Gradient-Based Optimum Aerodynamic Design Using Adjoint Methods

Lei Xie

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Aerospace Engineering

Dr. Eugene M. Cliff, Chairman
Dr. Bernard Grossman
Dr. William Mason
Dr. Jeff Borggaard
Dr. William Devenport

April 12, 2002
Blacksburg, Virginia

Keywords: Adjoint Method, Boundary Conditions, Optimum Aerodynamic Design,
Transonic Airfoil Design

Gradient-Based Optimum Aerodynamic Design Using Adjoint Methods

Lei Xie

(ABSTRACT)

Continuous adjoint methods and optimal control theory are applied to a pressure-matching inverse design problem of quasi 1-D nozzle flows. Pontryagin's Minimum Principle is used to derive the adjoint system and the reduced gradient of the cost functional. The properties of adjoint variables at the sonic throat and the shock location are studied, revealing a logarithmic singularity at the sonic throat and continuity at the shock location. A numerical method, based on the Steger-Warming flux-vector-splitting scheme, is proposed to solve the adjoint equations. This scheme can finely resolve the singularity at the sonic throat. A non-uniform grid, with points clustered near the throat region, can resolve it even better. The analytical solutions to the adjoint equations are also constructed via Green's function approach for the purpose of comparing the numerical results. The pressure-matching inverse design is then conducted for a nozzle parameterized by a single geometric parameter.

In the second part, the adjoint methods are applied to the problem of minimizing drag coefficient, at fixed lift coefficient, for 2-D transonic airfoil flows. Reduced gradients of several functionals are derived through application of a Lagrange Multiplier Theorem. The adjoint system is carefully studied including the adjoint characteristic boundary conditions at the far-field boundary. A super-reduced design formulation is also explored by treating the angle of attack as an additional state; super-reduced gradients can be constructed either by solving adjoint equations with non-local boundary conditions or by a direct Lagrange multiplier method. In this way, the constrained optimization reduces to an unconstrained design problem. Numerical methods based on Jameson's finite volume scheme are employed to solve the adjoint equations. The same grid system generated from an efficient hyperbolic grid generator are adopted in both the Euler flow solver and the adjoint solver. Several computational tests on transonic airfoil design are presented to show the reliability and efficiency of adjoint methods in calculating the reduced (super-reduced) gradients.

Dedication

To Yan Yang.

Acknowledgments

I would like to thank Dr. Eugene M. Cliff, my advisor, for his invaluable guidance, constant support, and constructive criticism during the research. His vast knowledge of mathematics and sharp skills for the optimization methods have proved irreplaceable in this work. I must thank committee members for their help to my research and comments to my dissertation. Thanks also go to Dr. Chao Yan, under whom I began the research in the optimum aerodynamic design.

Of course, I am grateful to my wife, my parents, my sister for *love* and encouragement. Without them this work would never have come into existence.

Finally, I wish to thank the following: A. Shenoy (for providing code of the Euler flow solver); Lihai Qin, Xiaofeng Wu, Yanping Wang (for their friendships); Rulong Ma and Donghui Chen (for useful discussions in research).

Blacksburg, Virginia — Lei Xie

April 12, 2002

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 General Problem Description

Historically, aerodynamic design relied on theoretical analyses and experimental "cut-and-try" methods. With the substantial developments of Computational Fluid Dynamics (CFD) in the past 30 years, CFD has become an important tool not only for simulating and analyzing flow fields, but also in aerodynamic conceptual and preliminary design [57, 47]. To prepare for this challenge, at least two research directions should be emphasized: first to develop reliable CFD codes to simulate the flow fields, and second to develop optimization techniques that can be easily integrated with the flow analysis codes.

Jameson [71] decomposed CFD-based aerodynamic design methods into two categories: inverse methods and numerical optimization methods. The inverse design methods are used to generate the prescribed distribution of some aerodynamic quantities, such as pressure or velocity, either along the surfaces (inverse surface design) or over the whole flow field (inverse field design). This method requires experience to specify the target pressure distribution or velocity distribution that satisfies the design requirements. In some cases the inverse design problems are not well-posed, and therefore are reformulated to least squares minimization

problems. Detailed discussions are provided by Labrujère and Slooff [51] and by Narducci, Grossman, *et al.* [64]. A hybrid inverse method proposed by Davis [18] and employed by Campbell & Smith [13] was very useful in pressure-matching problems. The discrepancies between the calculated and the target pressure distribution were related to the local slope and curvature of the geometric shape; then a tri-diagonal system was solved to find the change of shape. Santos & Sankar [74] coupled this inverse design method with a Navier-Stokes flow solver to address a transonic airfoil design, and with a transonic full-potential flow solver to address wing design. Later, Kim & Rho [49] extended this method to the dual-point design of transonic airfoils.

In numerical optimization design, the CFD and optimization processes are coupled to modify the design parameters, such as the shape of nozzles, airfoils, wings or the angle of attack (AoA), to improve the chosen cost functional subject to geometric and/or aerodynamic constraints. In some approaches the coupling of CFD and optimization process is subtle in that CFD is used to generate a response surface model of the aerodynamic behavior [2]. The optimization process uses the response surface as a surrogate for CFD analyses.

Previous studies in optimal aerodynamic design were mainly focused on shape optimization which consists of finding a geometric shape to minimize some performance functional(s) without violating specific geometric and/or aerodynamic constraints. In general, the geometric shapes or their changes are parameterized by a set of geometric design variables. General shape optimization has been explored by Sokolowski & Zolesio in [80], and by Lions in [54]. Their discussions are focused on problems governed by linear ordinary differential equations ($ODEs$) or partial differential equations ($PDEs$). Pironneau first proposed a minimum-drag problem in Stokes flow [67] and in attached laminar flow [68]. Because of the lack of CFD tools in those days, he only derived the necessary optimality conditions for several problems, leaving numerical issues aside. Hicks & Henne [33] studied the wing design problem in transonic inviscid flows using numerical optimization techniques. They used numerical solutions to the full-potential equations to calculate the lift and drag coefficients and employed the finite difference method (FDM) to obtain the geometric gradients. The geometric design

parameters were introduced by shape functions. The same methodology was utilized by Lee & Eyi [52] to study the transonic airfoil design with flow field governed by Euler equations. In [2], the aerofunctions that translated the pressure changes into geometry changes were employed to carry out transonic airfoil designs. However, optimum aerodynamic design is not limited to shape optimization. Not only the geometric design variables are involved, but some design variables (angle of attack, Mach number) related to the freestream may also be considered in a realistic optimum aerodynamic design process. A typical example consists of finding proper shape of airfoil and the angle of attack of the freestream to minimize the drag coefficient at given lift coefficient and at a specific freestream Mach number.

Various optimization methods have been developed for optimum aerodynamic design, and can be categorized into two kinds: non-gradient methods and gradient-based methods. In the former kind of method, no gradient information for the cost/constraint functionals is needed by the optimizer. For example, the genetic algorithm (GA) , which represents the design variables via discrete binary strings [91], performs a set of operations analogous to the biological evolution theory to search for the optimal solutions. Holst and Pulliam [36] extended a genetic algorithm to real number encoding to improve the time efficiency.

In [77, 78], an all-at-once method was described in which the discretized Euler equations are treated as the constraints to be satisfied at convergence, but not necessarily at each iteration. In other words, the optimizer searches for feasibility and optimality at the same time. Therefore, the governing Euler equations are expected to be satisfied at the same time as the optimality conditions are satisfied. Thus, the term "all-at-once". Even though this method results in a huge design space, this approach makes the design problem less nonlinear.

In the gradient-based optimization design the gradients of cost/constraint functionals with respect to design variables are important information in the design processes. The ideal way to obtain gradient information is to differentiate an analytical solution to the system. Although the analytical solution could be found for some easy cases governed by Laplace's

equations or Possion's equations in regions of simple geometry, it is impossible to obtain analytical solutions for most aerodynamic problems. Perturbation analysis [20] provides another insight into the gradient or sensitivity information. However this method can not lend itself to the complex flow problems. A traditional approach to calculate gradients is the finite difference method which involves finite differencing performance functionals computed from the solutions to the governing equations with perturbed parameter values. This method has been employed by Hicks & Henne [33] to find the steepest descent direction in 3-D wing design, and by Lee & Eyi [52] in 2-D transonic airfoil designs. FDM with complex variables was explored by Anderson *et al.* in [4] to improve its accuracy. Even though it is very easy to implement FDM in program coding, its prohibitive computational costs (solving governing equations many times) motivates other time-efficient methods to calculate reduced gradients.

The automatic differentiation (AD) method has been developed for the MATLAB, C and FORTRAN languages. In the AD method, a program calculating the derivative information of the user-defined functions with respect to the inputs is generated from the original analysis program. Every operation used in the analysis codes (including grid generator and flow solver) is differentiated based on the chain rule. As stated by Taylor & Oloso in [65], one of its principal advantages is to calculate derivative information to the same precision as the functional calculation. This method has been applied by Green *et al.* [32] to calculate the derivatives of aerodynamic functionals to certain turbulence modelling parameters. Sundaram *et al.* [84] reported efforts to use $ADIFOR$ 3.0 gradients in aircraft shape optimization designs. Taylor & Green [69] expanded AD reverse mode to calculate second-order derivatives of some aerodynamic performance functionals.

Other approaches include various forms of sensitivity methods and adjoint methods. The sensitivity method is powerful when dealing with problems involving many cost/constraint functionals, and with relatively few design parameters, because the sensitivity equations must be solved for each parameter. Applications of sensitivity methods can be found in [9, 63, 89, 38]. In the adjoint method, the adjoint equations are to be solved for each cost/constraint functional, therefore the advantage of this method over sensitivity methods

lies in solving problems with many design parameters, but with relatively few functionals. In the following section a simple example will be taken to display some advantages and disadvantages of these two methods.

1.2 Sensitivity Method vs. Adjoint Method

Consider an optimization design problem with state variables $\mathcal{X} \in X \subset R^n$ and control variables $\mathcal{U} \in U \subset R^m$ where the cost functional $f(\mathcal{X},\mathcal{U}) \in R$ is to be minimized subject to the constraint $g(\mathcal{X},\mathcal{U}) = \mathbf{0} \in \mathcal{R}^n$. Here $g : X \times U \longmapsto R^n$ is a Frèchet differentiable transformation which may be linear or nonlinear. By the implicit function theorem (A.2.1), when $\left(g_{\mathcal{X}}(\mathcal{X},\mathcal{U})\right)^{-1}$ exists, the constraint implicitly defines a map $h : R^m \longmapsto R^n$ such that $\mathcal{X} = h(\mathcal{U})$ solves $g(\mathcal{X},\mathcal{U}) = 0$. Therefore, the gradients of h with respect to the control variable $\mathcal{U}$ can be computed from the sensitivity equation

$$\left[\frac{\partial g}{\partial \mathcal{X}}\right]\left[\frac{\partial h}{\partial \mathcal{U}}\right] + \left[\frac{\partial g}{\partial \mathcal{U}}\right] = \mathbf{0} \tag{1.2.1}$$

to give

$$\frac{\partial h}{\partial \mathcal{U}} = -\left[\frac{\partial g}{\partial \mathcal{X}}\right]^{-1}\frac{\partial g}{\partial \mathcal{U}}.$$

Furthermore, we have the reduced functional $\hat{f}(\mathcal{U}) = f(h(\mathcal{U}),\mathcal{U})$, from which the *reduced* gradient of the cost functional can be obtained

$$\frac{\partial \hat{f}}{\partial \mathcal{U}} = \frac{\partial f}{\partial \mathcal{X}}\frac{\partial h}{\partial \mathcal{U}} + \frac{\partial f}{\partial \mathcal{U}} = -\frac{\partial f}{\partial \mathcal{X}}\left[\frac{\partial g}{\partial \mathcal{X}}\right]^{-1}\frac{\partial g}{\partial \mathcal{U}} + \frac{\partial f}{\partial \mathcal{U}}. \tag{1.2.2}$$

This approach to calculate the reduced gradient is the sensitivity method. Note that no information about the cost functional is involved in the sensitivity equation (1.2.1), therefore the sensitivity ($h_{\mathcal{U}}$) can be applied to calculate the gradient of any cost/constraint functional. However, the sensitivity equation must be solved for each control. In this case, the sensitivity equation (1.2.1) must be solved m times since there are m controls.

The adjoint method is described as follows. First the constraint is adjoined to the cost functional via the Lagrange multiplier $\lambda \in R^m$

$$\bar{f} = f(\mathcal{X},\mathcal{U}) + \lambda^T g(\mathcal{X},\mathcal{U}).$$

The first variation of the augmented cost functional is given by

$$\delta\bar{f} = (\frac{\partial f}{\partial \mathcal{X}} + \lambda^T \frac{\partial g}{\partial \mathcal{X}})\delta\mathcal{X} + (\frac{\partial f}{\partial \mathcal{U}} + \lambda^T \frac{\partial g}{\partial \mathcal{U}})\delta\mathcal{U},$$

the first term of which involves calculating the perturbations in states, and can be circumvented by setting

$$\frac{\partial f}{\partial \mathcal{X}} + \lambda^T \frac{\partial g}{\partial \mathcal{X}} = 0, \tag{1.2.3}$$

or

$$\lambda = -\left(\frac{\partial f}{\partial \mathcal{X}}\left[\frac{\partial g}{\partial \mathcal{X}}\right]^{-1}\right)^T.$$

This is the fundamental adjoint equation. We arrive at the reduced gradient of the cost functional by differentiating $\bar{f}$ with respect to the controls,

$$\frac{\partial \bar{f}}{\partial \mathcal{U}} = \frac{\partial f}{\partial \mathcal{U}} + \lambda^T \frac{\partial g}{\partial \mathcal{U}} = \frac{\partial f}{\partial \mathcal{U}} - \frac{\partial f}{\partial \mathcal{X}}\left[\frac{\partial g}{\partial \mathcal{X}}\right]^{-1}\frac{\partial g}{\partial \mathcal{U}}. \tag{1.2.4}$$

Although the adjoint equation must be solved for each functional involved in the problem, there is no information about the control variable involved in the adjoint equation, which means that the same adjoints can be applied in calculating the gradient of the cost functional with respect to more than one control variable. This observation indicates that the adjoint method will be especially efficient in calculating the gradient information in design problems of many design variables and relatively few functionals, since the computational cost of solving the adjoint equations is independent of the number of design variables. Both the sensitivity equation (1.2.1) and adjoint equation (1.2.3) are linear even when the constraint g is nonlinear. One of the main advantages of these methods is that the computational cost for solving the linear system will be expected to be much less than solving the governing equations $g(\mathcal{X},\mathcal{U}) = 0$ in most cases. As expected, (1.2.2) and (1.2.4) show that both

the sensitivity method and adjoint method arrive at the same reduced gradient of the cost functional with respect to the controls.

While our discussion was motivated by situations where the state vector ($\mathcal{X}$) is finite dimensional ($\mathcal{X} \in R^n$), in many applications of interest it is useful to admit a more general *Hilbert* space setting. In Chapter 3 where a quasi one-dimensional (1-D) nozzle flow will be considered, we utilize the space $L_2[0,1]$. For two-dimensional (2-D) airfoil flows to be considered in Chapter 4, the flow domain is an open set $\mathcal{D}$ in R^2 with the boundary $\partial\mathcal{D}$. Then, we will consider the spaces $L_2(\mathcal{D})$ and $L_2(\partial\mathcal{D})$. Some basic concepts about these Hilbert spaces are included in Appendix A.

