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

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.

List of Figures

Chapter 1

Introduction

1.1 Motivation

~~The~~ A air traffic has grown at 5% - 9% annually since 1960s, and it is expected to maintain such ~~the~~ growth rates for the next few decades. The ~~fast~~ expansion of the air transport causes ~~rapid~~ increases in jet fuel consumption and aviation emissions. According to a special report submitted to the Intergovernmental Panel on Climate Change (IPCC) [44], 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~~s~~ two important consequences:

1. Increasing Jet fuel consumption speeds up the depletion of oil, and the scarcity of oil ~~would~~ ultimately increases its price. Since ~~the~~ 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 ~~would~~ 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 [36], 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 ~~the~~ unconventional configurations ~~becomes~~ has become ~~the~~ a major trend in aerodynamic shape design.

relatively slow ly [24]. The other requires additional gradient evaluation of the objective function with respect to the design variables. This type of gradient-based optimizer~~s~~ ~~are~~ is capable of finding an optimum with relatively low computational cost. However, it inherently converges to a local optimum and is limited to ~~the~~ problems with a smooth design space.

Due to the ~~presence of the imperfections in~~ ~~an~~ limitations of aerodynamic shape optimization methods, advanced optimization strategies are needed to improve the implementation efficiency and 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 stiff problem into several easier sub-problems and solves them sequentially. ? One example can be referred to Alexandrov et al. [1]. They propose an approximate and model management framework that integrates a~~n~~ 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~~s~~. As indicated by a number of authors [4, 3, 35], the presence of an excessive number of design variables c~~ould~~an 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, ~~the~~ 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 ~~are~~ is provided by a geometry parametrization, an ideal parametrization method is required to explicitly specified, which leaves little flexibility for a designer to adjust the design space according to specific requirements and conditions.

The ~~D~~domain element approach [39] 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 ~~are~~ 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 recent 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; experimentations and turnings 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 ~~as~~ 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 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 [35]. 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 [33, 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. [39] formulate~~s~~ 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 efficiency and optimal solution 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 [41] and B-spline volume

Chapter 2

B-spline Parametrization

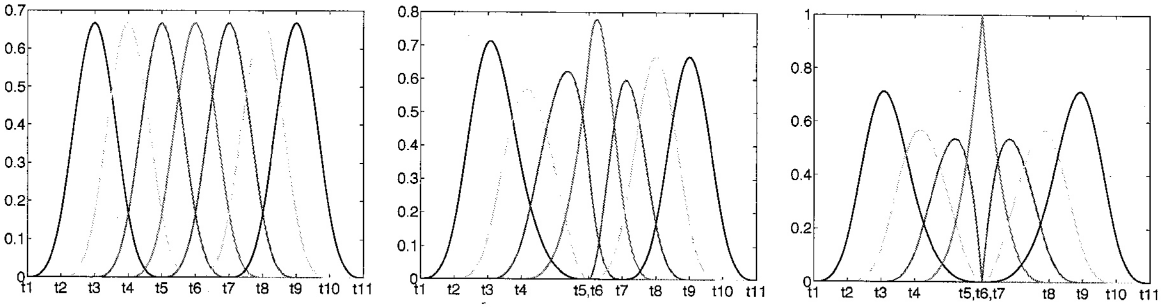
When a ~~the~~ B-spline parametrization is employed in a shape optimization, the Coordinates ~~positions~~ of its control points are normally used as design variables. Therefore, constructing an evolutionary parametrization requires a geometry to be represented by different number(s) of control points. 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, $\{f : \xi \to \mathbf{X}\}$. 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. A partition of the parametric domain is defined by dividing the domain into several intervals, and the vector containing this partition is referred to as the knot vector. ? If $\xi \in [0,1]$, $t = \{t_1 \leq t_2 \leq \ldots \leq t_{m-1} \leq t_m\}$ is 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 multidimensional parametric space, there exist~~s~~ different knot vectors for each dimension.

?

how does this depend on $\xi \in [0,1]$?



(a) simple knots

(b) double knots

(c) triple knots

Figure 2.2: B-spline basis function with different knot vectors

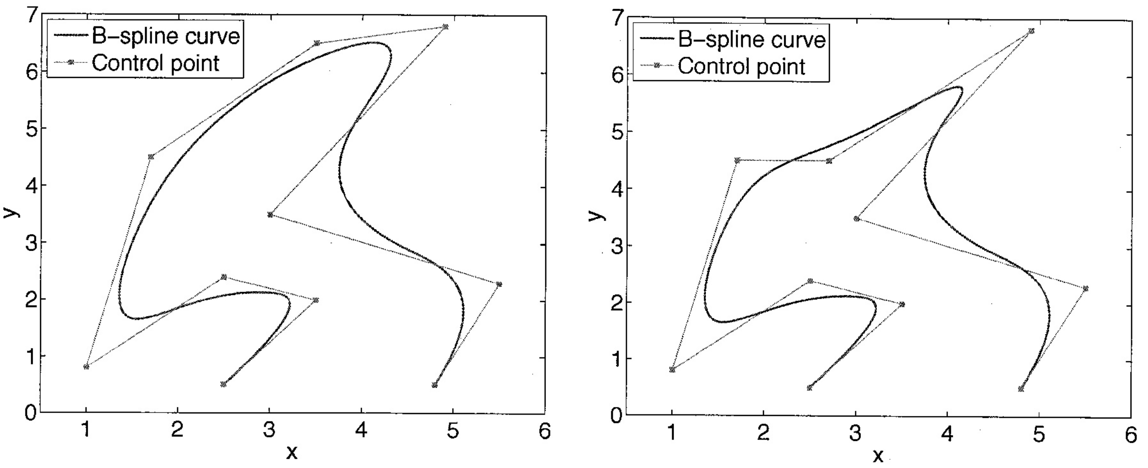
To illustrate the impact of ~~the~~ multiple knots, 4th-order basis functions computed based on three knot vectors are compared in a schematic diagram Figure 2.2. Figure 2.2(a) shows a vector consisting of only simple knots, Figure 2.2(b) displays the effect of moving t_5 to t_6, and Figure 2.2(c) shows 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 ~~behaviours~~ ~~properties~~ 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 number of non-zero basis functions 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-1-l} 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.

function, it will only affect the knot intervals where the basis function is non-zero. Figure 2.4 gives a clear~~ly~~ illustration ~~regarding~~ of this property.



(a) Original B-spline curve

(b) Modified B-spline curve

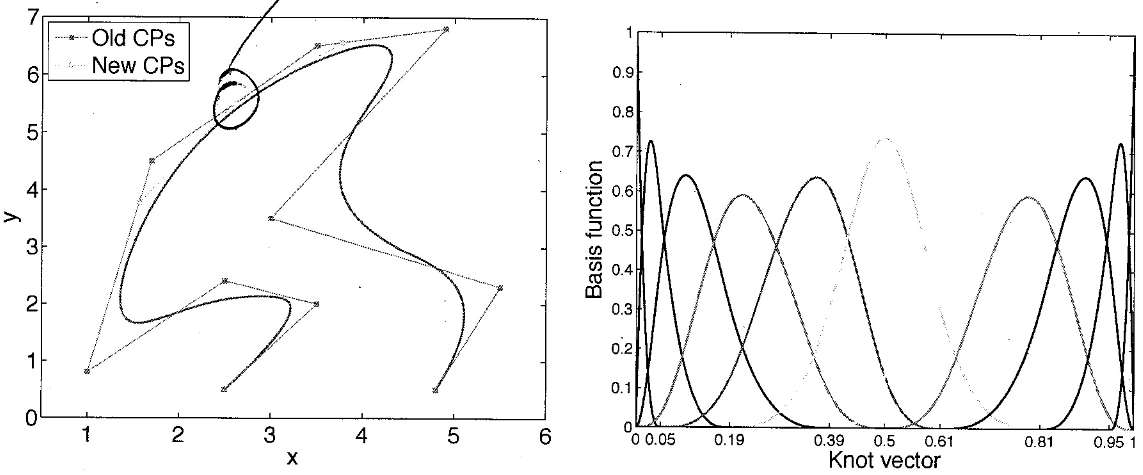
Figure 2.4: Local modification

- 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 property reveals an important ~~factor~~ 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 make this clearer



