ and $C_L$ are drag and lift coefficient, respectively, $\Lambda = \frac{b^2}{S}$ is the aspect ratio, $b$ is the span, and $S$ is the reference area which is used to compute the coefficients and the aspect ratio. In the following test cases, $S$ is chosen to be the planform area.

Two types of computational grids are used: the first one is a 12-block flat-plate grid which has an “H-H” topology; the second grid is a 6-block box wing grid. Table 5.1 summaries the details of each grid. All the spacing parameters are stated in terms of the root chord.
5.2 Test case

5.2.1 Case 1: Planform optimization

A typical method to achieve minimum induced drag is to vary the planform shape, i.e. change chord lengths at a few spanwise stations. In this test case, the flat-plate grid is approximated using B-spline volumes, and the initial geometry is constructed to be a rectangular wing with a uniform chord of 2/3, a semi-span of 2, and NACA0012 sections. The chord and span are non-dimensionlized by the chord length of the flat-plate grid. The planform area, $S = 4/3$, is used as the reference area, and remains fixed during the optimization. The freestream Mach number is 0.5, and the lift coefficient is constrained at 0.35. The initial parametrization for each block is $7 \times 7 \times 6$. In other words, on the wing surfaces, there are 7 control points in the streamwise and spanwise directions. The angle of attack is considered a design variable, and it is initially 3.367799 degrees, which produces the target lift coefficient with the baseline geometry.

To perform an optimization through planform variation, all the control points except the ones on the trailing edge are free to move in the streamwise direction, and the entire trailing edge is fixed to reduce the impact of a nonplanar wake. The leading edge control points possess complete degrees of freedom; other interior points are coupled with the leading edge control points to provide a scaling once the leading edge changes. Therefore, the initial effective design variables are the 7 chord lengths and the angle of attack. One additional box constraint is imposed to confine the wing within $-0.5 < x < 0.5$. The parametrization refinement is formulated such that more spanwise stations are added. Hence, the number of effective design variables gradually increases. By experience, each knot interval is required to contain at least 15% of the total number of parameters in the spanwise direction, so that the B-spline control points lie sufficiently far apart.

Figure 5.1 shows the planform deformation as more spanwise stations are progressively added to the parametrization. After cycle 5, all the knot intervals reach the size limitation, and the process is terminated. The lift distributions are plotted in Figure 5.2(a),
and it can be seen that the final lift distribution is close to the classical elliptical distribution for the most part but substantially different at the tip. This phenomenon has been identified by Hicken and Zingg [20], who point out that the presence of the sharp wing tip causes the vortex to release along the tip edge, which ultimately results in a nonpla-
nar wake. Nevertheless, the purpose of this example is to demonstrate the effectiveness of the evolutionary parametrization. Figure 5.2(b) compares the proposed optimization sequence with the optimal solutions from uniformly spaced parametrizations (the optimization with 10 spanwise control points does not converge, thus it is not plotted). In terms of the drag coefficient, the former achieves a total reduction of 2.4%, while the latter do not receive significant benefits from adding design variables, and their performance is inferior comparing to the evolutionary counterpart with the same number of spanwise control points.

5.2.2 Case 2: Planform and twist optimization

Besides through the planform deformation, an elliptical lift distribution can be obtained by manipulating the twist, sectional lift and some combinations. The second test case investigates an optimal shape allowing variations of the planform and twist. The initial geometry is the same unswept rectangular wing used in the previous problem. The reference area, lift constraint, and the operating Mach number are also identical. The upper and lower surfaces of the wing are parametrized using 7 control points in the streamwise direction and 5 control points in the spanwise direction. The angle of attack is used as a design variable, its initial value is 3.94321 degrees, which makes the baseline geometry satisfy the lift constraint.