1.3 Development of Adjoint Methods

Adjoint methods have witnessed many developments in optimal aerodynamic design in the past few years [47]. This method found its first application in quasi 1-D nozzle flows with a shock. Many authors developed a variety of numerical methods to overcome the difficulties introduced by discontinuity of the flow variables at the shock. In [40], Iollo, *et al.*, derived adjoint equations and corresponding boundary conditions ($B.C.$) for flows through the quasi 1-D convergent-divergent nozzle; and solved the adjoint equations numerically. In their method, the adjoints were set to zero at the shock location. Cliff, Shenoy, *et al.*, [16, 15, 76], explored an inverse design problem of quasi 1-D flow through a divergent nozzle with a supersonic inlet and an embedded shock. In [76] the mass equation and energy equation were integrated such that the quasi 1-D Euler equations were reduced to a single ODE. Optimal control theory was employed to derive the optimality system for the inverse design problem; and the existence and the uniqueness of adjoints were proved. The adjoint equations were solved using shock-fitting methods in transformed coordinates which lead to discontinuities in adjoints at the shock location. Giles and Pierce [27] employed a Green's function approach to construct analytic solutions to the adjoint equations for quasi 1-D nozzle flows, discovering

the properties of adjoints at the sonic throat and the shock location. This approach was also generalized to 2-D airfoil flow [26], however with little success. Jameson *et al.*, extended the adjoint methods and control theory to the design of 2-D and 3-D inviscid flows [44, 25], and viscous flows [37]; the adjoint equations were derived in the transformed computational domain. In [59, 71, 73] numerical methods of solving adjoint equations over constructed grids utilized central difference schemes with artificial dissipation. The adjoints at the far-field were set to zero. In [17], Dadone and Grossman explored the discrete adjoint method and applied it in the progressive optimization strategy. Baysal and Ghayour [8] derived the adjoint equations in Cartesian coordinates by constructing the material derivatives for integrations over the boundary and the domain. However, normal derivatives of flow variables on the airfoil were required in their methods. In [6, 8] the adjoint equations were solved on an unstructured grid system using Roe's schemes. Most recently, Nardarajah *et al.* [60] applied a discrete adjoint method to obtain the geometric sensitivities of pressure distribution at an arbitrary location within supersonic flow over the airfoil. [1]

The adjoint methods developed by all these authors were devoted to the calculation of gradients of the cost/constraint functionals with respect to geometric design parameters. The approaches were limited to shape optimization problems, and could not be utilized for problems with flow-related design parameters. Another open problem is the far-field boundary conditions for adjoints in 2-D and 3-D flows. In previous studies, many authors set the adjoints at the far-field to zero. The motivation of the present research is to broaden the applications of the adjoint method in the optimum aerodynamic design, by investigating the characteristic adjoint boundary conditions and by extending the adjoint method to calculate flow-related gradients for the cost/constraint functionals.

The dissertation is arranged as follows. A general description of some numerical optimization

[1] Although most interests in the adjoint method are focused on its efficiency in calculating functional gradients, Pierce & Giles [66] and Venditti & Darmofal [93] applied this method to the error estimation for a functional output from the governing quasi 1-D Euler equations. This topic is beyond the scope of this dissertation.

techniques is provided in Chapter 2, followed by exploring the pressure-matching inverse design for quasi 1-D nozzle flows in Chapter 3. A numerical method based on Steger-Warming flux-vector-splitting scheme is proposed to solve the adjoint equations numerically; and the properties of adjoints at the sonic throat and the shock location are discussed. Also presented in this chapter is the approach of Green's function to obtain the analytic solution to the adjoint equations in an isentropic transonic nozzle flow. Finally, in this chapter a pressure-matching problem is studied for a quasi 1-D nozzle with the shape parameterized by a single geometric design parameter. In Chapter 4, the adjoint method is applied to 2-D transonic airfoil design, in which the characteristic adjoint boundary conditions are derived, as well as the reduced gradients with respect to the geometric and flow-related design variables. Based on numerical solutions to the adjoint equations, optimization design is performed on a modified $NACA$ 0012 airfoil to find the optimal airfoil shape and the angle of attack to minimize the drag coefficient with a specified lift coefficient at given freestream Mach number. The super-reduced structure of the design problem has also been studied. Two different approaches are proposed to calculate the super-reduced gradients of the drag coefficient subject to the lift constraint. Computational experiments are also conducted exploiting this super-reduced structure. In concluding remarks, two potential research directions are proposed in order to deepen the insights to the adjoint methods.

Chapter 2

Optimization Methods

Each optimization problem we consider includes some type of differential equation as part of the constraint system. Numerical design methods for this class of problems can be usefully grouped into two categories: the approach of optimize-then-discretize and the approach of discretize-then-optimize. In the former approach, the optimality conditions are first derived, then approximation techniques are applied to solve the state/optimality system. In contrast to this method, the discretize-then-optimize approach first discretizes the governing differential equations via finite difference method (FDM), finite volume method (FVM) or finite element method (FEM), then derives the optimality conditions for the discrete governing equations. Continuous and discrete adjoint methods [59] are examples of this categorization.

Optimization algorithms can be classified into three groups: zero-order, first-order and second-order methods, according to the order of gradients of the cost functional used in the optimization method. No information of the gradients of cost/constraint functional is supplied in the zero-order method; searching for the optimal point is undertaken with comparisons of values of cost functionals at different design points. The methods of random search, genetic algorithms and Powell's method [91] fall into this category. This kind of method is easy to code and can deal with problems with discrete design variables. However, the requirement of numerous functional evaluations prohibit the application of zero-order methods in the design problems in which function evaluation requires great computational

cost (such as solving nonlinear $ODEs$ and $PDEs$). In the first-order methods, the gradients of the cost functional with respect to the design variables are employed to determine a search direction. A typical method of this class is the steepest descent method in which the search direction is set to the negative gradient direction.

Second-order methods require a Hessian or second-derivative matrix, in addition to the first order gradients. For smooth problems, first- and second-order methods are more efficient than zero-order methods. However, computing the first-order gradients is already a hard task in most cases where an analytical presentation of the cost functional is not available. Direct computation of the Hessian matrix is even more difficult. As a way out of this dilemma, quasi-Newton methods are developed, in which one uses changes in the gradient to infer information about the Hessian.

In this chapter, the necessary conditions for unconstrained and constrained optimization problems are first stated, followed by a brief description of the quasi-Newton method. Some constrained optimization methods are described in the last section.

2.1 Optimality Conditions

Gill summarized necessary and sufficient optimality conditions for a variety of optimization problems in [29]. Here, only the necessary conditions are stated, both for unconstrained problems to minimize cost functional $f(x), x \in R^n$ and for problems with nonlinear equality constraints $C(x) = (c_1(x), \cdots, c_m(x))^T = \mathbf{0}, m < n$. The gradient vector of the cost functional f is denoted by $\nabla f(x)$ or $g(x)$,

$$\nabla f(x) \equiv g(x) \equiv (\frac{\partial f}{\partial x_1}, \cdots, \frac{\partial f}{\partial x_n})^T.$$

The Hessian matrix of f, denoted by $\nabla^2 f(x)$ or $H(x)$, is given by,

$$\nabla^2 f(x) \equiv H(x) \equiv \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}.$$

The Jacobian of the constraints with respect to the design variables is given by,

$$J(x) = \begin{pmatrix} \frac{\partial c_1}{\partial x_1} & \cdots & \frac{\partial c_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial c_m}{\partial x_1} & \cdots & \frac{\partial c_m}{\partial x_n} \end{pmatrix}.$$

For the unconstrained problem it is easy to show that if x^* is a local minimizer for f, then

- $g(x^*) = \mathbf{0}$, i.e., x^* is a stationary point;
- $H(x^*)$ is positive semi-definite.

For nonlinear equality constrained problems, it is assumed that the constraints satisfy a *constraint qualification* condition. Specifically, we assume that the Jacobian of constraints $J(x^*)$ has full row rank. Let $Z(x)$ denote a matrix whose columns form a basis for the set of vectors orthogonal to the rows of $J(x)$, then if x^* is a local minimizer for the problem,

- x^* is a feasible point, $C(x^*) = \mathbf{0}$,
- $Z(x^*)^T g(x^*) = \mathbf{0} \quad \Leftrightarrow \quad g(x^*) = J^T(x^*)\lambda$
- $Z(x^*)^T W(x^*, \lambda^*) Z(x^*)$ is positive semi-definite,

where W is the Hessian matrix of Lagrangian functional $L(x, \lambda) = f(x) - \lambda^T C(x)$. Here, λ is a Lagrange multiplier, whose existence is required by the second bulleted item. From these conditions, we can conclude that an extremum point x^* is a stationary point for the Lagrangian functional. Note that x^* does not generally provide a local minimizer for the Lagrangian functional in the sense that any R^n space around x^* will contain points $\hat{x}$ with

$L(\hat{x}, \lambda) < L(x^*, \lambda)$. However, the third bulleted item does guarantee that x^* is a local minimizer for L in the nullspace of the constraints. This observation plays an important role in the development of numerical methods for such constrained optimization problems.

2.2 Quasi-Newton Method

The necessary condition for unconstrained optimization problems is the gradient of cost function g vanishing at a local minimizing point. Newton's method can be used to solve the equations $g = \mathbf{0}$, leading to

$$H_k p_k = -g_k, \tag{2.2.1}$$

where H_k and g_k are the Hessian matrix and the gradient vector at the current iteration, respectively; and p_k is the Newton direction. In most optimization problems, even though calculating the Hessian matrix can be a difficult task, it is possible to approximate it using observed behavior of the cost functional and its gradients. When the Hessian matrix H_k is replaced by its approximation B_k, Eq. (2.2.1) takes the form

$$B_k p_k = -g_k. \tag{2.2.2}$$

The search direction p_k defined by this equation is called the *quasi-Newton direction*. In the simplest case, Eq. (2.2.2) is solved for p_k, and a one-dimensional search is performed to find the step α,

$$\underset{\alpha > 0}{min} \quad f(x_k + \alpha p_k)$$

to determine how far to go in the direction of p_k. In the first iteration, the approximation for Hessian is taken to be the identity or some multiple of it; in this case Eq. (2.2.2) defines the steepest descent direction. After the next point x_{k+1} is obtained, the approximate Hessian matrix B_k should be updated to B_{k+1} to take account of newly-acquired curvature information in the form,

$$B_{k+1} = B_k + U_k,$$

with U_k being the update matrix. The Taylor's expansion of the gradient function about x_k is

$$g(x_k + \Delta x_k) = g_k + H_x \Delta x_k + \cdots ,$$

where g_k is understood to be $g(x_k)$. In this setting we require that the new Hessian approximation satisfy,

$$B_{k+1}\Delta x_k = y_k \tag{2.2.3}$$

where $y_k = g_{k+1} - g_k$. Condition (2.2.3) is called the *quasi-Newton condition.* In what follows, two efficient updates for the approximate Hessian matrix are briefly described.

2.2.1 Rank-one update

In this updating method, the update matrix U_k is set to be a rank-one matrix, i.e., $U_k = uv^T$ for some column vectors u, v. Replacing B_{k+1} by $B_k + uv^T$ in (2.2.3) yields,

$$u(v^T \Delta x_k) = y_k - B_k \Delta x_k.$$

Under the assumption that $v^T \Delta x_k$ is non-zero, the vector u can be obtained by dividing the above formulation through by it since it is just a scalar, $u = \frac{y_k - B_k \Delta x_k}{v^T \Delta x_k}$, and therefore B_{k+1} is given by,

$$B_{k+1} = B_k + \frac{1}{v^T \Delta x_k}(y_k - B_k \Delta x_k)v^T.$$

Since the Hessian matrix is symmetric, its approximation should also be symmetric; therefore the vector u can be chosen to be $v = y_k - B_k \Delta x_k$. Thus, the *symmetric rank-one update* is given by

$$B_{k+1} = B_k + \frac{1}{(y_k - B_k \Delta x_k)^T \Delta x_k}(y_k - B_k \Delta x_k)(y_k - B_k \Delta x_k)^T.$$

This update possesses the property of hereditary symmetry, i.e., B_{k+1} is also symmetric if B_k is symmetric. Various rank-two updates can be constructed based on the rank-one update.

2.2.2 Rank-two updates

While there are many rank-two update methods, only two are given in this section. Readers are referred to [29] and [30] for more details.

In the *David-Fletcher-Powell (DFP)* update, the Hessian is modified by,

$$B_{k+1} = B_k - \frac{1}{\Delta x_k^T B_k \Delta x_k} B_k \Delta x_k \Delta x_k^T B_k + \frac{1}{y_k^T \Delta x_k} y_k y_k^T + (\Delta x_k^T B_k \Delta x_k) w_k w_k^T,$$

with

$$w_k = \frac{1}{y_k^T \Delta x_k} y_k - \frac{1}{\Delta x_k^T B_k \Delta x_k} B_k \Delta x_k.$$

Note that $w_k^T \Delta x_k = 0$, and the addition of any multiple of rank-one matrix $w_k w_k^T$ to B_{k+1} will not violate the quasi-Newton condition (2.2.3) or the symmetric requirement. This observation gives the one-parameter Broyden family of updates

$$B_{k+1}^{\phi} = B_k - \frac{1}{\Delta x_k^T B_k \Delta x_k} B_k \Delta x_k \Delta x_k^T B_k + \frac{1}{y_k^T \Delta x_k} y_k y_k^T + \phi_k (\Delta x_k^T B_k \Delta x_k) w_k w_k^T, \quad (2.2.4)$$

with scalar ϕ_k taking value between 0 and 1. The resulting update with $\phi_k = 0$ is called the *Broyden-Fletcher-Goldfarb-Shanno (BFGS) update*. It can be verified that the *BFGS* update can maintain the positiveness of the approximate Hessian if and only if $y_k^T \Delta x_k > 0$ [29]. Broyden [10], Fletcher [23], Goldfarb[31] and Shanno [75] also constructed approximations to the inverse Hessian.

2.3 Trust-Region Method

In the iterations of unconstrained optimization, a quadratic model is usually constructed to approximate the nonlinear cost functional using the information of first-order gradient and the Hessian matrix of the cost functional. It may happen that minimization of the quadratic model requires a 'large' step. In this case there is a concern that the quadratic model may not be a sufficiently accurate approximation of the cost functional. While there are several

ways of controlling this behavior, one popular approach is to minimize the quadratic model subject to an explicit bound on the design step size, that is,

$$\begin{aligned} \min_{s} \quad & m_c(x_c + s) = f(x_c) + \nabla f(x_c)^T s + \frac{1}{2} s^T B_c s \\ s.t. \quad & \|s\| \leq r_c \end{aligned} \tag{2.3.5}$$

where B_c is an approximation to the Hessian matrix at the current point. This means the quadratic model is trusted to be a good approximation to the cost functional in a region defined by r_c. As pointed in [19], the solution to the above constrained optimization problem can be given by

$$s(\mu) = -(B_c + \mu I)^{-1} \nabla f(x_c), \tag{2.3.6}$$

where μ is chosen such that $\|s(\mu)\| = r_c$ if $s(0) > r_c$. The solution (2.3.6) defines a curve linking the current point x_c ($\mu \to \infty$) with the Newton point $x_+^N = x_c - B_c^{-1} \nabla f(x_c)$ ($\mu = 0$). In Powell's approach of a double dogleg step, this curve is approximated by a polygonal line, which connects the current point x_c, the *Cauchy point C.P.*, the point N and the Newton point x_+^N. (The Cauchy point is the minimizing point along the steepest descent direction. It can be proved that the Newton point is further away from x_c than the *C.P.*). The point N is chosen as

$$N = x_c - \eta B_c^{-1} \nabla f(x_c),$$

where $\eta \leq 1$ and such that $||N - x_c|| \geq ||C.P. - x_c||$. Fig. 2.1 shows the *double dogleg* approximation. A interesting property of this double dogleg step is that the quadratic model function is monotonically decreasing along it. So the solution to the above-mentioned optimization problem can be easily obtained.