(a) Modified control points

(b) Modified basis functions

Figure 2.6: Knot insertion

$\{\mathbf{d}_i^* : i = 1, \ldots, r-k+1, r+1, \ldots, n+1\}$ retains the positions of old control points based on the local support property of B-spline formulation.

$$\mathbf{d}_i^* = \mathbf{d}_i \qquad 1 \le i \le r-k+1 \tag{2.6}$$

$$\mathbf{d}_i^* = \mathbf{d}_{i-1} \qquad r+1 \le i \le n+1 \tag{2.7}$$

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

$$\mathbf{d}_i^* = (1-\alpha_i)\,\mathbf{d}_{i-1} + \alpha_i\,\mathbf{d}_i \tag{2.8}$$

$$\alpha_i = \frac{t^* - t_i}{t_{i+k-1} - t_i} \qquad ? \tag{2.9}$$

Figure 2.6(a) shows the effect of inserting a new knot of 0.5. The 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 $\mathbf{d}^*$ between $(\mathbf{d}_r, \mathbf{d}_{r+1})$ is required, this procedure can be reversed to calculate the required knot [45]:

$$s = \frac{\mathbf{d}^* - \mathbf{d}_r}{\mathbf{d}_{r+1} - \mathbf{d}_r} \tag{2.10}$$

$$t^* = t_{r+1} + s(t_{r+k} - t_{r+1}) \tag{2.11}$$

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 determina~~te~~e the control points. Assume the airfoil surface is defined by a set of points, $\{\mathbf{P}_j, j = 1, \ldots, N\}$. The control points can be computed by solving the least squares problem, $\min \sum_{j=1}^{N} ||\mathbf{P}_j - \mathbf{X}_j||_2$. The obtained approximation curve $\{\mathbf{X}_j, j = 1, \ldots, N\}$ through this procedure has a well-known deficiency that the error vector, $\mathbf{P}_j - \mathbf{X}_j$, is usually not perpendicular to the tangent. To resolve this problem, Hoschek [25] proposes the following iterative parameter correction algorithm:

$$\bar{\xi}_j = \xi_j + \Delta c_j \frac{t_n - t_1}{L} \tag{2.16}$$

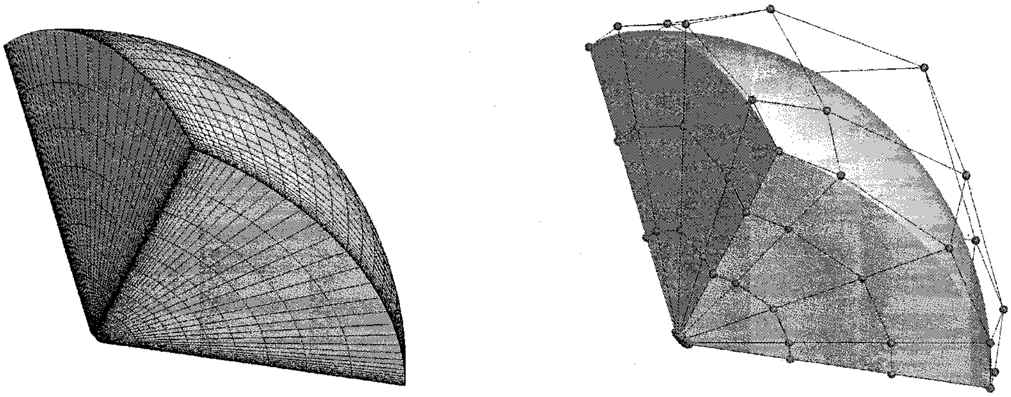
$$\Delta c_j = (\mathbf{P}_j - \mathbf{X}_j) \cdot \mathbf{Y}_j \tag{2.17}$$

where L is the total length of the control polygon defined by connecting the control points, and $\mathbf{Y}_j$ 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 ~~to the airfoil~~. 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 ~~only~~ parametrizing only the surface of the object, this approach represents the entire computational grid with B-spline volumes and ~~th~~e 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



(a) B-spline volume (b) control mesh

Figure 2.8: B-spline volume parametrization

given by

$$\mathcal{N}_{i,1}(\xi) = \begin{cases} 1 & t_i(\eta,\zeta) \le \xi \le t_{i+1}(\eta,\zeta) \\ 0 & \text{otherwise} \end{cases} \tag{2.19}$$

$$\begin{aligned} \mathcal{N}_{i,p}(\xi) = & \left(\frac{\xi - t_i(\eta,\zeta)}{t_{i+p-1}(\eta,\zeta) - t_i(\eta,\zeta)}\right)\mathcal{N}_{i,p-1}(\xi) \\ & + \left(\frac{t_{i+p}(\eta,\zeta) - \xi}{t_{i+p}(\eta,\zeta) - t_{i+1}(\eta,\zeta)}\right)\mathcal{N}_{i+1,p-1}(\xi) \end{aligned} \tag{2.20}$$

where the knot vector, $t_i(\eta,\zeta)$, is a spatially varying function of parameters, η and ζ. 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]:

$$t(\eta,\zeta) = t(0,0)(1-\eta)(1-\zeta) + t(1,0)\eta(1-\zeta) + t(0,1)(1-\eta)\zeta + t(1,1)\eta\zeta \tag{2.21}$$

Here $t(0,0)$, $t(1,0)$, $t(0,1)$ and $t(1,1)$ are four edge knot vectors in the ξ 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 result~~ed~~ing 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.

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~~s~~ in the previous chapter. An overview of the ~~rest~~ 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. [42] propose the following algorithm based on the normalized arclength from the surface:

$$\begin{bmatrix} y' \\ x' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + (1 + \cos(\pi S)) \begin{bmatrix} \Delta y \\ \Delta X \end{bmatrix} \tag{3.1}$$

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 ~~are~~ is found by the following equation:

? Your notation doesn't distinguish between Q at one node and Q at all nodes.

$$\mathcal{A}^{(n)}\Delta\mathbf{Q}^{(n)} = -\mathbf{R}^{(n)} \tag{3.4}$$

Convergence of

where $\mathcal{A}^{(n)}$ is the flow Jacobian matrix. Because the Newton~~'s~~ method depends on the ~~initial value~~ of $\mathbf{Q}$, a start-up algorithm is necessary to provide a proper initial iterate. In this solver, an implicit Euler time marching method with ~~the~~ an approximate-factorization is used as the start-up algorithm. Once an initial value~~,~~ $\mathbf{Q}^{(0)}$ is obtained, Eq. 3.4 is solved inexactly using the generalized minimal residual (GMRES) linear solver.

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

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{E}_i}{\partial x_i} = 0 \tag{3.5}$$

where $\mathbf{x} = [x1,\ x2,\ x3] = [x,\ y,\ z]$, $\mathbf{Q} = [\rho,\ \rho u_1,\ \rho u_2,\ \rho u_3,\ e]^T$ 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]). ~~Similarly, I~~f the temporal term is neglected, the Euler equations are reduced to ~~flow residual~~ nonlinear algebraic equations of the form, $\mathbf{R} = 0$. Solving the residual equation~~s~~ 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 ~~the~~ 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 objec-

Keep the same sequence of variables.