Different linear constraints are imposed to generate effective design variables. At each spanwise station, control points are limited to rotations and scalings defined by the leading and trailing edges:

$$\begin{align*}
(x - x_{LE})_k &= r_k \begin{bmatrix} \cos(\beta_k) & -\sin(\beta_k) \\
\sin(\beta_k) & \cos(\beta_k) \end{bmatrix} \begin{bmatrix} x_{TE} - x_{LE} \\
z_{TE} - z_{LE} \end{bmatrix}_k
\end{align*}$$

Here $k$ is the index of a spanwise station, $r_k = |x - x_{LE}|/|x_{TE} - x_{LE}|$, and $\beta_k = \arctan((z - z_{TE})/(x - x_{TE}))$. In order to reduce the effect of a nonplanar wake, the trailing edge is fixed. Also, to prevent generating a winglet, the vertical coordinate of the leading edge at the wing tip is constrained to be the same as its neighbouring points, and the bending angle of two adjacent spanwise stations is restricted within 20 degrees. As a result, the effective design variables are the chord lengths of each spanwise station, the twist of the inboard stations, and the angle of attack. The parametrization refinement occurs along the spanwise direction, and the minimum knot interval is set to contain 20% of the total parameters in this direction.
Figure 5.3: Shape changes of the planform and twist optimization
Figure 5.3 shows the planform and twist deformation as well as the control point distribution at the end of each optimization cycle. Figure 5.4(a) depicts the lift distributions, and it can be seen from this plot that the final lift distribution is close to the theoretical elliptical but a relatively large deviation again occurs near the wing tip. The argument of the edge separation still applies in this case, and the presence of the twist possibly enhances the effect of a nonplanar wake. To illustrate the effectiveness of the evolutionary parametrization, its optimal solution is compared with the results obtained using uniform parametrizations (Figure 5.4(b)). The proposed sequence consistently reduces the drag coefficient, producing a total of 6.0% reduction in the end. However, the optimizations with uniform parametrizations are unable to improve upon the result obtained the initial parametrization, which leads to only about 50% of the improvement achieved using the evolutionary parametrization.

5.2.3 Case 3: Winglet optimization

With regard to induced drag reduction, the effect of a non-planar structure has been well recognized [29]. In this problem, the optimization sequence based on evolutionary parametrization is adopted to investigate an optimal spanwise vertical structure that yields minimal induced drag. The baseline geometry is the same rectangular wing used for the previous two test cases. The freestream Mach number is 0.5, the planform area, $S = 4/3$, is considered as the reference area, and the lift coefficient is constrained at the
value of 0.35. The angle of attack is chosen such that the lift constraint is satisfied at the beginning.

The initial B-spline volume approximation uses $7 \times 5 \times 6$ control points for each block, so on the wing surfaces, there are 7 points in the streamwise direction and 5 in the spanwise direction. The vertical coordinates of the surface control points are free to move, and a box constraint is imposed to confine the entire geometry within $-0.2 \leq z \leq 0.2$. To prevent excessive degrees of freedom, the $z$ coordinates of interior control points at each spanwise station are related to the corresponding leading and trailing edge points so that their vertical positions relative to the chord line is maintained. Also, to generate a reasonable vertical structure, the control points near the wing tip are maintained in a consistent manner. The formed winglet can be either upward or downward with wavy surface details, but an abrupt change of angle is avoided. This is done by defining the maximum dihedral and anhedral between adjacent control points to be 145 and 35 degrees respectively. These relations are expressed in terms of linear constraints which effectively restrict the total number of degrees of freedom; complete freedom is only given to the vertical coordinates of the control points located at the leading and trailing edges. Consequently, the parametrization refinement occurs along the spanwise direction, adding more effective design variables. The same restriction on knot interval size still applies; moreover, the added stations are required to satisfy the existing linear constraints.

Figure 5.5 shows an upward winglet produced during the optimization. The added control points provide additional degrees of freedom to make the winglet more or less normal to the horizontal wing. The last two refinements fail to provide sufficient improve-
5.4 Case 4: Box-wing optimization

Another popular nonplanar geometry for induced drag reduction is a box-wing. In this test case, a box-wing configuration is optimized using the optimization sequence with evolutionary parametrization. The baseline box-wing geometry has a semi-span of 3.0, a chord length of 1.0, and NACA0012 sections. The initial height to span ratio is 0.105. The previously mentioned 6-block grid is adopted, and it is initially approximated using B-spline volumes with 9 control points in the streamwise direction and 5 in the spanwise and vertical directions. The planform area considering the contribution from two horizontal wings is 5.87; it is used as the reference area and constrained to this value during optimization. The imposed lift constraint requires a lift coefficient of 0.25. The angle of attack is fixed to reduce the chance of non-unique optimal designs, and its value is chosen to meet the lift constraint with the baseline geometry.