2.4 Constrained Optimization Method

The book [30] edited by Gill & Murray is a good reference on the constrained optimization methods. Here only a very brief review will be made about two Lagrangian-based methods.

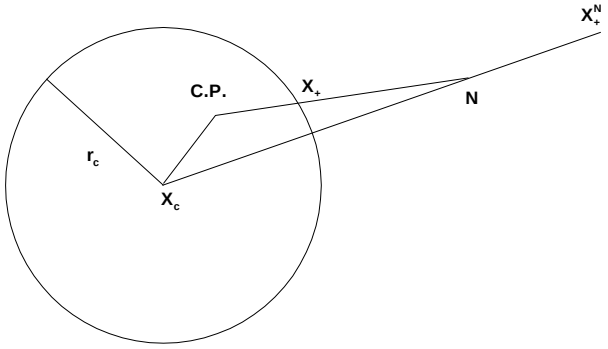

Figure 2.1: The double dogleg curve, $x_c \rightarrow C.P. \rightarrow N \rightarrow x_+^N$

A local idea to solve constrained optimization problems is to construct an unconstrained problem whose solution is related to the solution of the constrained problem in a known way so that a successive unconstrained problems can be solved to obtain the solution to the original constrained problem. In Vanderplaats' [91] and Gill's [29] textbook, the methods of using penalty and barrier functions are discussed. This kind of method is easy to be performed; however, it is not efficient. Popular methods for constrained problems are based on the local minimization property of the Lagrangian (see Section 2.1). Two of them are briefly introduced here: the augmented Lagrangian method and the projected Lagrangian method.

2.4.1 Augmented Lagrangian method

Theorem (A.3.1) states that the local minimizer (x^*) of a constrained optimization problem is a stationary point of the Lagrangian functional. Note that x^* is not necessarily a minimizer for the Lagrangian functional. Gill [29] points out this point is the minimum point of the Lagrangian functional within the nullspace of active constraints. In the augmented Lagrangian method, a term is added to the Lagrangian functional such that the local extremum is not only the stationary point of the augmented Lagrangian functional, but also a local minimizer. This additional term can be constructed by a penalty on the violation of

the constraints,

$$L_A(x, \lambda, \rho) = f(x) - \lambda^T C(x) + \frac{\rho}{2} C(x)^T C(x),$$

where the penalty parameter ρ is positive. It can be shown that there is a finite value ρ^* such that for $\rho > \rho^*$, the local extremum of the constrained problem will be a local minimizer of the augmented Lagrangian functional. So the constrained optimization problem is transformed to an unconstrained problem of minimizing the augmented Lagrangian functional.

2.4.2 Projected Lagrangian method

In each iteration of this method, the active constraints are linearized at the current point; and an objective functional L_c is to be minimized subject to these linear constraints. A possible choice for L_c is,

$$L_c = f(x) - \lambda^T C(x) + \lambda^T A x$$

where A is the Jacobian matrix of active constraints with respect to the design variables. A burden with this method and the above augmented Lagrangian method is the need to estimate the Lagrange multipliers at each iteration. Moreover, the set for the active constraints may change for the case with inequality constraints.

Another choice for the objective functional is a quadratic approximation to the augmented functional. In this case the method is called *sequential quadratic programming* (SQP) method and the subproblem takes the following form,

$$\begin{aligned} \min_p \quad & g_k^T p + \frac{1}{2} p^T H_k p \\ s.t. \quad & A_k p = -c_k \end{aligned}$$

where g_k is the gradient of the cost functional, H_k is the Hessian of the Lagrangian, A_k is the Jacobian of the active constraint and c_k is the value of constraint functions, all evaluated at the current point.

The trust-region method has also been expanded to the constrained optimization problem. Readers are referred to Dennis & Schnabel [19], Vardi [92], and more recent Yuan [94], Dennis, El-alem & Maciel [48] and Dennis, Heinkenschloss & Vicente [41].

2.5 Constrained Optimum Aerodynamic Design

The design problems considered here combine modern computational fluid dynamics and optimization theory in the study of several optimum aerodynamic design problems. In this setting, as in many others, it is useful to partition the variables used in the problem description into two groups, the *state* vector and the *control* vector, as described in Section A.3. The state/control terminology is borrowed from control theory. For all of our applications the state spaces are infinite dimensional Hilbert spaces, while the design spaces are finite dimensional. The final ingredient in an optimization design problem is a real-valued cost functional. By functional, we mean a map from the vector space into the space of real scalars.

In Chapter 3, the inverse design problem is explored for pressure-matching. Contrary to approaches by many other authors, in which the flow variables (either in conservative form or in non-conservative form) are viewed as states, the flux variables are treated as states in the present dissertation.

In Chapter 4, we focus on minimizing the drag coefficient C_D on an airfoil, subject to the governing 2-D Euler equations (transonic) and an aerodynamic constraint, $C_L - C_{L_0} = 0$. We take the flow variables Q as the states; while the controls include geometric design parameters $\mathcal{G}$ and the angle of attack α. Thus, the design problem can be put in a mathematical form

$$\begin{aligned} \min_{Q,\,\mathcal{G},\,\alpha} \quad & C_D(Q;\mathcal{G},\alpha) \\ s.t. \quad & r(Q;\mathcal{G},\alpha) = \mathbf{0} \\ & C_L(Q;\mathcal{G},\alpha) - C_{L_0} = 0, \end{aligned} \tag{2.5.7}$$

where the nonlinear operator r denotes the Euler equations in the flow domain and the boundary conditions at the boundaries.

The problem is treated in a black-box approach, in the sense that the governing Euler equations are not 'seen' by the optimizer[1]. The Euler equations (with appropriate boundary conditions) manifest themselves in calculating the state variables for given design variables $\mathcal{G}, \alpha$; this is justified by the implicit function theorem (A.2.1) discussed in Appendix A. In other words, we can solve the Euler equations $r(Q; \mathcal{G}, \alpha) = \mathbf{0}$ to find $Q = Q(\mathcal{G}, \alpha)$. Substituting for the states in the above optimization problem (2.5.7), we obtain the *reduced* problem

$$\begin{aligned} \underset{\mathcal{G},\,\alpha}{min} \quad & C_D(Q(\mathcal{G},\alpha);\mathcal{G},\alpha) \\ s.t. \quad & C_L(Q(\mathcal{G},\alpha);\mathcal{G},\alpha) - C_{L_0} = 0. \end{aligned} \tag{2.5.8}$$

When a gradient-based optimizer is employed to solve the problem (2.5.8), the *reduced* gradients, $\frac{\partial C_D}{\partial \mathcal{G}}$, $\frac{\partial C_D}{\partial \alpha}$, $\frac{\partial C_L}{\partial \mathcal{G}}$, and $\frac{\partial C_L}{\partial \alpha}$ must be provided. These gradients can be obtained via adjoint methods developed in the following chapters. The reduced problem can be solved by constrained optimization methods described above. Optimization problems of this reduced form will be studied in Chapter 4.

Since the only remaining aerodynamic constraint is a scalar equation, the reduced problem can be put in another *super-reduced* form by solving the aerodynamic constraint on C_L to obtain $\alpha = \alpha(\mathcal{G})$. In other words, the angle of attack α is not treated as a control, instead, it is viewed as a state, and we get the *super-reduced* problem,

$$\underset{\mathcal{G}}{min} \quad C_D\big(Q(\mathcal{G},\alpha(\mathcal{G}));\mathcal{G},\alpha(\mathcal{G})\big). \tag{2.5.9}$$

Obviously, the *super-reduced gradient* $\frac{\partial C_D}{\partial \mathcal{G}}$ to be calculated is different from the geometric reduced gradient to the reduced problem (2.5.8); it is the geometric gradient of the cost functional subject to the lift constraint. In this way, the constrained optimization problem

[1]In contrast to this approach, the all-at-once approaches directly work with discrete Euler equations as constraints in the optimizer.

(2.5.8) is reduced to a problem without the nonlinear lift constraint. One tradeoff for this type of reduction is that the resulting functional is often more nonlinear than the original cost functional. A second issue is the work required to compute the super-reduced gradients of the cost functional.

In optimal airfoil design, other cost functionals may be of interest. Some choices utilize combinations of C_D and C_L to replace the constrained optimization problem with a form of unconstrained problem. One choice is to minimize C_D/C_L^n. From an aerodynamic point of view, the situation when $n = 1$ corresponds minimizing the level-flight drag and is of interest in maximizing the gliding distance for power-off flight. The case $n = \frac{3}{2}$ corresponds to minimum power required and is of interest in minimizing the rate of descent (maximum endurance [7]). Another choice is to minimize $C_D - \sigma C_L$, where constant σ is an estimate to the Lagrange multiplier. This idea comes from applying the Lagrangian Multiplier Theorem to (2.5.8) to obtain the optimal necessary conditions that the Lagrangian function $C_D - \sigma(C_L - C_{L_0})$ is stationary at optimum. In this choice the optimization results are dependent on the estimate of the Lagrange multiplier. The studies of these choices for the cost functional are not considered here.

Chapter 3

Adjoint Methods for Quasi 1-D Nozzle Flows

It is reported in [24] that the inviscid transonic flow in the vicinity of the upper surface of the airfoil qualitatively resembles transonic quasi 1-D nozzle flow. Therefore, the examination of the adjoint method applied to quasi 1-D nozzle flows may be helpful in understanding the 2-D adjoint results.

In this chapter, a pressure-matching problem is treated as an optimal control problem with a geometric parameter, in which the governing Euler equations behave as the dynamic model. The design parameter is considered as an augmented state such that the design problem is reduced to a standard optimal control problem. Pontryagin's Minimum Principle (PMP) for this structure, which is stated in Appendix A, is applied to derive the adjoint equations, the adjoint boundary conditions and the reduced gradient for isentropic subsonic nozzle flows. For isentropic transonic flow, a variational approach is employed to investigate the adjoint boundary condition at the supersonic outlet. For shocked flow, we apply the Lagrange Multiplier Theorem described in Appendix A to derive the adjoint system. In Section 3.2, the analytical adjoint solution for the isentropic transonic nozzle flow is also obtained via Green's function approach. The properties of adjoints at the shock and the sonic throat are investigated to show that the adjoints are continuous at the shock and that there is a singularity in the adjoints at the sonic throat. A shock-capturing scheme, based on the flux-

vector-splitting method, is proposed in Section 3.4 to solve the adjoint equations numerically. Good agreement is achieved between the numerical solution and analytical solution. The adjoint singularity at the sonic throat can be better resolved by adopting a nonuniform grid system with grid points concentrated toward the throat; in this case, the recursive projection method (RPM) is employed to improve the convergence properties as described in Appendix B. Finally in Section 3.5, the numerical scheme of solving the adjoint methods is combined with an optimizer to conduct a pressure-matching inverse design on the shocked nozzle flow.

3.1 Adjoint Equations and Boundary Conditions

The steady flow of an inviscid fluid in a quasi 1-D nozzle with variable cross sectional area $A(x)$ is governed by the Euler equations

$$\mathcal{F}_x = -\mathcal{G}, \quad 0 \leq x \leq 1, \tag{3.1.1}$$

where

$$\mathcal{F} = \left\{ \begin{array}{c} \rho u \\ \rho u^2 + p \\ \rho u H \end{array} \right\}, \mathcal{G} = \frac{A_x}{A} \left\{ \begin{array}{c} \rho u \\ \rho u^2 \\ \rho u H \end{array} \right\}.$$

Standard notation has been used, with ρ being the fluid density, u the velocity, H the stagnation enthalpy, and p is the fluid static pressure. The subscript x denotes differentiation with respect to the position along the streamwise direction x. It is also assumed that the cross sectional area of the duct is described in a parameterized way as $A(x;\alpha)$, with α being a geometric design variable. Given a desired pressure distribution $p^*(x),\ x \in [0,1]$, suppose that we are to determine a value of the area-design parameter α to minimize

$$I(\alpha) = \frac{1}{2} \int_0^1 [p(x;\alpha) - p^*(x)]^2 dx, \tag{3.1.2}$$

where $p(x;\alpha)$ denotes the pressure distribution as the flow solution of the 1-D Euler equation with appropriate boundary conditions.

In what follows, the PMP (see Appendix A) will be applied to derive the adjoint equations and the reduced geometric gradient of the cost functional (3.1.2) for the isentropic subsonic flow. For isentropic transonic flow, the variational approach is adopted to investigate the adjoint boundary conditions at the supersonic outlet. For shocked flow, the steady Rankine-Hugoniot conditions are adjoined to the cost functional to investigate the adjoint properties at the shock location.

3.1.1 Isentropic subsonic flow

First let us consider isentropic subsonic nozzle flows in which the boundary conditions are total pressure p_0 and total enthalpy H given at the inlet, while static pressure p is given at the outlet.

Introduce the variational Hamiltonian by adjoining the RHS of the dynamic model $\mathcal{F}_x = -\mathcal{G}$ to the Lagrange cost function via adjoints $\lambda = [\lambda_1(x), \lambda_2(x), \lambda_3(x)]^T$,

$$\mathcal{H} \;=\; \frac{1}{2}(p - p^*)^2 - \lambda^T \mathcal{G}.$$

By Pontryagin's Minimum Principle for the parameter optimization described in Appendix A, we have the $ODEs$ governing the adjoints,

$$\lambda_x = -(\frac{\partial \mathcal{H}}{\partial \mathcal{F}})^T, \tag{3.1.3}$$

with initial transversality condition

$$\lambda(0) = \mu_1(\frac{\partial p_0}{\partial \mathcal{F}})^T \mid_{x=0} + \mu_2(\frac{\partial H}{\partial \mathcal{F}})^T|_{x=0}, \tag{3.1.4}$$

and terminal transversality condition

$$\lambda(1) = \nu(\frac{\partial p}{\partial \mathcal{F}})^T|_{x=1}, \tag{3.1.5}$$

where μ_1, μ_2 and ν are scalars. The reduced gradient of cost functional with respect to design

parameter α is given by [1]

$$\frac{\partial I}{\partial \alpha} = \int_0^1 \frac{\partial \mathcal{H}}{\partial \alpha} dx = -\int_0^1 \lambda^T \frac{\partial \mathcal{G}}{\partial \alpha} dx. \tag{3.1.6}$$

Let U denote the conservative variables $[\rho, \rho u, \rho E]^T$, then by the chain rule we have $\frac{\partial(\cdot)}{\partial \mathcal{F}} = \frac{\partial(\cdot)}{\partial U}\frac{\partial U}{\partial \mathcal{F}}$, where the Jacobian $\frac{\partial \mathcal{F}}{\partial U}$ is nonsingular if Mach number $M \neq 1$. Hence, adjoint equations (3.1.3) can be written as

$$-B^T \frac{d\lambda}{dx} + S^T \lambda - (p - p^*)(\frac{\partial p}{\partial U})^T = 0, \tag{3.1.7}$$

where the Jacobian matrices of flux vectors and the source term B, S are respectively given by

$$\begin{aligned} B(U) = \frac{\partial \mathcal{F}}{\partial U} &= \begin{pmatrix} 0 & 1 & 0 \\ \frac{1}{2}(\gamma-3)u^2 & (3-\gamma)u & (\gamma-1) \\ (\gamma-1)u^3 - \gamma u E & \gamma E - \frac{3(\gamma-1)}{2}u^2 & \gamma u \end{pmatrix}, \\ S(U) = \frac{\partial \mathcal{G}}{\partial U} &= \frac{A_x}{A}\begin{pmatrix} 0 & 1 & 0 \\ -u^2 & 2u & 0 \\ (\gamma-1)u^3 - \gamma u E & \gamma E - \frac{3(\gamma-1)}{2}u^2 & \gamma u \end{pmatrix}, \end{aligned} \tag{3.1.8}$$

with γ being the ratio of the specific heats and E the stagnation internal energy. The adjoint equations (3.1.7) with boundary conditions (3.1.4) at the inlet and (3.1.5) at the outlet constitute a two point boundary value problem (BVP).