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 $\mathcal{J} = \mathcal{J}(\mathbf{Q}, \mathbf{X}, \mathbf{b}^{(m)})$, and the dependence of the flow residual equations can be explicitly expressed ~~as~~ $\mathbf{R} = \mathbf{R}(\mathbf{Q}, \mathbf{b}^{(m)}, \mathbf{X})$, where $\mathbf{b}^{(m)}$ are the volume control points and $\mathbf{X}$ are the design variables. The derivation of the gradient is done through Lagrangi~~a~~en multipliers, which requires an optimal point to satisfy the flow residual equations and grid movement equations:

$$\begin{aligned} \min \quad & \mathcal{J}(\mathbf{Q}, \mathbf{X}, \mathbf{b}^{(m)}) \\ \text{w.r.t} \quad & \mathbf{Q}, \mathbf{X}, \mathbf{b}^{(m)} \\ \text{s.t.} \quad & \mathbf{R}(\mathbf{Q}, \mathbf{b}^{(m)}, \mathbf{X}) = 0 \\ & \mathbf{r}^{(i)}(\mathbf{b}^{(i)}, \mathbf{b}^{(i-1)}) = 0, \quad i \in \{1, \ldots, m\} \end{aligned}$$

The Lagrangian is defined by

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

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

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

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\psi}} = 0 = \mathbf{R} \tag{3.10}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{Q}} = 0 = \frac{\partial \mathcal{J}}{\partial \mathbf{Q}} + \boldsymbol{\psi}^T \frac{\partial \mathbf{R}}{\partial \mathbf{Q}} \tag{3.11}$$

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

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

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

Follow the strategy proposed by Truong et al. [51], Eq. 3.9-Eq. 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]), S~~ince the optimality condition requires it~~ to ~~vanishes at the optimal point, this it is~~ ~~regarded~~ ~~as~~ the gradient ~~in an~~ ~~optimization~~:

$$\mathcal{G} = \frac{\partial \mathcal{J}}{\partial \mathbf{X}} + \sum_{i=1}^{m} \left(\boldsymbol{\lambda}^{(i)T} \frac{\partial \mathbf{r}^{(i)}}{\partial \mathbf{b}^{(i)}} \frac{\partial \mathbf{b}^{(i)}}{\partial \mathbf{X}} \right) + \boldsymbol{\psi}^T \frac{\partial \mathbf{R}}{\partial \mathbf{X}} \tag{3.15}$$

which provides

Eqs. 3.9 to 3.13

Depending on the treatment of the constraints, two optimization algorithms are implemented. One of them is the BFGS quasi-Newton algorithm combined a~~ssocia~~ted with a backtracking line search method. This optimizer is design~~at~~ed 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:

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

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 ~~an~~ a strategy that adjusts the angle of attack in the flow solver to satisfies the lift constraint, and this is used as an alternative treatment for 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, the constraints are checked to prevent large violations. This is a bit vague. What do you do if a large violation occurs?

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~~s~~ 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 s~~tratifi~~ed satisfied, and this property is used to establish the necessary coupling among control points. Unless otherwise stated, the SNOPT package is used as the default optimizer.

tive control points among all the candidates, we have established the following selection criteria:

1. The number of parameters in ~~the~~ 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 [35, 4] argue that when the number of design variables is small, relatively few ~~number of~~ 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 ~~beginning~~ 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.

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 ~~by~~ 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 ad 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 to (t_r, t_{r+1}), the new control points, $\{\mathbf{d}_i : i = r-2, \ldots, r\}$, would be relocated. To ensure the appearance of new control points dose not affect the frozen control points, $\{\mathbf{d}_i : i = 1, 7, 8, 9, 15\}$, 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

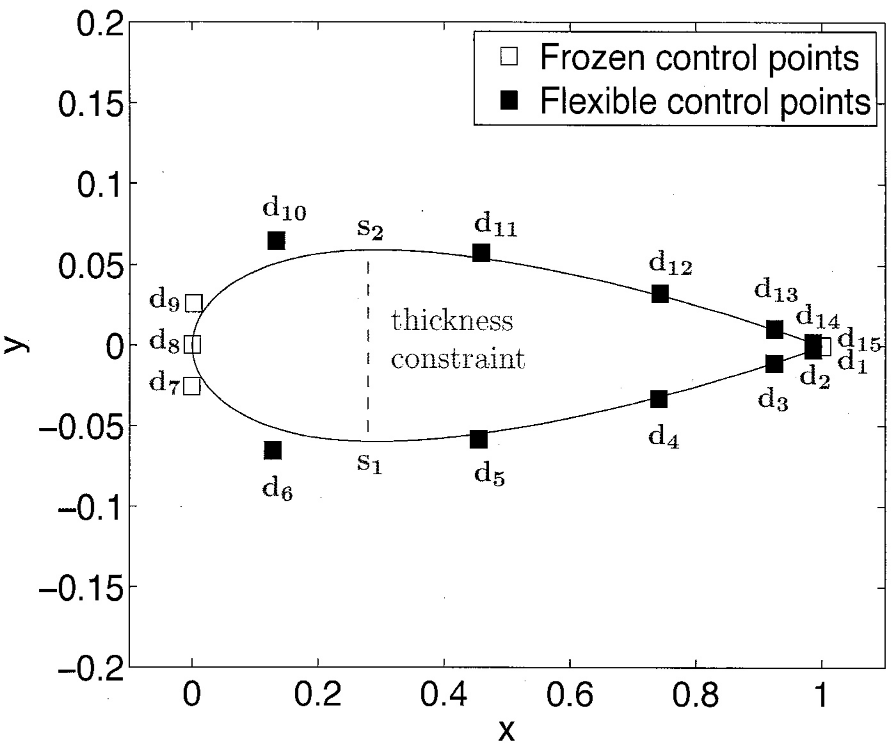


Figure 4.1: NACA0012 airfoil parametrized using 15 control points

Move these 2 figures to previous page

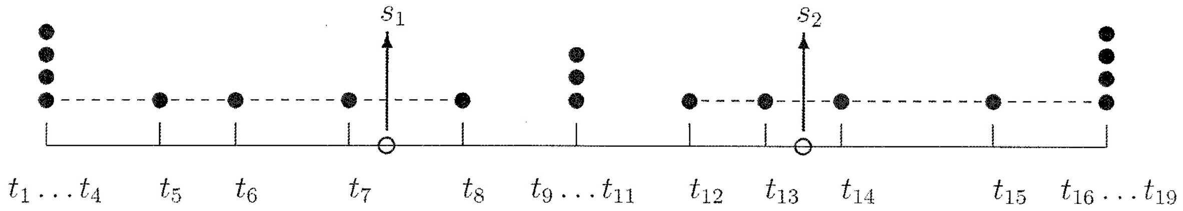


Figure 4.2: Knot sequence

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

The design objective is to ~~reduce~~ 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 adding 6 control points, because the subsequent

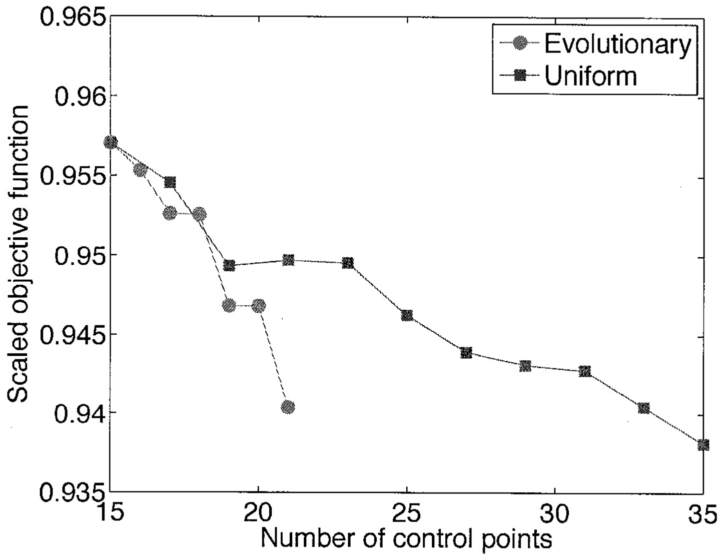


Figure 4.4: Objective function comparison of Case 1

which is #6 on that plot?

if the new point has a relatively large sensitivity, then how can the optimality value be so small?

provement is not significant. Evidence~~s~~ 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 cycles. In general, ~~I should emphasize tha~~t the optimization sequence with evolutionary parametrization gives a systematic way of selecting design variables, and its optimization efficiency is superior to the uniform parametrization. → What do you mean by optimization efficiency? Isn't it actually faster to use 33 control points and converge only once?