The control points along the horizontal wings are allowed to move in the vertical
direction, and the control points on the vertical plate are free to move in the spanwise direction. The $y$ coordinates of control points at the horizontal wings are linearly interpolated using the locations of the root and junction. Similarly, the $z$ coordinates of control points at the vertical plate are scaled by the upper and lower junctions. For control points in the same section, linear relations are established to couple them with the leading and trailing edge. Thus only the leading and trailing edge control points have complete freedom. In addition, to maintain the integrity of the geometry, dihedral and anhedral angle limitations are defined between successive control points, so no abrupt bending is allowed. At the junctions of the horizontal wings and the vertical plate, control points are extrapolated based on adjacent sections including the points at the leading and trailing edges. Besides these couplings, a box constraint is imposed to confine the entire geometry within $0 \leq y \leq 3, -0.315 \leq z \leq 0.315$. The parametrization refinements occur along the span and the vertical plate; additional sections are simultaneously inserted at the upper and lower horizontal wings as well as the vertical plate.

Figure 5.7 shows the shape variation at the end of each cycle. The process ceases after cycle 5 because further parametrization refinements are not able to satisfy the
predefined linear constraints. The aerodynamic performance of different parametrizations is depicted in Figure 5.8. From this plot, one can see that the optimization based on evolutionary parametrization gains substantial benefit from the added design variables, while its counterparts do not exhibit much dependence on the enriched design variables. This fact confirms the effectiveness of critical design variables, and the capability of the proposed optimization sequence is reinforced.

5.2.5 Case 5: Wing optimization using adaptive constraints

The last test case is an exploratory example. Its objective is to use the established optimization sequence to produce a complete wing design which is not limited to designated geometry changes. Thus, a large number of control point coordinates are treated as independent design variables, and no effective variables are specified.

The baseline geometry, reference area, lift constraint and operating conditions are identical to Case 1, and a volume constraint is imposed to maintain the initial value. The basic parametrization places 6 control points in the streamwise direction and 5 in the spanwise direction on both the upper and lower surfaces of the wing. In order to reduce the possibility of multiple optima, the angle of attack is constrained to 3.939685 degrees which generates the target lift coefficient with the baseline configuration.

Even though this optimization seeks significant shape manipulation, not every single
control point coordinate possesses complete freedom. In order to maintain the integrity
of the computational grid, the following equality linear constraints are necessary: 1) The
control point at the root section trailing edge is fixed to eliminate translational motion.
2) The $y$ coordinates of the control points on the symmetry plane (the root section control
points) are frozen. 3) The interior control points of each spanwise station attain their $x$
coordinates by interpolating the corresponding leading and trailing edge points. 4) The
$y$ coordinates of the interior control points at every streamwise station are interpolated
between the corresponding root and tip points.

The remaining control point coordinates serve as the design variables and are ma-
nipulated by the optimizer. However, due to the limited capability of SNOPT, conver-
gence difficulty occurs when a large quantity of unbounded design variables are present.
Therefore, to facilitate the optimizer, each individual control point coordinate is assigned
adjustable bounds using a set of linear inequality constraints, and these bounds are gradu-
ally loosened to minimize their impact on the final optimal solution. The summary of
the inequality constraints are provided below:

1. At the tip section, the trailing edge control point is related to the leading edge
control point.

$$|x_{TE} - x_{LE} - c| \leq \Delta_x \quad |y_{TE} - y_{LE}| \leq \Delta_y \quad |z_{TE} - z_{LE}| \leq \Delta_z\,$$

where $c$ is the initial chord length. $\Delta_x = \frac{1}{4} c$, $\Delta_y = 0.1$, and $\Delta_z = 0.1$ are the initial
offsets.

2. At the root section, The leading edge control point is associated with the trailing
edge control point.

$$|x_{TE} - x_{LE} - c| \leq \Delta_x \quad |z_{TE} - z_{LE}| \leq \Delta_z\,$$

3. The degree of freedom of interior leading/trailing edge control points are defined
based on their corresponding root and tip points.

$$|x - (1 - r_k)x_{root} - r_kx_{tip}| \leq \Delta_x \quad |z - (1 - r_k)z_{root} - r_kz_{tip}| \leq \Delta_z\,$$

where $k$ is the index of the interior control points, and $r_k = |y - y_{root}|/|y_{tip} - y_{root}|$.