3.1.2 Isentropic transonic flow

In isentropic transonic flow, the stagnation enthalpy and stagnation pressure should be specified at the subsonic inlet; the Mach number at the throat should be unity [28]. At the supersonic outlet, no boundary condition can be imposed on the flow variables, indicating that the set Θ^1 of terminal points is anything in R^3. Thus, $\lambda(1) \perp \Theta^1$ implies $\lambda(1) = \mathbf{0} \in R^3$.

[1] In PMP, this gradient is set to zero, as a necessary condition for the optimality.

While one can construct a reduced gradient using the Minimum Principle, here we illustrate the use of the variational approach introduced by Iollo *et al.* [40]. First the Lagrangian functional is constructed,

$$\begin{aligned}\mathcal{L} &= \int_0^1 \frac{1}{2}(p-p^*)^2 dx - \int_0^1 \lambda^T (\frac{d\mathcal{F}}{dx} + \mathcal{G}) dx \\ &= \int_0^1 \frac{1}{2}(p-p^*)^2 dx - \int_0^1 (-\frac{d\lambda^T}{dx}\mathcal{F} + \lambda^T \mathcal{G}) dx - [\lambda^T \mathcal{F}]_0^1.\end{aligned}$$

where integration by parts and Green's theorem have been applied. The first variation of $\mathcal{L}$ can be obtained due to the perturbations in flow variables, δU, and in the geometric design parameter, $\delta\alpha$,

$$\begin{aligned}\delta\mathcal{L} &= \int_0^1 (p-p^*)\frac{\partial p}{\partial U}\delta U dx \\ &- \int_0^1 (-\frac{d\lambda^T}{dx}\frac{\partial \mathcal{F}}{\partial U}\delta U + \lambda^T \frac{\partial \mathcal{G}}{\partial U}\delta U) dx - [\lambda^T \frac{\partial \mathcal{F}}{\partial U}\delta U]_0^1 \\ &- \int_0^1 \lambda^T \frac{\partial \mathcal{G}}{\partial \alpha}\delta\alpha dx.\end{aligned}$$

The key idea in this approach is to eliminate the terms involving the perturbations of flow variables. In this way, we obtain the local boundary conditions for adjoints at the inlet and the outlet

$$\lambda^T \frac{\partial \mathcal{F}}{\partial U}\delta U|_0 = \lambda^T \frac{\partial \mathcal{F}}{\partial U}\delta U|_1 = 0.$$

Since at the outlet ($x = 1$), no flow boundary conditions can be imposed for the supersonic flow variables, we should set $\lambda = \mathbf{0}$, identically. Furthermore, the boundary condition at the inlet can be reduced to the form of (3.1.4); and the adjoint equations and the reduced gradient obtained in this way are of the same form as (3.1.3) and (3.1.6), respectively.

It can be observed that when there are two boundary conditions for the flow variables at a location, we will get one adjoint boundary condition, and vice versa. For supersonic flow at the outlet, no boundary conditions can be imposed on the flow variables; on the contrary, we should impose three boundary conditions on the adjoints. In other words, the sum of the

number of flow boundary conditions plus the number of the adjoint boundary conditions is three.

Numerical implementations commonly seek solutions of the steady flow equations (3.1.1) and steady adjoint equations (3.1.7) as limits of corresponding unsteady problems. The complementary nature of the boundary conditions for adjoint variables and flow variables indicates the reverse characteristic model[2] of adjoints as compared to that of flow variables. This inverse characteristic will manifest itself in developing a numerical method to solve the adjoint equations.

3.1.3 Shocked flow

In the above analysis for the isentropic subsonic/transonic flows, the flow variables vary smoothly through the nozzle. To explicitly examine the properties of adjoints in the presence of a shock, this flow discontinuity is considered in a shock-fitting approach.

The governing equations for the flow are given in (3.1.1) almost everywhere in the flow region except at the shock location, x_s. At this flow discontinuity, the steady Rankine-Hugoniot condition should be imposed as[3]

$$\mathcal{F}|_{x_s^-} - \mathcal{F}|_{x_s^+} = \mathbf{0}. \tag{3.1.9}$$

The boundary conditions at the nozzle inlet and outlet are then denoted by operators

$$\mathcal{C}_0 = \mathbf{0}, \tag{3.1.10}$$

and

$$\mathcal{C}_1 = \mathbf{0}, \tag{3.1.11}$$

[2]The unsteady adjoint equations and the unsteady Euler equations are both hyperbolic $PDEs$ of three characteristics.

[3]This condition admits two solutions, in one $U_{x_s^-} = U_{x_s^+}$, while the other corresponds to a normal shock. Here we need to take the second solution.

for simplicity, respectively. The flow region is divided into two segments $[0, x_s^-)$ and $(x_s^+, 1]$ on each of which the states are differentiable. The governing equation (3.1.1), the Rankine-Hugoniot condition (3.1.9) and the boundary conditions (3.1.10) and (3.1.11) are then adjoined to the cost functional (3.1.2) via a set of Lagrange multipliers to form the Lagrangian functional

$$\begin{aligned}
\mathcal{L} &= \int_0^{x_s^-} \big(\frac{1}{2}(p-p^*)^2 - \lambda^T(\frac{d\mathcal{F}}{dx} + \mathcal{G})\big)dx \\
&+ \int_{x_s^+}^1 \big(\frac{1}{2}(p-p^*)^2 - \lambda^T(\frac{d\mathcal{F}}{dx} + \mathcal{G})\big)dx \\
&+ \mu^T\mathcal{C}_0 + \nu^T\mathcal{C}_1 \\
&+ \sigma^T(\mathcal{F}|_{x_s^-} - \mathcal{F}|_{x_s^+}),
\end{aligned}$$

where the dimensions of μ and ν are determined by the number of flow boundary conditions at the inlet and the outlet. After standard mathematical manipulations, the above Lagrangian functional can be rewritten as

$$\begin{aligned}
\mathcal{L} &= \int_0^{x_s^-} \big(\frac{1}{2}(p-p^*)^2 + \frac{d\lambda^T}{dx}\mathcal{F} - \lambda^T\mathcal{G}\big)dx \\
&+ \int_{x_s^+}^1 \big(\frac{1}{2}(p-p^*)^2 + \frac{d\lambda^T}{dx}F - \lambda^T\mathcal{G}\big)dx \\
&+ (-\lambda^T(x_s^-) + \sigma^T)\mathcal{F}|_{x_s^-} - (-\lambda^T(x_s^+) + \sigma^T)\mathcal{F}|_{x_s^+} \\
&+ (\mu^T\mathcal{C}_0 + \lambda^T\mathcal{F}|_0) + (\nu^T\mathcal{C}_1 - \lambda^T\mathcal{F}|_1). \qquad (3.1.12)
\end{aligned}$$

Before we utilize the Lagrange Multiplier Theorem to derive the adjoint system and the reduced gradient, it should be noted that we take $\mathcal{F}$ as the states and the control is associated with the RHS of the governing equations, $\mathcal{G}$. Another point that can not be neglected is that the location of the shock is not fixed; it is dependent on the states $\mathcal{F}$. With these considerations, we can readily obtain the adjoint equations through differentiating the Lagrangian

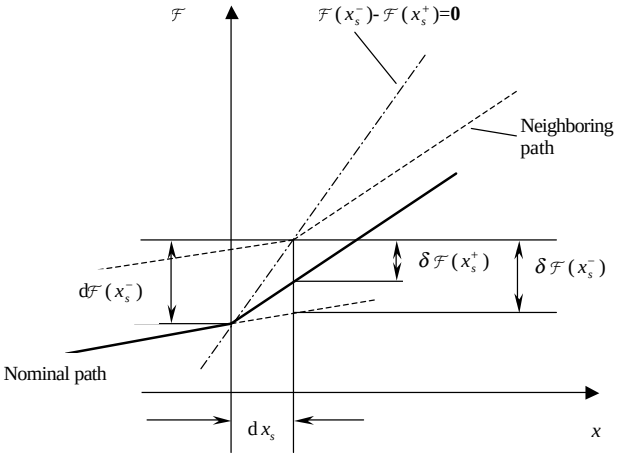

Figure 3.1: Relationships among $d\mathcal{F}$, $\delta\mathcal{F}$ and dx

functional with respect to states,

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \mathcal{F}} &= \int_0^{x_s^-} \big((p-p^*)\frac{\partial p}{\partial \mathcal{F}} + \frac{d\lambda^T}{dx} - \lambda^T \frac{\partial \mathcal{G}}{\partial \mathcal{F}}\big)dx + \big(\frac{1}{2}(p-p^*)^2 + \frac{d\lambda^T}{dx}\mathcal{F} - \lambda^T\mathcal{G}\big)|_{x_s^-}\frac{\partial x_s^-}{\partial \mathcal{F}} \\
&+ \int_{x_s^+}^1 \big((p-p^*)\frac{\partial p}{\partial \mathcal{F}} + \frac{d\lambda^T}{dx} - \lambda^T \frac{\partial \mathcal{G}}{\partial \mathcal{F}}\big)dx - \big(\frac{1}{2}(p-p^*)^2 + \frac{d\lambda^T}{dx}\mathcal{F} - \lambda^T\mathcal{G}\big)|_{x_s^+}\frac{\partial x_s^+}{\partial \mathcal{F}} \\
&+ (-\lambda^T(x_s^-) + \sigma^T) + \big(-\frac{d\lambda^T}{dx}|_{x_s^-}\mathcal{F} + (-\lambda^T(x_s^-)+\sigma^T)\frac{d\mathcal{F}}{dx}\big)\frac{\partial x_s^-}{\partial \mathcal{F}} \\
&- (-\lambda^T(x_s^+) + \sigma^T) - \big(-\frac{d\lambda^T}{dx}|_{x_s^+}\mathcal{F} + (-\lambda^T(x_s^+)+\sigma^T)\frac{d\mathcal{F}}{dx}\big)\frac{\partial x_s^+}{\partial \mathcal{F}} \\
&+ \big(\mu^T\frac{\partial \mathcal{C}_0}{\partial \mathcal{F}} + \lambda^T(0)\big) + \big(\nu^T\frac{\partial \mathcal{C}_1}{\partial \mathcal{F}} - \lambda^T(1)\big).
\end{aligned}$$

Here Leibnitz rule for differentiating an integral with variable integration region is utilized. When differentiating $(\lambda^T(x_s^-)+\sigma)\mathcal{F}|_{x_s^-}$ and $(\lambda^T(x_s^+)+\sigma)\mathcal{F}|_{x_s^+}$ with respect to states $\mathcal{F}$, we use the relationships [11] among the differential $d\mathcal{F}$, the variations $\delta\mathcal{F}$ and dx,

$$\begin{aligned}
d\mathcal{F}(x_s^-) &= \delta\mathcal{F}(x_s^-) + \dot{\mathcal{F}}(x_s^-)dx_s \\
d\mathcal{F}(x_s^+) &= \delta\mathcal{F}(x_s^+) + \dot{\mathcal{F}}(x_s^+)dx_s
\end{aligned}$$

as displayed in Fig. 3.1, where $\dot{\mathcal{F}} = \frac{d\mathcal{F}}{dx}$ since x is the time-like variable.

After regrouping terms, we get,

$$\frac{\partial \mathcal{L}}{\partial \mathcal{F}} = \int_0^{x_s^-} \big((p-p^*)\frac{\partial p}{\partial \mathcal{F}} + \frac{d\lambda^T}{dx} - \lambda^T \frac{\partial \mathcal{G}}{\partial \mathcal{F}}\big)dx \tag{3.1.13}$$

$$+ \int_{x_s^+}^1 \big((p-p^*)\frac{\partial p}{\partial \mathcal{F}} + \frac{d\lambda^T}{dx} - \lambda^T \frac{\partial \mathcal{G}}{\partial \mathcal{F}}\big)dx \tag{3.1.14}$$

$$+ \big(\frac{1}{2}(p-p^*)^2 - \sigma^T \mathcal{G}\big)|_{x_s^-} \frac{\partial x_s^-}{\partial \mathcal{F}}$$

$$- \big(\frac{1}{2}(p-p^*)^2 - \sigma^T \mathcal{G}\big)|_{x_s^+} \frac{\partial x_s^+}{\partial \mathcal{F}}$$

$$+ (-\lambda^T(x_s^-) + \sigma^T) - (-\lambda^T(x_s^+) + \sigma^T) \tag{3.1.15}$$

$$+ \big(\mu^T \frac{\partial \mathcal{C}_0}{\partial \mathcal{F}} + \lambda^T(0)\big) + \big(\nu^T \frac{\partial \mathcal{C}_1}{\partial \mathcal{F}} - \lambda^T(1)\big). \tag{3.1.16}$$

Then each term in the above formulation is set to zero to get the adjoint system (adjoint equations and boundary conditions). We then obtain the adjoint equation from (3.1.13) and (3.1.14),

$$-\frac{d\lambda^T}{dx} + \lambda^T \frac{\partial \mathcal{G}}{\partial \mathcal{F}} - (p-p^*)\frac{\partial p}{\partial \mathcal{F}} = 0$$

almost everywhere in the flow region except at the shock location. These equations can be shown to be identical to the adjoint equations derived for the isentropic flows. From (3.1.15), we get

$$\lambda(x_s^-) = \lambda(x_s^+),$$

which indicates the adjoints are continuous at the shock location. There is some variability in the reported literature on this point. In [40], Iollo *et al.* over-constrained the problem by forcing the adjoint to vanish at the shock. This over-constraint has been pointed out by Giles & Pierce in [27, 28]. However, Giles & Pierce also found an over-specified condition for adjoints at shock through neglecting the fact that perturbations in $\mathcal{F}$ are identical on both sides of the shock because of the Rankine-Hugoniot condition.

Contrary to the continuity of adjoints at the shock, the sensitivities of the flow variables with respect to the design parameters might be discontinuous at the shock as displayed by a computational experiment for a 3-D analytic forebody in supersonic flow in [38].

In equality constrained optimization, the Lagrange multiplier or adjoint is a bounded linear functional on the range of the constraints. In the present setting, this range is an $L_2[0,1]$ space with the inner product defined as an integral (see Appendix A) in the whole flow domain. In [16, 15, 76], Cliff *et al.* decomposed the integration into two parts, one integrating from the inlet to the shock location, the other from the outlet to the shock location with a switch in the integration direction. This operation equivalently switched the sign of the adjoints in that flow region. That is why in their approaches the adjoints had a sign switch across the shock.

In this section, different approaches have been applied to derive the adjoint equations and boundary conditions for different quasi 1-D nozzle flows. All of them lead to the same mathematical formulations for the adjoint equations, since these methods bear the same essence.

3.2 Analytical Adjoint Solutions for Isentropic Transonic Flow

Giles and Pierce's method [27, 28] to obtain the analytic adjoint solution via Green's function approach is introduced here for the purpose of comparisons between numerical results and analytical solutions.