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 $J = \frac{C_d}{C_l} + \omega_T T$ with the applied weight being 1.0.

Because the thickness constraints are handled by the penalty method~~s~~, a slight violation is observed when the constraints are active. Therefore, the reported lift-to-drag ratio is a little lower than the true value. Figures 4.5(a) and ~~Figure 4.5~~(b) reveal the change of the airfoil shape and pressure distribution.

reported where? You can either plot $\frac{L}{D}$ or J. This should not be an issue.

The scaled lift-to-drag ratios obtained from the proposed sequence and the optimiza-

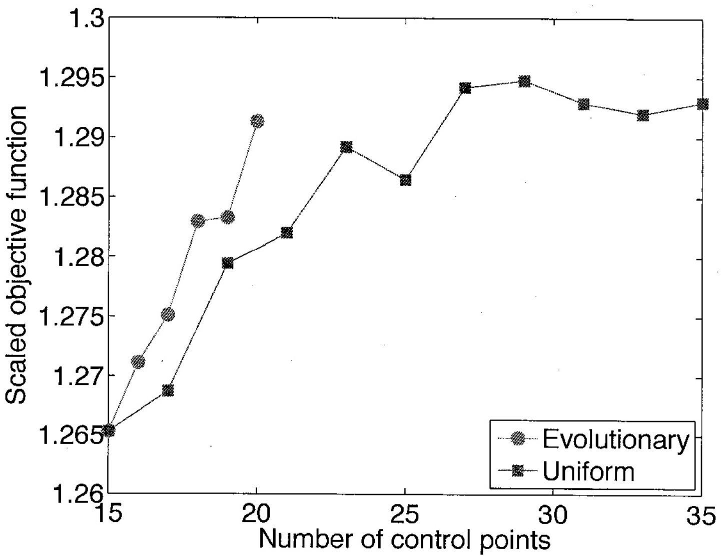


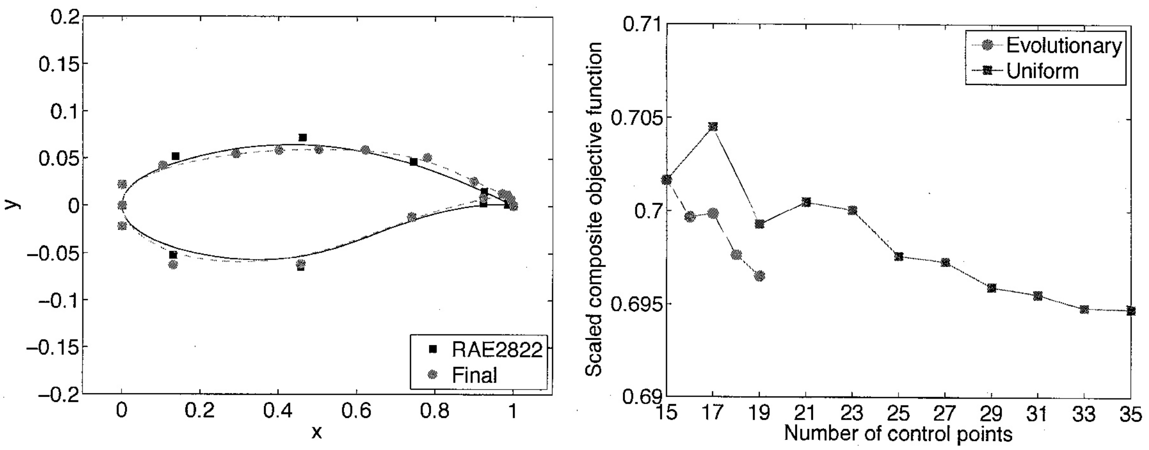
Figure 4.6: Objective function comparison of Case 2

Table 4.3: Geometric constraints for Case 3

Constraints	1	2	3
Location (c)	0.96	0.99	-
Thickness (c)	0.006	0.0012	-
Area (c^2)	-	-	0.07761

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.

To provide a comparison for optimization efficiency, this result is compared with a series of optimal solutions 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, the consistent reduction of objective function is observed from the optimization sequence with evolutionary parametrization. Using 21 control points, the



(a) Optimized geometry

(b) Scaled composite objective function

Figure 4.9: Optimized geometry and objective function comparison of Case 4

0.733, and the freestream flow has a Raynolds e number of 9×10^6. The angle of attack for each operating point is adjusted to meet the lift requirement. The thickness constraints are identical to Case 3, but the area constraint is slightly modified to $0.07790(c^2)$. why? what was the area of the RAE 2822?

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 adjusted 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 from various uniform parametrizations. In Figure 4.9(b), it can be seen that the sequence with evolutionary parametrization ~~still~~ outperforms the conventional uniform parametrizations ~~as the complexity of the problem increases~~. Thus the optimization efficiency can be improved by effectively selecting critical design variables.

Cost?
CPU time?
# of iterations?

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 ~~is effective finding~~ the optimal aerodynamic shape through a few test cases.

effectively finds

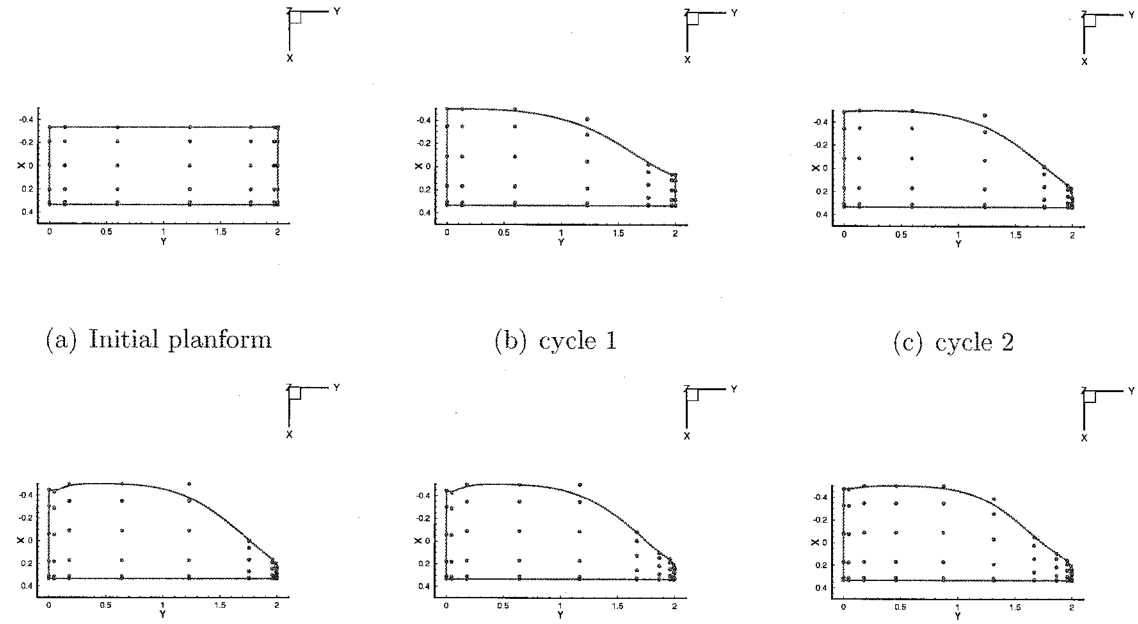
5.1 Test Case Setup

When investigat~~e~~ing induced drag reduction, the classical Prandtl's 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:

$$C_{D,min} = \frac{C_L^2}{\pi \Lambda} \tag{5.1}$$

where C_D and C_L are drag and lift coefficient, respectively, and $\Lambda = b^2/S$ is the aspect ratio. The reference area, S, used to compute the coefficients and the aspect ratio in the test cases is chosen to be the planform area. Don't forget to define b.