4. The interior control points on the wing tip edge are coupled with its leading and
trailing edge points.

$$|y - (1 - \lambda_j)x_{LE} - \lambda_jx_{TE}| \leq \Delta_y \quad |z - (1 - \lambda_j)z_{LE} - \lambda_jz_{TE}| \leq \Delta_z\,$$
Table 5.2: Surface parametrization and drag coefficient of each optimization cycle

<table>
<thead>
<tr>
<th>Cycle</th>
<th>Number of streamwise CPs</th>
<th>Number of spanwise CPs</th>
<th>Drag coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial</td>
<td>6</td>
<td>5</td>
<td>0.007059</td>
</tr>
<tr>
<td>1</td>
<td>6</td>
<td>5</td>
<td>0.004115</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>6</td>
<td>0.003936</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>6</td>
<td>0.003823</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>7</td>
<td>0.003555</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>7</td>
<td>0.003165</td>
</tr>
</tbody>
</table>

where \( j \) is the index of the interior control points, and \( \lambda_j = |x - x_{LE}|/|x_{TE} - x_{LE}| \).

5. The interior control points of inboard sections are free to move in the vertical direction.

\[
|z - (1 - \lambda_k)z_{LE} - \lambda_k z_{TE} - h| \leq \Delta z_2
\]

where \( k \) is the index of the inboard sections, \( h \) is the initial vertical distance relative to the chord line, and \( \lambda_k = |x - x_{LE}|/|x_{TE} - x_{LE}| \). The offset \( \Delta z_2 \) is set to be 0.02.

6. To avoid infinite expansion, a global box constraint is imposed to confine the entire configuration within \(|x| \leq 2/3, |y| \leq 2.2 \) and \(|z| \leq 0.22\).

The loosening of the control point bounds are attained by increasing the linear constraint offsets. At the end of each optimization cycle, all the inequality constraints are examined. If one constraint becomes active, its offset is increased by half of the initial value.

The evolutionary parametrization refinements also take place as the optimization proceeds. Additional control points are inserted in both streamwise and spanwise directions. However, adding points close to the trailing edge is excluded because excessive manipulation of the trailing edge would produce unsteady flows. As a result, this proposed optimization process not only gradually increases the number of design variables but also progressively loosens the limitations on design variables.

The shape deformations at the end of each optimization cycle are shown in Figure 5.9, and the performance of the geometries is summarized in Table 5.2. The large drag reduction occurring in cycle 1 is mainly due to the increase of the span and the generation of the nonplanar geometry. Upon the completion of cycle 1, the drag coefficient decreases by 41.7%. The following cycles introduce two control points in the spanwise direction.
and two points in the streamwise directions. The appearance of these additional design variables reinforces the vertical structure so that the nonplanar effect becomes more prominent. Moreover, sectional variations are also distinct, and a complex twisted shape emerges at the region close to the wing tip. Besides the effect of new control points, the planform variations are primarily due to loosing the boundaries of the design variables. As can be seen from the planform shape of cycles 2 to 5, the chord length of the wing tip progressively increase from 0.5777 to 1.0461, while the root chord decreases from 0.5 to 0.3205. Finally, in terms of the drag coefficient, from cycle 2 to cycle 5, its value is further reduced by 23.1%, which reveals some potential of using this process to investigate unconventional configurations.
Figure 5.9: Geometry deformation as the parametrization is gradually refined and the constraints are progressively released.
Chapter 6

Conclusions

A B-spline geometry parametrization technique is integrated with a knot insertion method to provide flexible representations for aerodynamic geometries. In Chapter 2, two-dimensional airfoils and three-dimensional configurations have been parametrized by planar B-spline curves and spatial B-spline volumes with a finite number of control points. Adopting the knot insertion method provides the capability of changing the number and the distribution of control points without modifying the shape of the existing configuration. Thus, the B-spline parametrization associated with multiple knot insertions defines an evolutionary parametrization so that an aerodynamic configuration can be represented with a progressively increasing number of control points.