In their method, the cost functional

$$I = \int_{-1}^{1} p dx \tag{3.2.17}$$

is the integrated pressure along the nozzle with the cross sectional area given by

$$A(x) = \begin{cases} 2 & -1 \le x < -\frac{1}{2} \\ 1 + \sin^2(\pi x) & -\frac{1}{2} \le x < \frac{1}{2} \\ 2 & \frac{1}{2} \le x \le 1 \end{cases} .$$

The governing Euler equations

$$R(U;\alpha) = \frac{d}{dx}(A\mathcal{F}) - \frac{dA}{dx}P = 0$$

are first linearized about a known flow solution with perturbation $\tilde{U}$ in the flow variables and $\tilde{A}$ in the geometric shape to give

$$\begin{array}{ccc} \underbrace{\left(\frac{d}{dx}(A\frac{\partial \mathcal{F}}{\partial U}\tilde{U}) - \frac{dA}{dx}\frac{\partial P}{\partial U}\right)} & = & \underbrace{\left(\frac{d\tilde{A}}{dx}P - \frac{d}{dx}(\tilde{A}\mathcal{F})\right)} \\ \mathcal{L}\tilde{U} & = & f, \end{array} \tag{3.2.18}$$

where $P = [0, p, 0]^T$. The adjoint variables $\lambda = [\lambda_1(x),\ \lambda_2(x),\ \lambda_3(x)]^T$ are chosen to satisfy the adjoint equations,

$$-AB^T\frac{d\lambda}{dx} - A_x(\frac{\partial P}{\partial U})^T\lambda - (\frac{\partial p}{\partial U})^T = 0, \tag{3.2.19}$$

and the corresponding boundary conditions at the inlet and outlet such that the first variation of the cost functional (3.2.17) can be reduced to

$$\mathcal{I} = \int_{-1}^{1} \lambda^T f dx. \tag{3.2.20}$$

Eqs. (3.2.18) accounts for the relationship between the perturbations in flow variables and the perturbations in geometric design parameters; the source term f is a distributed source along the nozzle. The adjoint equations (3.2.19) do not involve any information from geometric perturbations; therefore, adjoints are actually independent of the geometric perturbations. What is influenced by the geometric perturbations, or source term f, is the first variation of the cost functional given by (3.2.20). This observation leads to the approach of Green's functions to solve the adjoint equation. If the distributed source term f in the linearized Euler Eqs. (3.2.18) is replaced by a point source, i.e.,

$$\mathcal{L}\tilde{U} = f(x)\delta(x-\xi), \tag{3.2.21}$$

where $\delta(\cdot)$ is the Dirac delta function, Eq. (3.2.20) is accordingly reduced to

$$\mathcal{I} = \int_{-1}^{-1} \lambda^T(x) f(x)\delta(x-\xi)dx = \lambda^T(\xi)f(\xi).$$

The basic idea is to choose three linearly independent source terms, f_j, $j = 1, 2, 3$, such that the adjoints can be obtained from the resulting linear system

$$\lambda(\xi) = (f_1(\xi)|f_2(\xi)|f_3(\xi))^{-T}(\mathcal{I}_1(\xi)|\mathcal{I}_2(\xi)|\mathcal{I}_3(\xi))^T. \tag{3.2.22}$$

To find the solutions to Eq. (3.2.21), we start by considering the solutions to the homogeneous equations $\mathcal{L}\tilde{U} = 0$ in which only perturbations in flow variables exist and geometric perturbations vanish. Since for isentropic transonic nozzle flows, the mass flow rate $mA = \rho uA$, the stagnation enthalpy H and stagnation pressure p_0 remain constant along the nozzle, the perturbations of conservative variables $\tilde{U}$ can be expressed as the linear combination of perturbations due to three quantities, m, H and p_0,

$$\tilde{U}(x) = \frac{a}{A(x)}\frac{\partial U}{\partial m}(x)|_{H,p_0} + b\frac{\partial U}{\partial H}(x)|_{p_0,M} + c\frac{\partial U}{\partial p_0}(x)|_{H,M}, \tag{3.2.23}$$

where a, b and c are unit perturbations in U due to changes in mA, H and p_0, respectively. Furthermore, the perturbations due to stagnation enthalpy and pressure are introduced at frozen Mach number M for the purpose of simplicity. It can be verified that (3.2.23) is the general solution to the homogeneous linearized Euler equation.

Corresponding to the source term $f_j(x)\delta(x - \xi)$, the inhomogeneous solutions should take the form

$$\tilde{U}_j(x, \xi) = a(x, \xi)\frac{1}{A(x)}\frac{\partial U}{\partial m}(x)|_{H,p_0} + b(x, \xi)\frac{\partial U}{\partial H}(x)|_{p_0,M} + c(x, \xi)\frac{\partial U}{\partial p_0}|_{H,M}. \tag{3.2.24}$$

Since the source term is only located at $x = \xi$, the inhomogeneous equations (3.2.21) degenerate to homogeneous equations on either side of source location. Therefore, a, b and c should take the constant values on each side of ξ; let $a = a_1$, $b = b_1$ and $c = c_1$ in the region of $x < \xi$ and $a = a_2$, $b = b_2$ and $c = c_2$ in the region of $x > \xi$. Then the nonhomogeneous equation (3.2.21) are integrated from $x = \xi^-$ to $x = \xi^+$ to determine the jump relationship between these constants,

$$A(\xi)((a_2 - a_1)\frac{1}{A(\xi)}\frac{\partial \mathcal{F}}{\partial m}(\xi)|_{H,p_0} + (b_2 - b_1)\frac{\partial \mathcal{F}}{\partial H}(\xi)|_{p_0,M} + (c_2 - c_1)\frac{\partial \mathcal{F}}{\partial p_0}(\xi)|_{H,M}) = f_j(\xi),$$

which will give the jump relation if we choose three linearly independent vectors,

$$
\begin{aligned}
f_1(\xi) &= \frac{\partial \mathcal{F}}{\partial m}|_{H,p_0} = \left\{ \begin{array}{c} 1 \\ q \\ H \end{array} \right\}, \\
f_2(\xi) &= A(\xi)\frac{\partial \mathcal{F}}{\partial H}|_{p_0,M} = A(\xi) \left\{ \begin{array}{c} -\frac{\rho u}{2H} \\ 0 \\ \frac{\rho u}{2} \end{array} \right\}, \\
f_3(\xi) &= A(\xi)\frac{\partial \mathcal{F}}{\partial p_0}|_{H,M} = \frac{A(\xi)}{p_0} \left\{ \begin{array}{c} \rho u \\ \rho u^2 + p \\ \rho u H \end{array} \right\}.
\end{aligned}
$$

For transonic isentropic nozzle flow, the stagnation pressure and enthalpy should be specified at the inlet; the Mach number should be such that the throat is choked. At the outlet no boundary conditions can be imposed since the flow is supersonic at that location. When $f_j = f_1$, i.e., the stagnation enthalpy and pressure are frozen and only the mass flow rate is perturbed, we have $a_2 - a_1 = 1, b_1 = b_2 = 0$, and $c_1 = c_2 = 0$. Because the nozzle is choked at the throat, the mass flow rate at that location should remain unchanged through perturbations. This observation requires $a = 0$ at the throat. Thus, we get the perturbation $\tilde{U}_1$ corresponding to the source term f_1,

$$
\tilde{U}_1(x,\xi) = \left\{ \begin{array}{ll} -\mathbf{H}(\xi - x)\frac{1}{A(x)}\frac{\partial U}{\partial m}(x)|_{H,p_0} & \xi < 0 \\ \mathbf{H}(x - \xi)\frac{1}{A(x)}\frac{\partial U}{\partial m}(x)|_{H,p_0} & \xi > 0 \end{array} \right.
$$

where $\mathbf{H}$ is the Heaviside function. When we choose $f_j = f_2$, and therefore allow perturbations in stagnation enthalpy with frozen stagnation pressure and Mach number, the jump relations are $b_2 - b_1 = 1$, $a_1 = a_2 = 0$, and $c_1 = c_2 = 0$. Since the flow boundary condition at the inlet requires stagnation enthalpy to be specified, we have $b_1 = 0$, and therefore $b_2 = 1$. Thus, the perturbations of conservative variables corresponding to source term f_2 can be written in the form,

$$
\tilde{U}_2(x,\xi) = \mathbf{H}(x - \xi)\frac{\partial U}{\partial H}(x)|_{p_0,M}.
$$

Similarly for the case $f_j = f_3$, $a_1 = a_2 = 0$, $b_1 = b_2 = 0$ and $c_2 - c_1 = 1$. From consideration of the flow boundary conditions at the inlet, we should have $c_1 = 0$, thus, we arrive at

$$\tilde{U}_3(x,\xi) = \mathbf{H}(x-\xi)\frac{\partial U}{\partial p_0}(x)|_{H,M}.$$

The first variations of the cost functional are given by

$$\mathcal{I}_j(\xi) = \int_{-1}^{1} \frac{\partial p}{\partial U}\tilde{U}_j dx,$$

for $j = 1, 2, 3$, that is,

$$\begin{aligned}
\mathcal{I}_1(\xi) &= \begin{cases} -\int_{-1}^{\xi} \frac{1}{A(\xi)}\frac{\partial p}{\partial m}|_{H,p_0} dx & \xi < 0 \\ \int_{\xi}^{1} \frac{1}{A(\xi)}\frac{\partial p}{\partial m}|_{H,p_0} dx & \xi > 0 \end{cases} \\
\mathcal{I}_2(\xi) &= 0 \\
\mathcal{I}_3(\xi) &= \int_{\xi}^{1} \frac{\partial p}{\partial p_0}(x)|_{H,M} dx.
\end{aligned}$$

With source terms f_j and variational cost functionals $\mathcal{I}_j$ in hand, the adjoints can be obtained from (3.2.22).

3.3 Properties of Adjoints at Sonic Throat

For the isentropic transonic nozzle flow, when x is approaching zero, the nozzle area and the Mach number can be represented as

$$\begin{aligned}
A(x) &\sim A_0 + O(x^2) \\
M(x) &\sim 1 + O(x),
\end{aligned}$$

so

$$\frac{1}{A(x)}\frac{\partial p}{\partial m}|_{H,M} = \frac{-u}{1-M^2} \sim \frac{1}{x}.$$

It follows that there is a logarithmic singularity for the adjoints at the sonic throat [27, 28]. This singularity results from the singularity of the flux Jacobian $B = \frac{\partial F}{\partial Q}$ at $M = 1$ and

the vanishing area derivative, A_x, at the throat, since adjoint equations (3.2.19) can not be solved uniquely for $\frac{d\lambda}{dx}$ under these conditions.

Generally, this singularity phenomenon at the sonic line will not occur in 2-D airfoil flows [26], because of non-orthogonality of the sonic line to the velocity vector and the disappearance of source term in the 2-D adjoint equations. However, there will appear a new kind of singularity in 2-D airfoil flows: the stagnation point. It's difficult to analyze the adjoint properties at this point.

3.4 Steger-Warming Flux-Vector-Splitting Scheme

The solutions to the adjoint equations (3.1.7) can be viewed as the asymptotic solution to the corresponding unsteady $PDEs$

$$\frac{\partial \Lambda}{\partial t} - B^T \frac{\partial \Lambda}{\partial x} + S^T \Lambda - (p - p^*)(\frac{\partial p}{\partial U})^T = 0 \tag{3.4.25}$$

where $\Lambda = [\Lambda_1(t,x), \Lambda_2(t,x), \Lambda_3(t,x)]^T$. The choice of sign for the unsteady term is dictated by the requirement that (3.4.25) be forward time stable. Meanwhile the transversality conditions (3.1.4,3.1.5) can be rewritten as

$$(\frac{\partial \mathcal{F}}{\partial U})^T|_{x=0} \Lambda(t,0) = \left[(\frac{\partial p_0}{\partial U})^T|_{x=0} \quad (\frac{\partial H}{\partial U})^T|_{x=0} \right] \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix},$$

and

$$(\frac{\partial \mathcal{F}}{\partial U})^T|_{x=1} \Lambda(t,1) = \nu (\frac{\partial p}{\partial U})^T|_{x=1}.$$

Then the boundary conditions for adjoints are formulated as, at the inlet[4]

$$\Lambda_1(t,0) + u\Lambda_2(t,0) + \left[\frac{\gamma p}{(\gamma - 1)\rho} + \frac{1}{2}u^2 \right] \Lambda_3(t,0) = 0$$

[4]In calculating Jacobians of H and p_0, the use of relationships from compressible flows is taken: $p_0 = p(1 + \frac{\gamma-1}{2}M^2)^{\frac{\gamma}{\gamma-1}}$, $H = E + \frac{p}{\rho}$, and p is related to the conservative variables by state equation $p = (\gamma - 1)(\rho E - \frac{1}{2}\rho u^2)$.

and at the outlet

$$\Lambda_1(t,1) + 2u\Lambda_2(t,1) + \left[\frac{\gamma p}{(\gamma-1)\rho} + \frac{3}{2}u^2\right]\Lambda_3(t,1) = 0$$
$$u\Lambda_2(t,1) + \left[\frac{\gamma p}{(\gamma-1)\rho} + u^2\right]\Lambda_3(t,1) = 0.$$

The discontinuity in the flow solution will present difficulties in numerical approximations. The approach here is to solve the adjoint equation using a flux-vector-splitting scheme without explicitly applying the Rankine-Hugoniot condition in a shock-fitting way at the shock. The similarity between the nonhomogeneous adjoint equation (3.4.25) and the unsteady Euler equations indicates that the same flux-vector- splitting scheme can be developed to solve the adjoint equations as to solve the Euler equation. In what follows, the Steger-Warming flux-vector-splitting scheme is outlined to solve the adjoint equations. This method starts by splitting the flux Jacobian into a minus and a plus part according to the signs of its eigenvalues. Then the CIR [5] scheme is utilized to approximate space derivatives in an upwind way.