In the following test cases, two types of computational grids are used. The first one is a 12-block flat-plate grid which has an "H-H" topology, and the other 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.



(a) Initial planform

(b) cycle 1

(c) cycle 2

(d) cycle 3

(e) cycle 4

(f) cycle 5

Figure 5.1: Planform deformation

with how many spanwise control points?

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], ~~and they~~ who point out that the presence of the shape tip ? causes the vortex to release along the tip edge, which ultimately results in a nonplanar wake. Nevertheless, the purpose of this example is to demonstrate the effectiveness of the evolutionary parametrization. In terms of the performance of the optimal design, the drag coefficient is reduced from 0.006769 to 0.006606, a roughly 2.4% drop, ~~is achieved~~. The span efficiency of the final geometry is 0.9838, which is close to the efficiency of 0.988 obtained by Hicken and Zingg [20] using 15 spanwise control points. Figure 5.2(b) compares the evolutionary parametrization with uniformly spaced parametrizations (the optimization with 10 spanwise control points does not converge, thus it is not plotted). The latter do not receive significant benefits from adding effective design variables, and their performance is inferior ~~comparing~~ compared to the evolutionary parametrization with the same number of spanwise control points.

same grid?

it looks pretty close, doesn't it?

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.

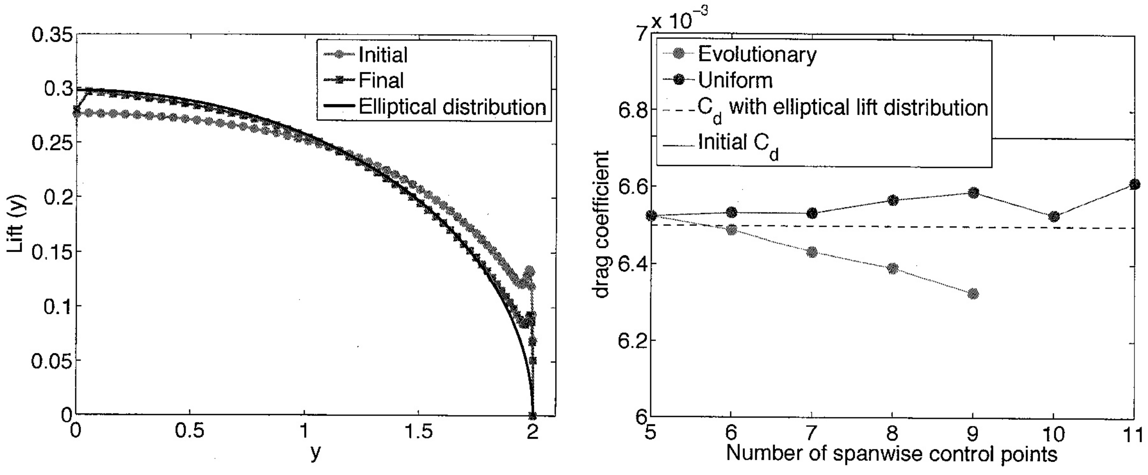
The shape deformations are depicted in Figure 5.3, and the lift distributions are shown in Figure 5.4(a). To better illustrate the performance of the obtained geometry, the drag coefficient is plotted with the number of spanwise control points (Figure 5.4(b)). It can be seen from these diagrams that the optimal solution using the evolutionary parametrization does not have a elliptical planform or lift distribution, and the final drag coefficient is reduced to 0.006326 which is 2.65% lower than the lifting line prediction. The ~~deficiency~~ difference may be attributed to the edge separation effect and ~~the~~ numerical errors. A grid refinement study is conducted, and the resolution in every direction is increased by a factor of 2.0, which generates a grid with 9.3 million nodes. The span efficiencies obtained with the refined grid and the coarse grid are 1.029 and 1.027; this small difference suggests that ~~the~~ numerical errors are not the domina~~ted~~nt factor~~s~~. Besides the theoretical value, the comparison is also conducted between evolutionary parametrization refinements and the uniform refinements. The former achieves a 6.0% induced drag reduction, but the latter are unable to improve upon the result obtained with the initial parametrization, which leads to only 50% of the improvement achieved using the evolutionary parametrization. ~~refinements~~.

I'm a bit confused. Do you expect this case to outperform the elliptical planform (based on Liersch et al.) or not?

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, 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,



(a) Lift distribution

(b) C_D vs. number of control points

how?

Figure 5.4: Planform and twist optimization

a

coupled with the corresponding leading and trailing edges such that the thickness of the section is maintained. Also, to generate 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 improvement, so the optimization is terminated with 9 spanwise stations. The drag coefficient is reduced from 0.006730 to 0.005835, and the span efficiency is increased from 0.9656 to 1.1137. This result is close to the performance of the optimized winglet configuration obtained by Hicken and Zingg [20] as they reported a span efficiency of 1.147 using 13 spanwise stations for the upward winglet. This optimization result is compared to the optimal solutions if control points are uniformly distributed. Figure 5.6 shows that the

th where

unless their grid is similar, this doesn't mean much.

constraint

the

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, and the angle of attack is fixed to reduce the chance of non-unique optimal designs and its value is chosen to meet thi~~s~~e ~~target~~ lift with 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 geometry deformation at the end of each cycle. The process is terminated because sufficient a amount of control points have been placed on the vertical plate. 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, and the drag coefficient is reduced by a considerable amount. However, the optimization results with uniformly placed control points do not exhibit much dependence on the enriched design variables, and the optimization efficiency is clearly lower than the proposed sequence.

this has never been defined

5.2.5 Case 5: Flexible wing optimization

The last test case is an exploratory example. The objective is to use the proposed optimization sequence to produce a wing design that minimizes the induced drag. The baseline geometry, reference area, lift constraint and operating conditions are identical

In this case, no effective design variables are specified, i.e. the control point coordinates are the design variables / but to maintain the integrity of the grid, a series of linear constraints are imposed and for each individual degree of freedom, upper and lower limits are specified:

1. The control point at the root section trailing edge is fixed to eliminate translational motion.

2. The control points at the root section leading edge, tip section leading and trailing edge are free to move in three directions.

3. The leading and trailing edge control points can move in the x and z directions, and their y coordinates are interpolated between the root and tip points.