For aerodynamic shape optimization, design variables are normally defined by geometry parametrizations. With B-spline formulations, the coordinates of control points are regarded as design variables. Therefore, an evolutionary parametrization allows flexible design variables for a shape optimization problem. Taking advantage of this feature, a new optimization strategy is proposed. It organizes a design problem as a sequence of optimizations with the number of design variables gradually increasing. The added design variables are systematically selected through sensitivity analysis and other criteria such that they significantly affect the subsequent optimal solutions, in order to achieve an optimal solution with a minimum number of design variables. At the end of each optimization, a series of termination criteria are employed to examine performance of the optimization sequence, and the enrichment of the parametrization occurs only if it is likely to lead to improved performance.

The proposed optimization sequence is applied to airfoil optimization and induced drag minimization test cases in Chapters 4 and 5. The optimization results consistently
demonstrate that the proposed strategy effectively captures critical design variables, and the optimal solutions are gradually improved as the set of design variables enlarges. Comparison examples with uniform parametrizations are also provided for most cases. In airfoil optimization problems, uniformly increasing the number of control points can normally improve the design objective, but the benefit is not as significant as the evolutionary parametrization refinements. For induced drag reduction cases, the optimizations using evenly distributed control points do not exhibit a strong dependence on the number of control points, while the results from evolutionary parametrization exhibit a substantial improvement in the design objective.

6.1 Future Work

6.1.1 Generalization

The generality of the current process is not very ideal since the setup of parametrization refinements and the linear couplings of control points are based on empirical data. Also, the sensitivity analyses are only carried out among a finite number of candidates. Thus, a more general treatment is necessary to cover a wider range of possible refinements.

6.1.2 Evolutionary parametrization with adaptation

Coupling control points through linear constraints is effective at reducing degrees of freedom, but it sometimes prevents adding essential control points or restricts the performance of the additional control points. Given the capability of the current optimizer, inequality relations with adaptive boundaries may provide more freedom to the optimizer and enhance the performance of the added design variables.

6.1.3 Multiple insertions

In the presented test cases, except the box-wing optimization, the optimization sequence introduces one point or one section for each refinement. However, for complex configurations, adding multiple control points is a necessary procedure. The current sensitivity analysis is constructed to test every possible insertion, and this becomes infeasible as multiple control points are inserted at a time, the number of combinations increases
abruptly. Therefore, further investigation is required to establish the selection criteria for insertion of multiple control points.

6.1.4 Alternative parametrizations

There exist a large variety of geometry representation techniques for aerodynamic configurations. Many of them possess the flexibility to produce multiple parametrizations. Thus, the proposed optimization sequence can be naturally extended to these methods, and the unique properties associated with each parametrization method would result in different implementations and applicability for each individual case.

6.1.5 Multilevel optimization

The existing process defines the parametrization refinements as an optimization proceeds. One can also predetermine several different parametrization levels, and adopt multigrid strategies (e.g. “V” cycles) to perform an optimization among different parametrization levels.
References


[43] Joyce E. Penner. *Aviation and the global atmosphere: A special report of IPCC Working Groups I and III in collaboration with the Scientific Assessment Panel*


Appendix A

Convergence Histories

For the BFGS optimizer, the convergence histories are depicted in terms of the magnitude of the gradient. For SNOPT, the convergence measurement is closely related to the KKT condition. The gradient of the Lagrangian is required to be sufficiently small at convergence, thus an optimality value defined by the magnitude of this gradient and the Lagrange multipliers is used to represent the convergence history.
A.1 Airfoil Optimization

Figure A.1: Convergence histories for airfoil optimization case 1
Figure A.2: Convergence histories for airfoil optimization case 2
Figure A.3: Convergence histories for airfoil optimization case 3
Appendix A. Convergence Histories

Figure A.4: Convergence histories for airfoil optimization case 4

A.2 Induced Drag Minimization

Figure A.5: Convergence histories for planform optimization
Figure A.6: Convergence histories for planform and twist optimization

Figure A.7: Convergence histories for winglet optimization
Figure A.8: Convergence histories for box-wing optimization

Figure A.9: Convergence histories for wing optimization using adaptive constraints