The Jacobian of the flux vector B can be written as $B = PDP^{-1}$, where the matrix of right eigenvectors P and the eigenvalues D are given by,

$$P = \begin{pmatrix} 1 & 1 & 1 \\ u & u+a & u-a \\ \frac{u^2}{2} & H+ua & H-ua \end{pmatrix}$$

and D =diag$(u,\ \ u+a,\ \ u-a)$, along with a being the speed of sound. Then the adjoint equation can be reduced to reflect its characteristic pattern as follows

$$P^T\frac{\partial \Lambda}{\partial t} - DP^T\frac{\partial \Lambda}{\partial x} + P^T S^T \Lambda - (p-p^*)P^T(\frac{\partial p}{\partial U})^T = 0$$

If we denote the local increment $\Delta W = P^T\Delta\Lambda$, the finite difference form of the above

[5] CIR (Courant-Isaacson-Rees) scheme uses upwind difference for the spatial difference depending on the sign of the local wave speed, i.e., when the local wave speed is negative, the forward difference is adopted to approximate the space difference, otherwise, the backward difference is employed.

equation can be written as

$$\frac{\Delta W}{\Delta t} = D\frac{\Delta W}{\Delta x} - P^T S^T \Lambda + (p - p^*) P^T (\frac{\partial p}{\partial U})^T \tag{3.4.26}$$

The time integration is performed by a 4-stage Runge-Kutta method and space difference is handled by Steger-Warming flux-vector-splitting method. The operator D is split into two parts $D = D^+ + D^-$,

$$D^+ = \frac{1}{2}\begin{pmatrix} u+|u| & 0 & 0 \\ 0 & u+a+|u+a| & 0 \\ 0 & 0 & u-a+|u-a| \end{pmatrix};$$

$$D^- = \frac{1}{2}\begin{pmatrix} u-|u| & 0 & 0 \\ 0 & u+a-|u+a| & 0 \\ 0 & 0 & u-a-|u-a| \end{pmatrix}.$$

In contrast to the CIR scheme used in the flow solver[6], the space difference in adjoint equation is approximated by

$$D\frac{\Delta W}{\Delta x} = \frac{1}{\Delta x}\big((D^+(U_i)\Delta W_i^+ + D^-(U_i)\Delta W_i^-\big)$$

where $\Delta W_i^{\pm} = P^T(\Lambda^{\pm}_{i+\frac{1}{2}} - \Lambda^{\pm}_{i-\frac{1}{2}})$, and adjoints on the cell interfaces are given by $MUSCL$ (monotone upstream-centered schemes for conservation laws) scheme,

$$\begin{aligned} \Lambda^-_{i+\frac{1}{2}} &= \left[I + \frac{\phi}{4}\big((1+\kappa)\Delta + (1-\kappa)\nabla\big)\right]\Lambda_i \\ \Lambda^+_{i+\frac{1}{2}} &= \left[I - \frac{\phi}{4}\big((1-\kappa)\Delta + (1+\kappa)\nabla\big)\right]\Lambda_{i+1}, \end{aligned} \tag{3.4.27}$$

where Δ denotes forward difference operator and ∇ backward. The accuracy order and the type of the scheme are determined by κ and ϕ. For $\phi = 0$, the scheme is of first order. When $\phi = 1$, κ can be chosen to obtain different schemes of higher order, i.e.,

$$\kappa = \begin{cases} -1 & 2^{nd} \text{ order upwind} \\ 1 & \text{central differencing} \\ \frac{1}{3} & \text{upwind bias} \end{cases}.$$

[6]In [37], Ibrahim *et al.* employed the same CIR scheme to the adjoint solver as generally used in flow solver, and conducted the stability analysis for the adjoint solver.

In the present research, we take $\kappa = \frac{1}{3}$. A nonlinear filter should be applied in flow regions with large gradients or discontinuities. Here we employ *minmod* limiter; Δ and ∇ in the *MUSCL* scheme (3.4.27) are respectively replaced by $\bar{\Delta}$ and $\bar{\nabla}$

$$\begin{aligned} \bar{\Delta} &= minmod\{\Delta, \beta\nabla\} \\ \bar{\nabla} &= minmod\{\nabla, \beta\Delta\} \end{aligned} \tag{3.4.28}$$

where *minmod* function is defined as

$$minmod\{x, y\} = sgn(x)max\{0, min[xsgn(y), ysgn(x)]\}$$

with $\beta = \frac{3-\kappa}{1-\kappa}$. The limiter behaves as artificial dissipation to smear the spurious vibration around the shock region. In numerical methods, the 4-stage Runge-Kutta scheme

$$\begin{aligned} W^{(0)} &= W^n \\ W^{(1)} &= W^{(0)} + \alpha_1 \Delta t R(W^{(0)}) \\ W^{(2)} &= W^{(0)} + \alpha_2 \Delta t R(W^{(1)}) \\ W^{(3)} &= W^{(0)} + \alpha_3 \Delta t R(W^{(2)}) \\ W^{(4)} &= W^{(0)} + \alpha_4 \Delta t R(W^{(3)}) \\ W^{n+1} &= W^{(4)} \end{aligned} \tag{3.4.29}$$

is applied for time integration, where R is the right hand side of (3.4.26).

3.5 Computational Tests

Three computational tests are presented here, the first of which concerns the isentropic transonic flow and in the second of which the adjoints for a shocked nozzle flow are obtained to calculate the reduced gradient of a cost functional. In the third test, an inverse design problem is presented.

Isentropic Transonic Case

In this case, a geometric design variable α is introduced to allow modification of the nozzle shape,

$$A(x) = \begin{cases} 2 & -1 \le x < -\frac{1}{2} \\ 1 + \sin^2(\pi x) + \alpha x(x^2 - 0.25)^2 & -\frac{1}{2} \le x < \frac{1}{2} \\ 2 & \frac{1}{2} \le x \le 1 \end{cases} .$$

The Mach number at the inlet is $M_i = 0.3059$ such that the flow is choked at the throat $x = 0$. The stagnation enthalpy and pressure at the inlet are $H = 4$ and $p_0 = 2$, respectively. The flow variables and adjoints are calculated at $\alpha = 0$ on the same grid system with 161 grid points. In numerical experiments, the numerical methods introduced in the previous section are applied to solve the adjoint equations (3.2.19); the analytical solutions for adjoints are obtained via Green's function approach. Fig. 3.2 and 3.3 show the comparisons of adjoints from analytical solutions and numerical solutions on the uniform and nonuniform grid system. As seen from these figures, the nonuniform grid system (with grid points concentrated toward the throat) can improve the resolution for the adjoint singularity at the throat. Since flow variables (state variables) and adjoints (co-state variables) may have different regularity properties at some locations (such as shock and sonic throat), different grid systems may be adopted to the flow solver and the adjoint solver. For simplicity, however, the same grid system is used for both solvers in these computational tests to avoid interpolating flow variables to a different grid system. In Table 3.1 the gradients of the cost functional (3.2.17) are displayed for the purpose of comparisons. The 'uniform' and 'nonuniform' results are obtained via using numerical adjoint solutions on the uniform and nonuniform grid systems, respectively. The 'analytical' result is obtained via using the analytical adjoint solutions. The gradients from the adjoint methods agree with each other very well, even though about 10% error exists when compared to the result form the FDM.

A problem of slow convergence arises with the nonuniform grid system in the adjoint solver. The RPM is employed to enhance convergence (see Appendix B).

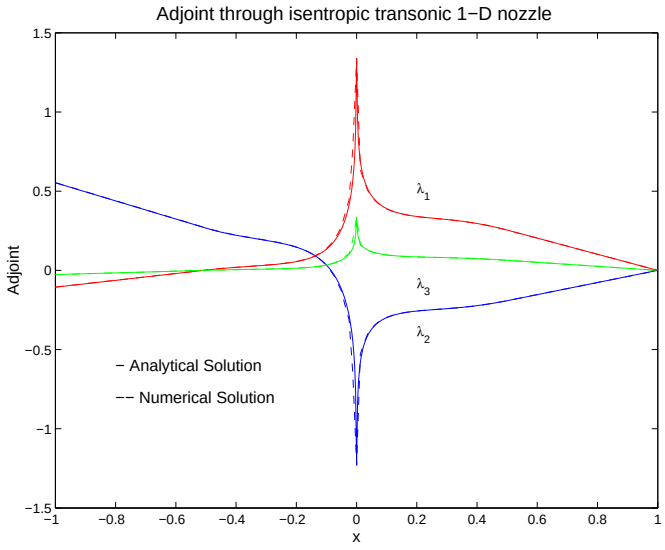

Figure 3.2: Comparison of adjoints from numerical and analytical solutions on uniform grid system

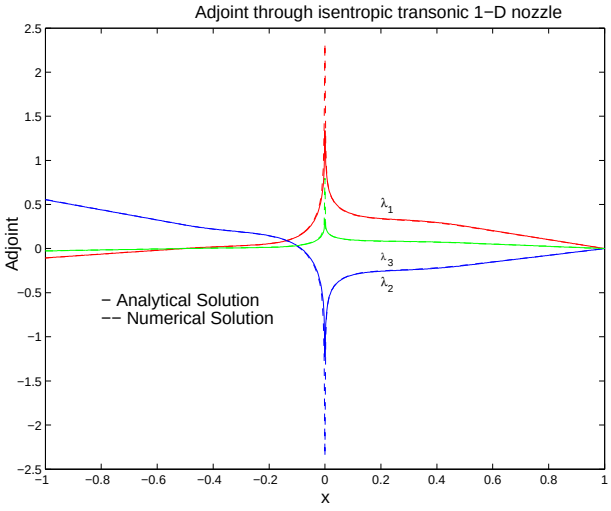

Figure 3.3: Comparison of adjoints from numerical and analytical solutions on nonuniform grid system

	uniform	nonuniform	analytical	FDM
$\frac{\partial \mathcal{I}}{\partial \alpha}$	-5.564e-3	-5.554e-3	-5.534e-3	-5.002e-3

Table 3.1: Comparisons of gradients of cost functional for isentropic transonic quasi 1-D nozzle flow

Shocked flow

In this part, the nozzle shape is given by

$$A(x;\alpha) = \alpha x^2 - \sqrt{0.8\alpha}x + 1. \tag{3.5.30}$$

For this family of nozzles, the area of the throat is independent of design parameter α, however, the location of the throat depends on α. The Mach number at the inlet, M_i is set to be 0.5533 to make the throat choked; and the pressure p_i and the temperature T_i at the inlet are $10^5\ Pa$ and $10^4\ K$, respectively. The pressure at the outlet is specified, $p_{back} = 0.9247 \times 10^5\ Pa$, such that a shock appears somewhere between the throat and the outlet. The pressure distribution at $\alpha = 0.6$ is the target, denoted by p^* ; and the initial guess for α is 0.8. Fig. 3.4 shows the static pressure distribution along the nozzle at $\alpha = 0.6$. Fig. 3.5 displays the pressure distribution at $\alpha = 0.8$. In both figures, the solid lines represent 'exact' solutions, and the circles denote the numerical solutions. The 'exact' flow solver utilizes the MATLAB built-in functions FSOLVE and ODE45 in a shock-fitting way. The numerical flow solver employs van Leer's flux splitting approach with 3^{rd} order $MUSCL$ scheme; the *minmod* limiter is also adopted. As seen from these figures, the numerical solutions accurately predict the shock location, except that there exist small spurious oscillations before and after the shock. The cost functional is defined by $I = \frac{1}{2}\int_0^1 (p - p^*)^2 dx$. The distributions of three adjoint components solved from adjoint equations (3.1.7) are displayed in Fig. 3.6. As expected, there is a singularity at the throat location $x = 0.5$; meanwhile adjoints are continuous at the shock location $x = 0.85$, even though they are not smooth. The convergence history of adjoint solver is displayed in Fig. 3.7. The gradients of the cost functional from the adjoint method (ADJ) and the finite

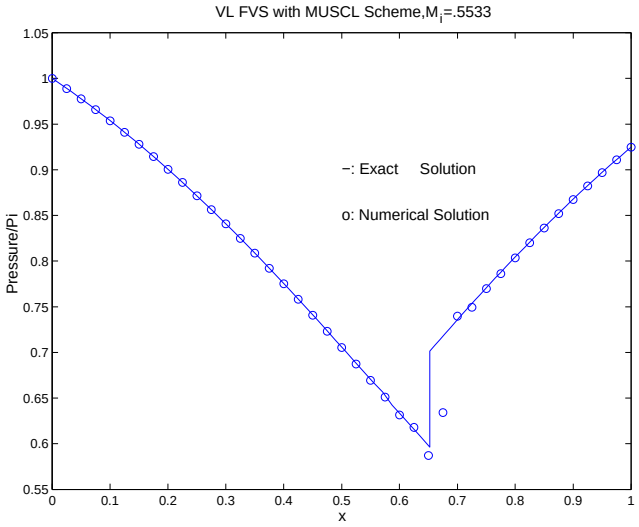

Figure 3.4: Target pressure distribution, $\alpha = 0.6$

	ADJ	FDM
$\frac{\partial I}{\partial \alpha}$	4.6590e8	4.4546e8

Table 3.2: Comparisons of gradients of cost functional for shocked quasi 1-D nozzle flow

difference method (FDM) are compared in Table 3.2.

The above computational tests reveal that the proposed numerical method can accurately predict adjoints at reasonable computational costs; the singularity in adjoints can also be sharply captured at the sonic throat. Therefore, the resulting gradients of cost functionals agree well with the FDM results. In what follows, the adjoint method is integrated with the flow solver to conduct a pressure-matching inverse design.

Pressure-matching inverse design

In this case, the nozzle shape is given by (3.5.30), and the cost functional is normalized by the pressure at the inlet, p_i, i.e.,

$$I = \frac{1}{2}\int_0^1 (\frac{p}{p_i} - \frac{p^*}{p_i})^2 dx.$$

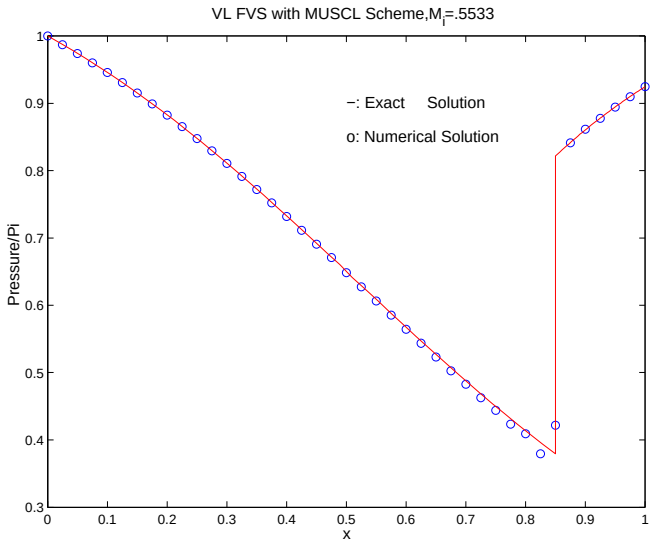

Figure 3.5: Initial pressure distribution, $\alpha = 0.8$

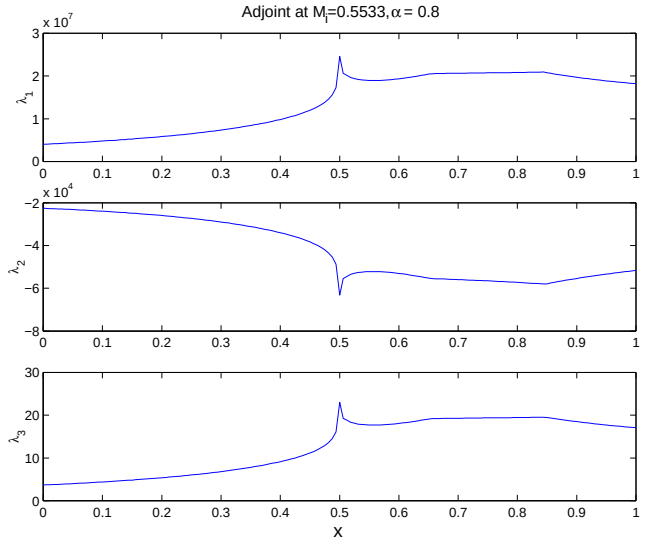

Figure 3.6: Adjoints at $M_i = 0.5533, \alpha = 0.8$

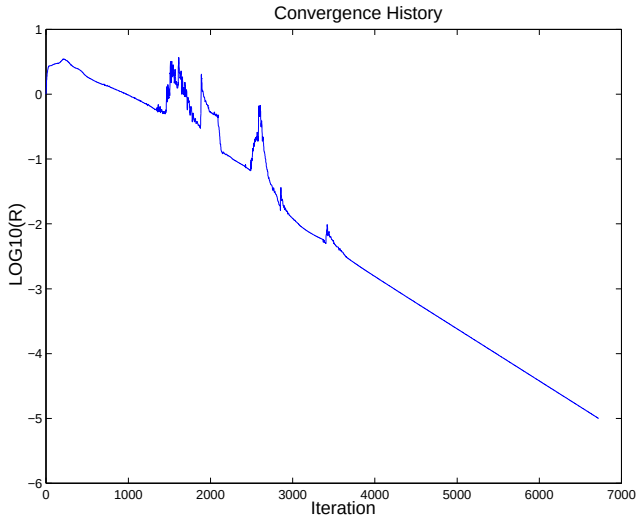

Figure 3.7: Convergence history of adjoint solver on uniform grid system

The flow conditions at the inlet and the outlet remain the same as in the above test case. The target pressure distribution is obtained by setting $\alpha^* = 0.6$; and the initial guess for the geometric parameter is $\alpha = 0.8$. The inverse design is conducted by using FMINCON from MATLAB's optimization toolbox suite and converged after 9 iterations to satisfy the stopping criterion that the function value changing is less than the specified tolerance. As seen from Table 3.3, the cost functional decreases from 0.425763 to $7.1501e-7$, and the first-order optimality condition decreases from 0.0113 to $1.21e-5$. The optimal design parameter is 0.600750 which is very close to the desired design parameter $\alpha^* = 0.6$. Fig. 3.8 displays the optimization procedure with $*$ denoting the cost functional and $+$ representing the optimality condition.