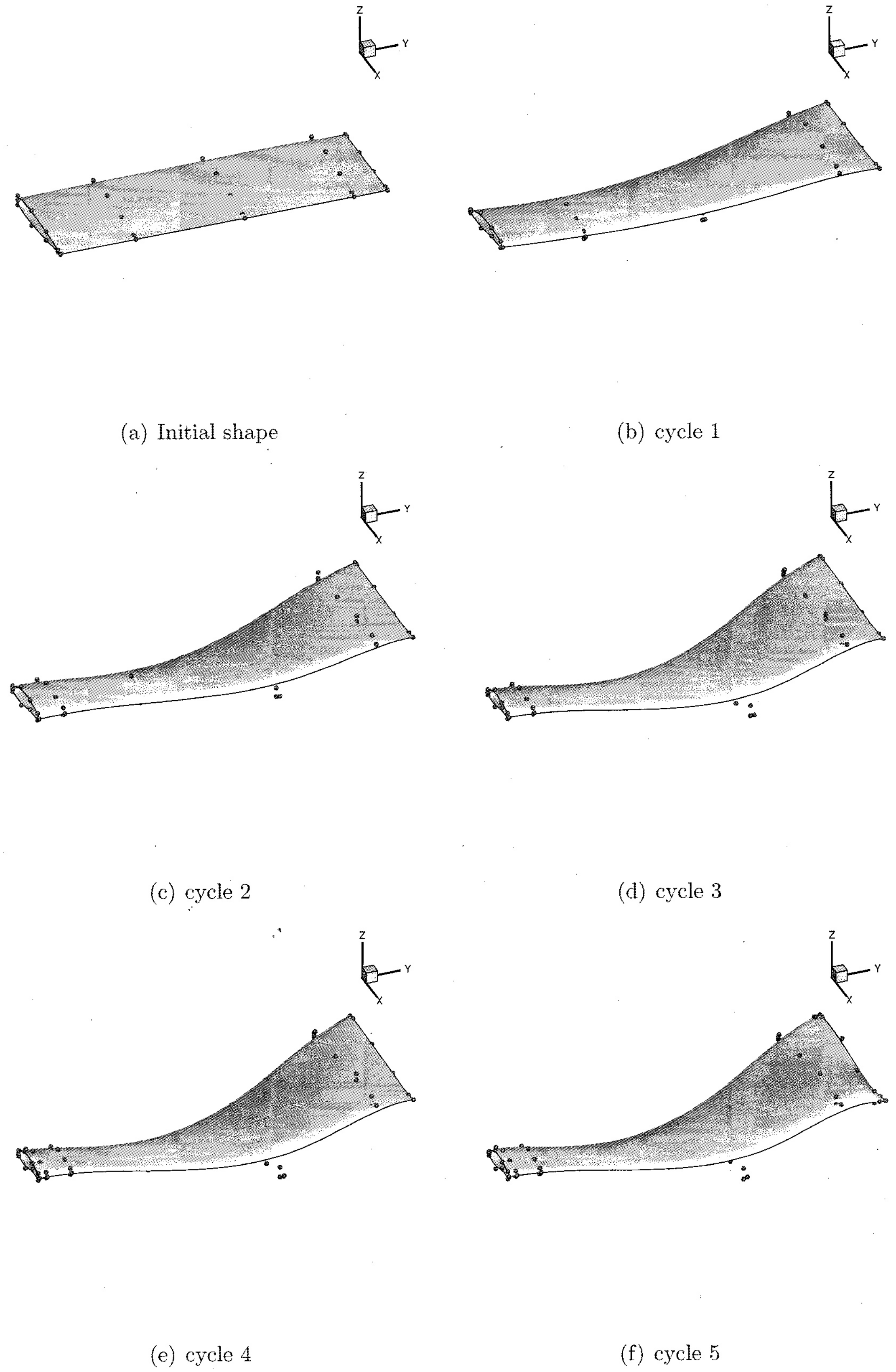
4. The control points at the tip can adjust in the y and z directions, and their x coordinates are obtained by scaling the leading and trailing edgeS.

5. The interior points and the points on the symmetry plane only have the degree of freedom in the z direction~~s~~, and their horizontal locations are determined by interpolations among the boundaries.

6. The entire configuration is confined within $|x| < 2/3$, $|y| < 2.2$, $|z| < 0.22$.

The parametrization refinements allow additional control points to be inserted in both streamwsie and spanswise directions. However, adding points close to the trailing edge is excluded because excessive manipulations of the trailing edge would produce unsteady flows. Besides modifying the number of control points, the upper and lower limits of each degree of freedom are also examined at the end of every optimization cycle. If one control point reaches a certain bound, it is released in the next cycle to provide additional freedom. As a result, this optimization sequence gradually increases the number of design variables and progressively releases the ~~bounds of~~ constraint~~s~~. bounds. → Does this apply to all bounds?

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 occurr~~ed~~ing 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



(a) Initial shape

(b) cycle 1

(c) cycle 2

(d) cycle 3

(e) cycle 4

(f) cycle 5

Figure 5.9: Shape changes of the flexible wing optimization

Is there a way to show control points that are below the geometry?

sistent feature 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 the evolutionary parametrization ~~clearly indicates~~ demonstrate a substantial ~~gain~~ improvement in the design objective. ~~is accomplished.~~

6.1 Future Work

where do these come in?

6.1.1 Automation

The current process is not fully automated since the implementation of parametrization refinements and the termination conditions still requires experimentation and tu~~n~~ning. Since sensitivity analysis can only provide a local measure of the potential of a design variable~~.~~, a ~~M~~more general treatment is necessary to utilize sensitivity analysis and design constraints.

6.1.2 Evolutionary parametrization with adaptation

The linear couplings among control points sometimes prevent adding essential control points or restrict the performance of the additional control points. As presented in the exploratory case of a flexible wing, inequality relations with adaptive bounds may provide more freedom to the optimizer and enhance the impact of the introduced design variables.

I don't think you've made this clear enough in 5.2.5. Why loosen the bounds gradually? Why set them in the first place?

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

Why some first names?

References

[1] Natalia M. Alexandrov, Robert Michael Lewis, Clyde R. Gumbert, Lawrence L. Green, and Perry A. Newman. Approximation and model management in aerodynamic optimization with variable-fidelity models. *Journal of Aircraft*, 38(6):1093–1101, 2001.

[2] John D. Anderson. *Foundamentals of Aerodynamics.* McGraw-Hill, 2001.

[3] M. Andreoli, A. Janka, and J. Desideri. Free-form-deformation parameterization for multilevel 3d shape optimization in aerodynamics. Technical Report 5019, INRIA, 2003.

[4] F. Beux and A. Dervieux. A hierarchical approach for shape optimization. *Engineering Computations*, 11(6):25–38, 1994.

[5] V. Braibant and C. Fleury. Shape optimal design using b-splines. *Computer Methods in Applied Mechanics and Engineering*, 44(3):247–267, 1984.

[6] P. Castonguay and S. Nadarajah. Effect of shape parameterization on aerodynamic shape optimization. In *AIAA-2007-0059, 45th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, 8-11 January 2007.

[7] A. Clarich and J. Desideri. Self-adaptive parameterisation for aerodynamic optimum-shape design. Technical Report 4428, INRIA, 2002.

[8] C. de Boor. *A Practical Guide to Splines.* Springer-Verlag Berlin and Heidelberg GmbH & Co. K, December 1978.

[9] J. Desideri. Hierarchical optimum-shape algorithms using embedded bezier parameterizations. In *Numerical Methods for Scientific Computing, Variational Problems and Applications*, 2003.

[21] J.E. Hicken and D. W. Zingg. A parallel newton-krylov solver for the euler equations discretized using simultaneous approximation terms. *AIAA J.*, 46:2773–2786, 2008.

[22] J.E. Hicken and D. W. Zingg. Aerodynamic optimization algorithm with integrated geometry parameterization and mesh movement. *AIAA J.*, 48:401–413, 2010.

[23] R. Hicks and P. Henne. Wing design by numerical optimization. *Journal of Aircraft*, 15:407–412, 1978.

[24] T. Holst and T. Pulliam. Aerodynamic shape optimization using a real-number-encoded genetic algorithm. In *19th AIAA Applied Aerodynamics Conference*, number AIAA 2001-2473, Anaheim, California, June 2001.