	Cost functional	Optimality condition	Design variable
Starting	0.425763	0.0113	0.8
Optimal	7.1501e-7	1.21e-5	0.600750

Table 3.3: Results of pressure-matching inverse design for quasi 1-D nozzle flow

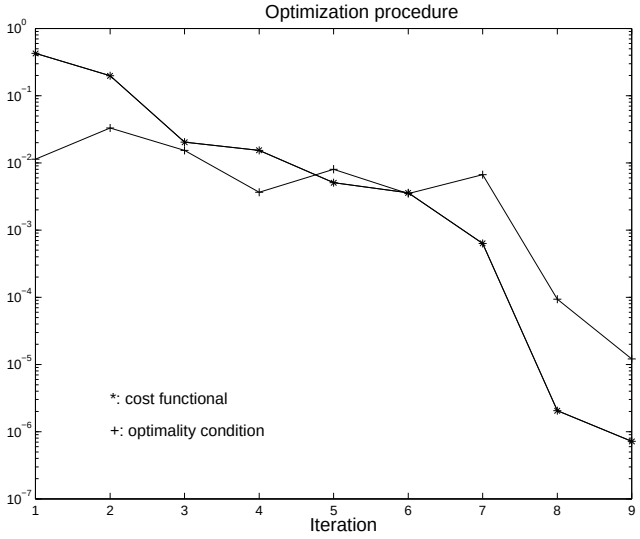

Figure 3.8: Optimization procedure of pressure-matching inverse design for quasi 1-D nozzle flow

Chapter 4

Transonic Airfoil Optimal Design Using Adjoint Methods

In this chapter we consider a constrained optimization problem involving minimizing the drag coefficient of an airfoil by modifying the airfoil geometry and the angle of attack, while maintaining fixed lift coefficient. The mathematical formulation of the problem has been outlined in the Section 2.5. In the design problem, two aerodynamic functionals are encountered, namely, C_D and C_L, whose gradients with respect to the design variables are important information in optimization procedures. After dividing variables into state variables and control variables, a Lagrange Multiplier Theorem (A.3.1) will be utilized to calculate the reduced gradients of these aerodynamic functionals.

First, the 2-D Euler equations in the body-fitted coordinate are reviewed followed by derivations of adjoint equations, characteristic adjoint boundary conditions and reduced gradients. In Section 4.3, a super-reduced problem is explored, with two approaches outlined to obtain the super-reduced gradients of the drag functional subject to the lift constraint. A variant of Jameson's finite volume scheme, which is employed to solve the resulting adjoint equations, is then briefly described in Section 4.4. In numerical experiments, an efficient hyperbolic grid generator with adapted artificial dissipation is utilized in Section 4.5 to generate smooth and orthogonal O-type grid systems for the airfoil. Two ways of parameterizing the airfoil shape are investigated in Section 4.6: one uses a formulation of the $NACA$ 4-digit airfoil series [1],

while the other uses Hicks & Henne [33] shape functions to model the geometric changes. Good agreement between the gradients from the adjoint method and FDM is achieved for both parameterizations. Finally in Section 4.7, several transonic airfoil optimization designs are conducted using a gradient-based optimization method with reduced and super-reduced gradients provided by the adjoint method.

4.1 2-D Euler Equations

For 2-D inviscid flows of a perfect gas, the unsteady Euler equations take the following form in the Cartesian system (x, y)

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = \mathbf{0}, \tag{4.1.1}$$

where

$$Q = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{Bmatrix}, F = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uH \end{Bmatrix}, G = \begin{Bmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vH \end{Bmatrix},$$

and standard notations are adopted, with the stagnation enthalpy $H = E + \frac{p}{\rho}$. Also we have the equation of state which relates the static pressure to the flow variables

$$p = (\gamma - 1)\rho(E - \frac{u^2 + v^2}{2}),$$

where γ is the ratio of the specific heats. The non-penetrating boundary condition is imposed on the airfoil. When a numerical method is employed to solve the Euler equations, the flow domain is restricted to a finite region which necessitates far-field boundary conditions. Preferably, these boundary conditions should not admit reflections from the far-field back into the flow domain.

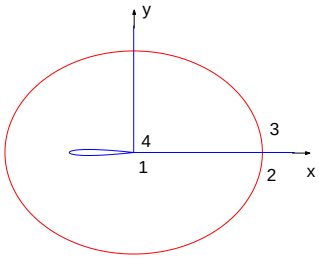

Figure 4.1: Physical domain of O-type grid

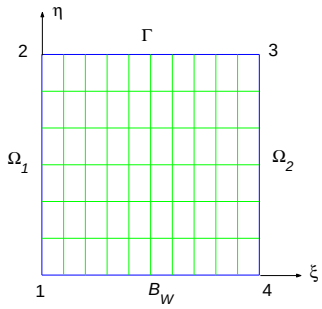

Figure 4.2: Computational domain (from O-type grid)

Since geometrical design parameters will be considered in the optimization design, it is convenient to rewrite (4.1.1) in a body-conforming coordinate system. To this purpose, a geometric transformation will be introduced to transform the physical domain (x, y) into the computational domain (ξ, η). The transformation $TO : (x, y) \longmapsto (\xi, \eta)$ will be explored, which adopts O-type grid topology in the physical domain. Fig. 4.1 sketches the physical domain of O-type grid topology, while the transformed computational domain is displayed in Fig. 4.2.

For this transformation, the Jacobian K_{ij}, and its determinant J are given by

$$[K_{ij}] = \frac{\partial(\xi,\eta)}{\partial(x,y)}, \quad J = \det(K_{ij}).$$

The steady Euler equations can be written in the computational domain $\mathcal{D}$ as

$$\frac{\partial \hat{F}}{\partial \xi} + \frac{\partial \hat{G}}{\partial \eta} = \mathbf{0}, \tag{4.1.2}$$

with

$$\hat{F}(Q) = S_{11}F(Q) + S_{12}G(Q), \quad \hat{G}(Q) = S_{21}F(Q) + S_{22}G(Q),$$

and

$$S_{ij} = \tfrac{1}{J}K_{ij}$$

being scalar metrics.

4.1.1 Flow boundary conditions

The non-penetrating wall boundary condition is imposed on the airfoil surface $\mathcal{B}_{\mathcal{W}}(\eta = 0)$ and can be written as

$$S_{21}u + S_{22}v = 0. \tag{4.1.3}$$

On the boundaries $\Omega_1(\xi = 0)$ and $\Omega_2(\xi = \xi_{max})$, periodic conditions under the TO transformation give

$$Q|_{\Omega_1} = Q|_{\Omega_2}. \tag{4.1.4}$$

The solutions to the steady Euler equations (4.1.2) are obtained by solving its unsteady version via marching in the time direction to a steady state. The unsteady Euler equations are hyperbolic $PDEs$ [34], therefore the boundary conditions on the far-field boundary Γ can be formulated based on an analysis of the characteristics.

Let $A_n = n_x\frac{\partial F}{\partial Q} + n_y\frac{\partial G}{\partial Q}$, where the outward unit normal is $n_x = \frac{S_{21}}{\sqrt{S_{21}^2+S_{22}^2}}, n_y = \frac{S_{22}}{\sqrt{S_{21}^2+S_{22}^2}}$. Under the assumption that A_n is non-degenerate, there exists a nonsingular matrix R such that $A_n = RDR^{-1}$ where the diagonal matrix D is composed of the four eigenvalues $u_n + a, u_n, u_n, u_n - a$ of the matrix A_n,

$$R = \begin{pmatrix} \frac{2(\gamma-1)}{s_1} & \frac{2(\gamma-1)(u+n_x a)}{s_1} & \frac{2(\gamma-1)(v+n_y a)}{s_1} & 1 \\ \frac{2n_x}{s_x} & \frac{2u_n}{s_x} & 0 & 1 \\ \frac{2(n_y u - n_x v)}{s_x} & \frac{n_y u^2 - 2n_x uv - n_y v^2}{s_x} & 1 & 0 \\ \frac{2(\gamma-1)}{s_2} & \frac{2(\gamma-1)(u-n_x a)}{s_2} & \frac{2(\gamma-1)(v-n_y a)}{s_2} & 1 \end{pmatrix}^T,$$

and

$$R^{-1} = \begin{pmatrix} -\frac{s_1(2au_n-(\gamma-1)q)}{8(\gamma-1)a^2} & \frac{s_1(n_xa-(\gamma-1)u)}{4(\gamma-1)a^2} & \frac{s_1(n_ya-(\gamma-1)v)}{4(\gamma-1)a^2} & \frac{s_1}{4a^2} \\ \frac{2a^2(n_xs_x+n_ys_y)-(\gamma-1)q^2}{4a^2} & \frac{2n_ya^2u_t+(\gamma-1)uq}{2a^2} & \frac{-2n_xa^2u_t+(\gamma-1)vq}{2a^2} & -\frac{(\gamma-1)q}{2a^2} \\ \frac{2n_yu_na^2-(\gamma-1)vq}{2a^2} & \frac{-n_xn_ya^2+(\gamma-1)uv}{a^2} & \frac{n_x^2a^2+(\gamma-1)v^2}{a^2} & -\frac{(\gamma-1)v}{a^2} \\ \frac{s_2(2au_n+(\gamma-1)q)}{8(\gamma-1)a^2} & -\frac{s_2(n_xa+(\gamma-1)u)}{4(\gamma-1)a^2} & -\frac{s_2(n_ya+(\gamma-1)v)}{4(\gamma-1)a^2} & \frac{s_2}{4a^2} \end{pmatrix}, \qquad (4.1.5)$$

with

$$\begin{aligned} s_1 &= 2a^2 + 2(\gamma-1)au_n + (\gamma-1)q, \\ s_2 &= 2a^2 - 2(\gamma-1)au_n + (\gamma-1)q, \\ s_x &= n_xu^2 + 2n_yuv - n_xv^2, \\ s_y &= n_yv^2 + 2n_xuv - n_yu^2, \end{aligned}$$

and $u_n = n_xu + n_yv$, $u_t = n_yu - n_xv$, $q = u^2 + v^2$.

Let $u_c = R^{-1}\delta Q$ be the characteristic perturbation velocity, then the characteristic far-field boundary conditions can be formulated from the various cases:

$u_n < -a$ On this part of the boundary, all eigenvalues are negative. This implies that all characteristics are incoming and therefore carry information from the exterior. The perturbation velocity u_c should be identically set to $\mathbf{0}$ [87], which implies $\delta Q = Q - Q_\infty = \mathbf{0}$ (Q_∞: the conservative flow variables of the freestream), since R is non-singular.

$-a < u_n < 0$ The local characteristic $u_n + a$ is outgoing from the interior and the other three characteristics are incoming from the exterior. Correspondingly, only the perturbation velocity associated with $u_n + a$ is free and the others should be set to zero.

$0 < u_n < a$ Since the local characteristic $u_n - a$ is incoming from the exterior on this part of surface, the perturbation velocity associated with $u_n - a$ should be set to zero.

$a < u_n$ All perturbation velocities are free.

The first and fourth cases correspond to supersonic flows and only the second and third cases occur in subsonic or transonic flows around airfoils. Accordingly, for subsonic and transonic flows, the far-field boundary Γ can be divided to two segments: Γ_1, on which $-a < u_n < 0$ and boundary conditions take the following form

$$\begin{pmatrix} u_{c_2} \\ u_{c_3} \\ u_{c_4} \end{pmatrix} = \begin{pmatrix} \frac{2a^2(n_x s_x + n_y s_y) - (\gamma-1)q^2}{4a^2} & \frac{2n_y a^2 u_t + (\gamma-1)uq}{2a^2} & \frac{-2n_x a^2 u_t + (\gamma-1)vq}{2a^2} & -\frac{(\gamma-1)q}{2a^2} \\ \frac{2n_y u_n a^2 - (\gamma-1)vq}{2a^2} & \frac{-n_x n_y a^2 + (\gamma-1)uv}{a^2} & \frac{n_x^2 a^2 + (\gamma-1)v^2}{a^2} & -\frac{(\gamma-1)v}{a^2} \\ \frac{s_2(2au_n + (\gamma-1)q)}{8(\gamma-1)a^2} & -\frac{s_2(n_x a + (\gamma-1)u)}{4(\gamma-1)a^2} & -\frac{s_2(n_y a + (\gamma-1)v)}{4(\gamma-1)a^2} & \frac{s_2}{4a^2} \end{pmatrix} \delta Q = \mathbf{0}, \tag{4.1.6}$$

and, Γ_2 on which $0 < u_n < a$ and

$$u_{c_4} = \begin{pmatrix} \frac{s_2(2au_n + (\gamma-1)q)}{8(\gamma-1)a^2} & -\frac{s_2(n_x a + (\gamma-1)u)}{4(\gamma-1)a^2} & -\frac{s_2(n_y a + (\gamma-1)v)}{4(\gamma-1)a^2} & \frac{s_2}{4a^2} \end{pmatrix} \delta Q = 0. \tag{4.1.7}$$

Note that the 3×4 matrix in (4.1.6) is composed of the lower three rows of R^{-1} in (4.1.5), and the 1×4 matrix in (4.1.7) is its last row.

Numerical methods to solve the Euler equations can be found in [5, 45, 21, 87]. As a practical method, characteristic boundary conditions look like overkill, in the sense that simplified boundary conditions work well for most problems [87]. As the most popular treatment with the far-field boundary condition in 2-D or 3-D Euler solvers, the 1-D Riemann invariants are utilized since the flow in the far-field can be approximated as 1-D flow by neglecting the derivatives in the other one (or two in 3-D case) direction. Even though the characteristic boundary conditions (4.1.6) and (4.1.7) are not adopted in most Euler solvers, they are important in the investigation of the adjoint characteristic boundary conditions.

4.2 Optimality System on Computational Domain

In this section the optimality system is explored in the computational domain transformed from the physical domain of O-type topology. First the Lagrangian functional is constructed by adjoining the weak form constraints (Euler equations with $B.C.$) to the cost functional.

By setting the derivatives of the Lagrangian functional with respect to the state variables to zero, we obtain the adjoint equations and corresponding boundary conditions. The adjoints can be computed from the boundary value problem (BVP) via a numerical scheme. With the adjoints in hand, the reduced gradients of the cost functional can be obtained by differentiating the Lagrangian functional with respect to the design variables.