[25] J. Hoschek. Intrinsic parametrization for approximation. *Comput. Aided Geom. Des.*, 5(1):27–31, 1988.

[26] J. Hoschek and D. Lasser. *Fundamentals of Computer Aided Geometric Design*. A. K. Peters, Ltd., Natick, MA, USA, 1993. Translator-Schumaker, Larry L.

[27] A. Jameson. Aerodynamic design via control theory. *J. Sci. Comput.*, 3(3):233–260, 1988.

[28] R. T. Jones. *Wing Theory*. Princeton University Press, 1990.

[29] I. Kroo. Drag due to lift: Concepts for prediction and reduction. *Annual Review of Fluid Mechanics*, 33(1):587–617, 2001.

[30] M. Kulfan. Universal parametric geometry representation method. *Journal of Aircraft*, 45(1):142–158, 2008.

Can you please send me this paper?

[31] C. Liersch, T. Streit, and K. Visser. Numerical implications of spanwise camber on minimum induced drag configurations. In *47th AIAA Aerospace Science Meeting and Exhibit*, number AIAA Paper 2009-898, Jan. 2009.

[32] H. Lomax, T. Pulliam, and D. Zingg. *Fundamentals of Computational Fluid Dynamics*. Spinger-Verlag, Berlin, 2001.

[33] B. Majd, R. Duvigneau, and J. Desideri. Aerodynamic shape optimization using a full and adaptive multilevel algorithm. In *ERCOFTAC Conference Design Optimization: Methods and Applications*, Canary Island, Span, April 2006.

Montreal Protocol on Substances that Deplete the Ozone Layer. Cambridge University Press, Cambridge, 1999.

[45] L. Piegl. Modifying the shape of rational b-splines. part 1: Curves. *Computer-Aided Design*, 21(8):509–518, Oct. 1989.

[46] G. Robinson and A. Keane. Concise orthogonal representation of supercritical airfoils. *Journal of Aircraft*, 38(3):580–583, 2001.

[47] J. Samareh. Survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization. *AIAA J.*, 39(5):877–884, 2001.

Can you send me this one too?

[48] P. Sherar, C. Thompson, B. Xu, and B. Zhong. An optimization method based on b-spline shape functions & the knot insertion algorithm. In *World Congress on Engineering*, pages 862–866, 2007.

[49] W. Song and A. Keane. A study of shape parameterization methods for airfoil optimisation. In *AIAA 2004-4482, 10th AIAA/ISSMO Multidisciplinary Analysis and Optimisation Conference*, Albany, New York, 30 August - 1 September 2004.

[50] Z. Tang and J. Desideri. Towards self-adaptive parameterization of bezier curves for airfoil aerodynamic design. Technical Report 4572, IRNIA, 2002.

[51] A.H. Truong, C.A. Oldfield, and D. W. Zingg. Mesh movement for a discrete-adjoint newton-krylov algorithm for aerodynamic optimization. *AIAA J.*, 46:1695–1704, 2008.

[52] A. Vicini and D. Quagliarella. Airfoil and wing design through hybrid optimization strategies. *AIAA J.*, 37(5):634–641, 1999.

[53] D. Zingg and L. Billing. Toward practical aerodynamic design through numerical optimization. In *AIAA 2007-3950, 18th AIAA Computational Fluid Dynamics Conference*, Miami, FL, June 2007.

[54] D. W. Zingg, M. Nemec, and T. H. Pulliam. A comparative evaluation of genetic and gradient-based algorithms applied to aerodynamic optimization. *REMN*, 17:103–126, 2008.

A.1 Airfoil Optimization

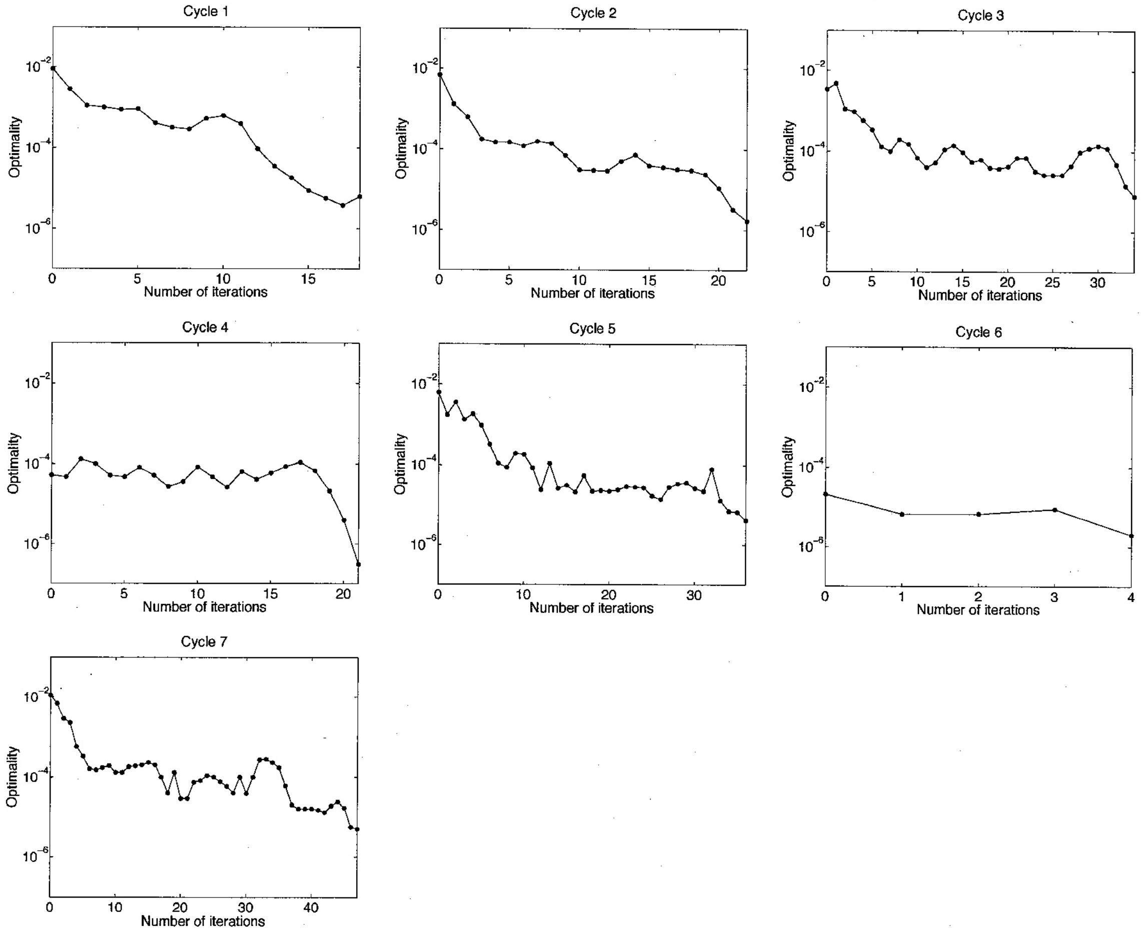


Figure A.1: Convergence ~~H~~h~~istory~~istories for airfoil optimization case 1

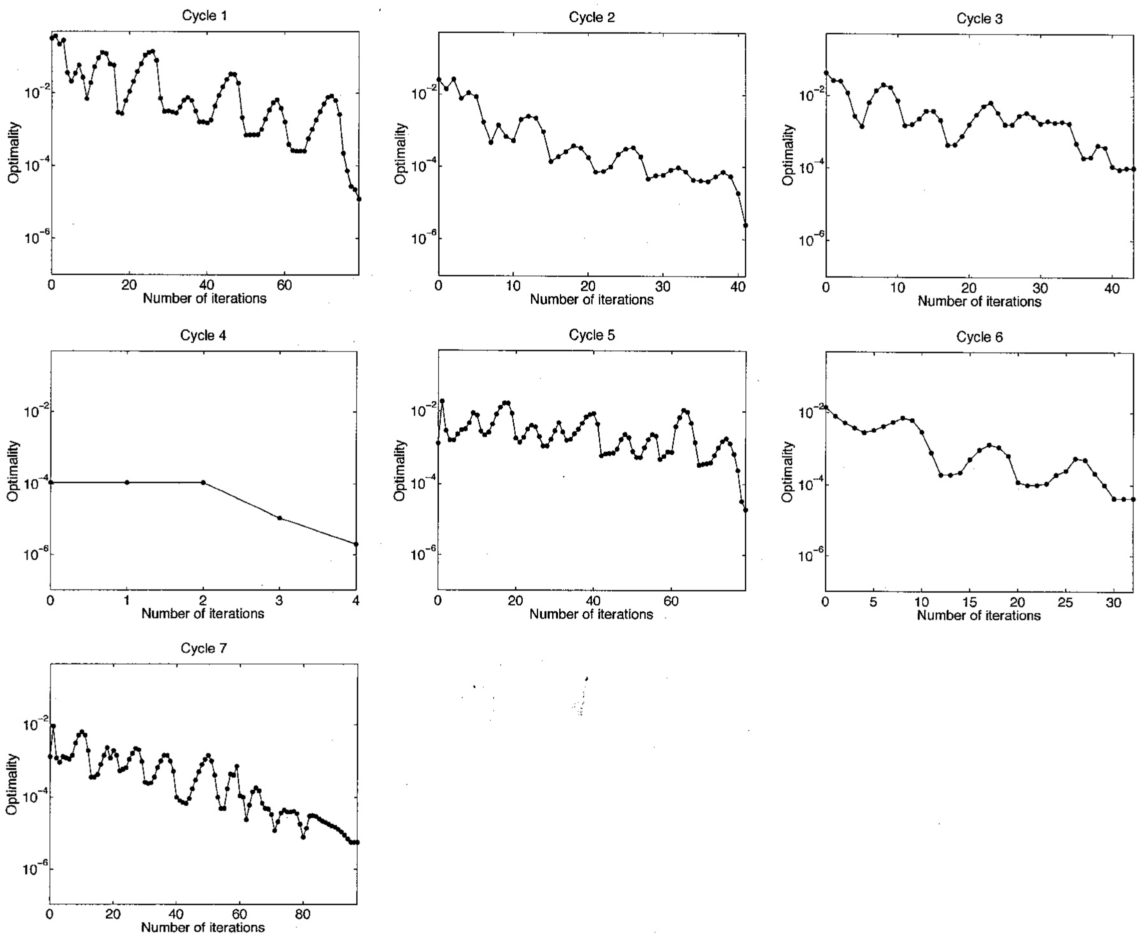


Figure A.3: Convergence hHistory(es) for airfoil optimization case 3

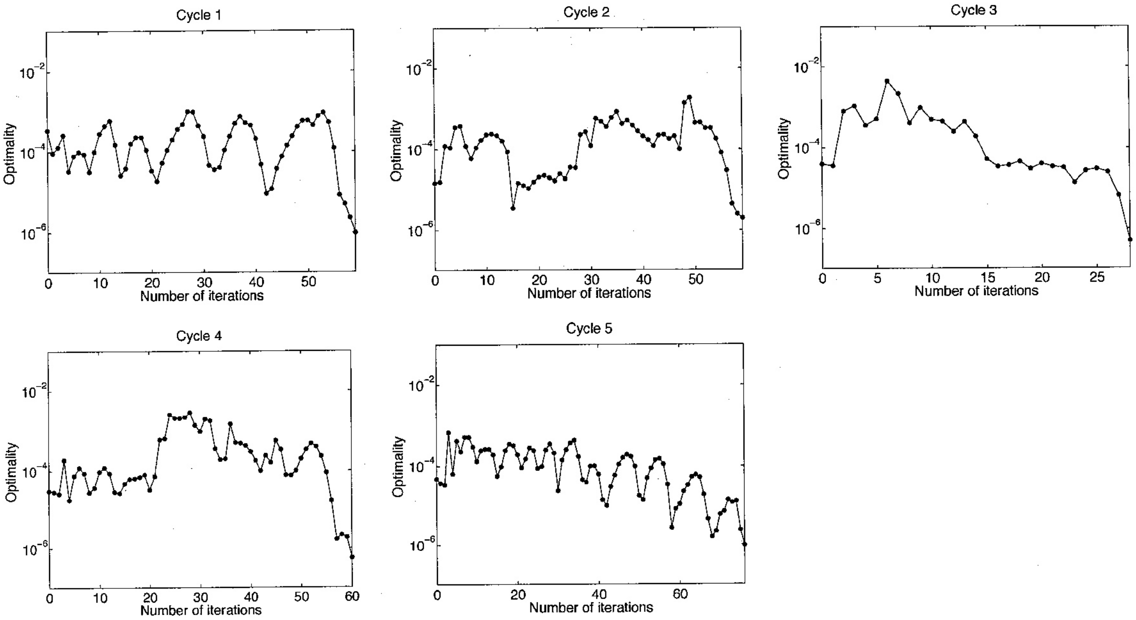


Figure A.6: Convergence ~~H~~history for planform and twist optimization

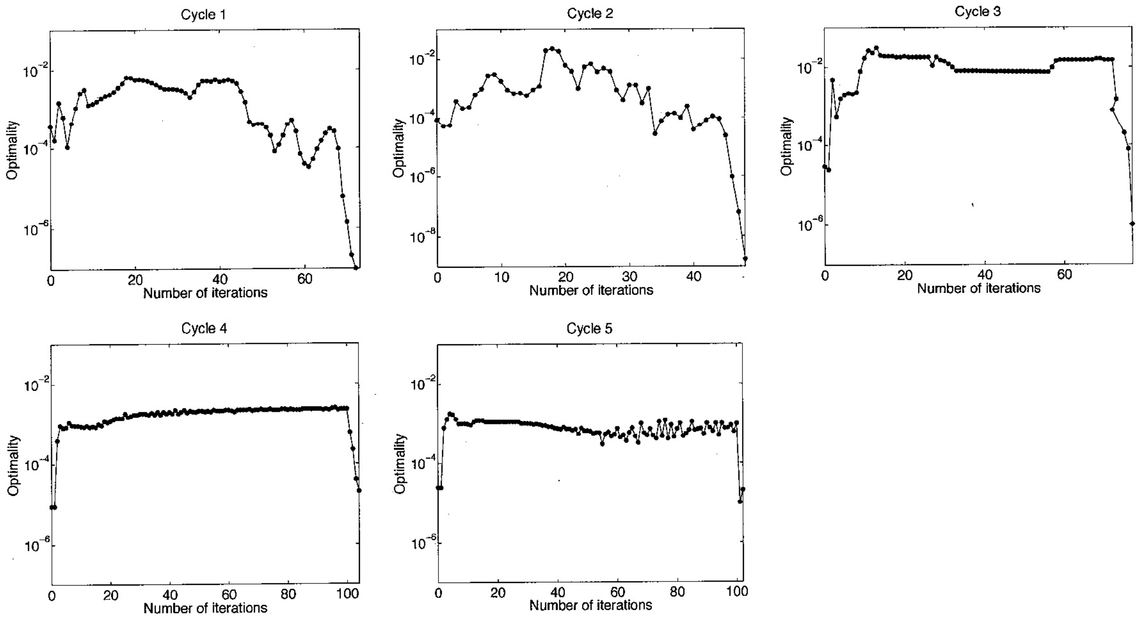


Figure A.7: Convergence ~~H~~histor~~y~~ies for winglet optimization