4.2.1 Boundary-value problem for adjoints

From the control point of view, the variables can be separated into three categories: • indepedent spatial variables (ξ, η), • state variables $Q(\xi, \eta)$, • control variables including the geometric variables $\mathcal{G}$ and a flow-related parameter, the angle of attack, α. In some cases we will be interested in a family of problems parameterized by α, while in other cases the problem will include an additional constraint among Q, $\mathcal{G}$ and α.

The Euler equations (4.1.2) on the flow domain and the boundary conditions (4.1.3), (4.1.4) (4.1.6), and (4.1.7) are viewed as constraints through which the state variables can be uniquely determined for given controls.[1] These constraints are adjoined to the cost functional via Lagrange multipliers $\mathcal{L} = [\Lambda(\xi, \eta), \psi(\xi), \nu(\xi), \mu(\xi), \omega(\eta)]$, to form the Lagrangian functional

$$\begin{aligned}
\mathcal{I}(Q; \mathcal{G}, \alpha; \mathcal{L}) &= \int_{\mathcal{B}_\mathcal{W}} I(\xi; p(Q); \mathcal{G}, \alpha) d\xi \\
&+ \int\int_{\mathcal{D}} \Lambda^T (\frac{\partial \hat{F}}{\partial \xi} + \frac{\partial \hat{G}}{\partial \eta}) d\xi d\eta \\
&+ \int_{\mathcal{B}_\mathcal{W}} \psi (S_{21} u + S_{22} v) d\xi \\
&+ \int_{\Gamma_1(\mathcal{G}, \alpha)} \nu^T P (Q - Q_\infty(\alpha)) d\xi
\end{aligned}$$

[1] Jameson in [43] reported some cases in which Euler equations admit non-unique transonic solutions in a very narrow range of angle of attack and Mach number. The reported computational cases are all at negative angles of attack and performed at very fine grid system. However, it can be assumed that these conditions will not arise in the present studies.

$$
\begin{aligned}
&+ \int_{\Gamma_2(\mathcal{G},\alpha)} \mu W(Q - Q_\infty(\alpha))d\xi \\
&+ \int_{\eta_1}^{\eta_{max}} \omega^T(Q|_{\Omega_1} - Q|_{\Omega_2})d\eta, \qquad (4.2.8)
\end{aligned}
$$

where $\Lambda(\xi,\eta) \in R^4$ with its components denoted by Λ_1, Λ_2, Λ_3 and Λ_4, and P, W are the matrices to the left of δQ in (4.1.6) and (4.1.7), respectively. Note that the boundary terms include the following multipliers, $\psi(\xi) \in R$, $\nu(\xi) \in R^3$, $\mu(\xi) \in R$ and $\omega(\eta) \in R^4$.[2]

In Jameson's approach [46, 59, 71, 44, 73], only the Euler equations were adjoined to the cost functional. This approach, from mathematical point of view, is not complete, since the constraints referred to in the Lagrange Multiplier Theorem are required to uniquely determine a solution to the states for given controls. Hence the boundary conditions are part of the state equations, and should be adjoined to the cost functional to construct the Lagrangian functional.

Integrating the 2^{nd} term of the RHS of (4.2.8) by parts and applying Green's theorem yields

$$
\begin{aligned}
\mathcal{I}(Q;\mathcal{G},\alpha;\mathcal{L}) &= \int_{\mathcal{B}_\mathcal{W}} (I - \Lambda^T\hat{G} + \psi(S_{21}u + S_{22}v))d\xi \\
&- \int\int_{\mathcal{D}} \big(\frac{\partial \Lambda^T}{\partial \xi}\hat{F} + \frac{\partial \Lambda^T}{\partial \eta}\hat{G}\big)d\xi d\eta \\
&+ \int_{\Gamma_1(\mathcal{G},\alpha)} \big(\nu^T P(Q - Q_\infty(\alpha)) + \Lambda^T\hat{G}\big)d\xi \\
&+ \int_{\Gamma_2(\mathcal{G},\alpha)} \big(\mu W(Q - Q_\infty(\alpha)) + \Lambda^T\hat{G}\big)d\xi \\
&+ \int_{\Omega_1} (\omega^T Q - \Lambda^T\hat{F})d\eta + \int_{\Omega_2} (\omega^T Q + \Lambda^T\hat{F})d\eta. \qquad (4.2.9)
\end{aligned}
$$

Differentiating (4.2.9) with respect to the state variables gives

$$
\frac{\partial \mathcal{I}}{\partial Q} = \int_{\mathcal{B}_\mathcal{W}} \big(\frac{\partial I}{\partial p}\frac{\partial p}{\partial Q} - \Lambda^T\frac{\partial \hat{G}}{\partial Q} + \psi(S_{21}\frac{\partial u}{\partial Q} + S_{22}\frac{\partial v}{\partial Q})\big)d\xi
$$

[2]Here the Euler equation and boundary conditions are represented in a weak solution form; and the Lagrange multipliers play a twofold role in this situation: they are test function in terms of weak solution [70], at the same time they act as the Lagrange multipliers from the optimization point of view.

$$
\begin{aligned}
- & \int\int_{\mathcal{D}} (\frac{\partial \Lambda^T}{\partial \xi}\frac{\partial \hat{F}}{\partial Q} + \frac{\partial \Lambda^T}{\partial \eta}\frac{\partial \hat{G}}{\partial Q}) d\xi d\eta \\
+ & \int_{\Gamma_1(\mathcal{G},\alpha)} (\nu^T P + \Lambda^T \frac{\partial \hat{G}}{\partial Q}) d\xi \\
+ & \int_{\Gamma_2(\mathcal{G},\alpha)} (\mu W + \Lambda^T \frac{\partial \hat{G}}{\partial Q}) d\xi \\
+ & \int_{\Omega_1} (\omega^T - \Lambda^T \frac{\partial \hat{F}}{\partial Q}) d\eta + \int_{\Omega_2} (-\omega^T + \Lambda^T \frac{\partial \hat{F}}{\partial Q}) d\eta.
\end{aligned}
$$

From the Lagrange Multiplier Theorem (A.3.1), the adjoint equations can be obtained by setting $\frac{\partial \mathcal{I}}{\partial Q} = \mathbf{0}$. Therefore, the governing equations for adjoints in the flow domain $\mathcal{D}$ are,

$$
\frac{\partial \Lambda^T}{\partial \xi}\frac{\partial \hat{F}}{\partial Q} + \frac{\partial \Lambda^T}{\partial \eta}\frac{\partial \hat{G}}{\partial Q} = \mathbf{0}
$$

or

$$
(\frac{\partial \hat{F}}{\partial Q})^T \frac{\partial \Lambda}{\partial \xi} + (\frac{\partial \hat{G}}{\partial Q})^T \frac{\partial \Lambda}{\partial \eta} = \mathbf{0} \tag{4.2.10}
$$

and on the airfoil $\mathcal{B}_{\mathcal{W}}$ boundary conditions are:

$$
\frac{\partial I}{\partial p}\frac{\partial p}{\partial Q} - \Lambda^T \frac{\partial \hat{G}}{\partial Q} + \psi(S_{21}\frac{\partial u}{\partial Q} + S_{22}\frac{\partial v}{\partial Q}) = \mathbf{0}, \tag{4.2.11}
$$

on the far-field boundary $\Gamma_1(\mathcal{G}, \alpha)$

$$
\nu^T P + \Lambda^T \frac{\partial \hat{G}}{\partial Q} = \mathbf{0}, \tag{4.2.12}
$$

on the far-field boundary $\Gamma_2(\mathcal{G}, \alpha)$

$$
\mu W + \Lambda^T \frac{\partial \hat{G}}{\partial Q} = \mathbf{0}, \tag{4.2.13}
$$

and on Ω_1, Ω_2

$$
\Lambda = \omega(\frac{\partial \hat{F}}{\partial Q})^{-T}. \tag{4.2.14}
$$

Eqs. (4.2.10 - 4.2.14) constitute a boundary-value problem (BVP) for the adjoints Λ. The governing equations (4.2.10) are homogeneous linear $PDEs$ and similar to the Euler equations in the non-conservative form, except that the flux Jacobian matrices are transposed.

This observation suggests that a numerical method to solve the Euler equations can be suitably modified to solve the adjoint equations. Moreover, the computational cost for solving the adjoint equation is expected to be less than the cost for solving the Euler equations, since (4.2.10) are linear $PDEs$ in that the coefficient matrices $\frac{\partial \hat{F}}{\partial Q}$ and $\frac{\partial \hat{G}}{\partial Q}$ are independent of the unknowns Λ. These matrices can be supplied from the flow solver, and thereafter remain unchanged in the procedure of solving for the adjoints. The remaining Lagrange multipliers, ψ, ν, μ, ω, which are to be used in calculating the reduced gradients, can be obtained from boundary conditions (4.2.11-4.2.14). In what follows, the boundary conditions for adjoints are investigated, and the reduced gradients are obtained via differentiating the Lagrangian functional (4.2.9) with respect to the design parameters.

4.2.2 Adjoint boundary conditions on the airfoil

In this section the boundary condition (4.2.11) on the airfoil is investigated. First, we obtain the Jacobians of p, u and v with respect to the conservative variables Q,

$$\frac{\partial p}{\partial Q} = (\gamma - 1) \begin{pmatrix} \frac{1}{2}(u^2 + v^2) \\ -u \\ -v \\ 1 \end{pmatrix}^T, \frac{\partial u}{\partial Q} = \begin{pmatrix} -\frac{u}{\rho} \\ \frac{1}{\rho} \\ 0 \\ 0 \end{pmatrix}^T, \frac{\partial v}{\partial Q} = \begin{pmatrix} -\frac{v}{\rho} \\ 0 \\ \frac{1}{\rho} \\ 0 \end{pmatrix}^T,$$

and the flux Jacobian

$$\frac{\partial \hat{G}}{\partial Q} = S_{21}\begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{\gamma-3}{2}u^2+\frac{\gamma-1}{2}v^2 & (3-\gamma)u & -(\gamma-1)v & \gamma-1 \\ -uv & v & u & 0 \\ -\gamma uE+(\gamma-1)(u^3+uv^2) & \gamma E-\frac{\gamma-1}{2}(v^2+3u^2) & -(\gamma-1)uv & \gamma u \end{pmatrix} + S_{22}\begin{pmatrix} 0 & 0 & 1 & 0 \\ -uv & v & u & 0 \\ \frac{\gamma-3}{2}v^2+\frac{\gamma-1}{2}u^2 & -(\gamma-1)u & (3-\gamma)v & \gamma-1 \\ -\gamma vE+(\gamma-1)(vu^2+v^3) & -(\gamma-1)uv & \gamma E-\frac{\gamma-1}{2}(u^2+3v^2) & \gamma v \end{pmatrix}.$$

After substituting these Jacobians into (4.2.11) and using flow boundary conditions on the airfoil $S_{21}u + S_{22}v = 0$, we get

$$(\frac{\partial \hat{G}}{\partial Q})^T\Lambda = \frac{\partial I}{\partial p}(\gamma-1)\begin{pmatrix} \frac{1}{2}(u^2+v^2) \\ -u \\ -v \\ 1 \end{pmatrix} + \psi\begin{pmatrix} 0 \\ \frac{S_{21}}{\rho} \\ \frac{S_{22}}{\rho} \\ 0 \end{pmatrix}.$$

The 4^{th} component of the above equations gives

$$\Lambda_2 S_{21} + \Lambda_3 S_{22} = \frac{\partial I}{\partial p}, \tag{4.2.15}$$

which can also be shown to satisfy its 1^{st} component. This is the physical boundary condition for the adjoints Λ on the airfoil, and will be recast to a numerical form to be incorporated into a numerical method to solve the adjoint equations. This formulation is identical to Jameson's results from the variational approach [59, 71]. Furthermore, the 2^{nd} and 3^{rd} components give

$$\psi = \rho\Lambda^T\begin{pmatrix} 1 \\ u \\ v \\ H \end{pmatrix}$$

by substituting (4.2.15) back into (4.2.11).

4.2.3 Adjoint boundary conditions on the far-field

As mentioned earlier, the outer far-field boundary is divided into two segments Γ_1, Γ_2 according to the sign of the normal velocity u_n. The adjoints will have different boundary conditions on these segments. However, the two kinds of boundary conditions will be shown to reduce to the same formulation at two points where $u_n = 0$, implying that the adjoints are continuous at these two points. Here we start from characteristic boundary conditions on Γ_1.

Recalling $\frac{\partial \hat{G}}{\partial Q} = \sqrt{S_{21}^2 + S_{22}^2} R D R^{-1}$, boundary conditions (4.2.12) on Γ_1 can be rewritten as

$$\begin{aligned}
\Lambda^T R &= -\frac{\nu^T}{\sqrt{S_{21}^2 + S_{22}^2}} P R D^{-1} \\
&= -\frac{\nu^T}{\sqrt{S_{21}^2 + S_{22}^2}} \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{u_n+a} & 0 & 0 & 0 \\ 0 & \frac{1}{u_n} & 0 & 0 \\ 0 & 0 & \frac{1}{u_n} & 0 \\ 0 & 0 & 0 & \frac{1}{u_n-a} \end{pmatrix} \\
&= -\frac{\nu^T}{\sqrt{S_{21}^2 + S_{22}^2}} \begin{pmatrix} 0 & \frac{1}{u_n} & 0 & 0 \\ 0 & 0 & \frac{1}{u_n} & 0 \\ 0 & 0 & 0 & \frac{1}{u_n-a} \end{pmatrix}.
\end{aligned} \tag{4.2.16}$$

Then the first component of $\Lambda^T R$ is zero, which means

$$\hat{l}_1^T \Lambda = 0. \tag{4.2.17}$$

with $\hat{l}_1 = (\frac{2(\gamma-1)}{s_1}, \frac{2(\gamma-1)(u+n_x a)}{s_1}, \frac{2(\gamma-1)(v+n_y a)}{s_1}, 1)^T$. Furthermore, the Lagrange multiplier ν can be expressed in terms of Λ,

$$\begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \end{pmatrix} = -\sqrt{S_{21}^2 + S_{22}^2} \begin{pmatrix} u_n & 0 & 0 \\ 0 & u_n & 0 \\ 0 & 0 & u_n - a \end{pmatrix} \mathcal{P} \Lambda. \tag{4.2.18}$$

where $\mathcal{P}$ is composed of the last three rows of R^T,

$$\mathcal{P} = \left\{ \begin{array}{cccc} \frac{2n_x}{s_x} & \frac{2u_n}{s_x} & 0 & 1 \\ \frac{2(n_y u - n_x v)}{s_x} & \frac{n_y u^2 - 2n_x uv - n_y v^2}{s_x} & 1 & 0 \\ \frac{2(\gamma-1)}{s_2} & \frac{2(\gamma-1)(u-n_x a)}{s_2} & \frac{2(\gamma-1)(v-n_y a)}{s_2} & 1 \end{array} \right\}.$$

Eq. (4.2.17) reveals that the adjoints lie in the three-dimensional nullspace of the vector $\hat{l}_1$. Therefore, we get only one condition on the adjoints. (Note that there are three conditions on flow variables on this segment of far-field boundary.)

Similarly, examination of (4.2.13) reveals the adjoint boundary conditions on Γ_2,

$$\Lambda \in \aleph \left\{ \begin{array}{cccc} \frac{2(\gamma-1)}{s_1} & \frac{2(\gamma-1)(u+n_x a)}{s_1} & \frac{2(\gamma-1)(v+n_y a)}{s_1} & 1 \\ \frac{2n_x}{s_x} & \frac{2u_n}{s_x} & 0 & 1 \\ \frac{2(n_y u - n_x v)}{s_x} & \frac{n_y u^2 - 2n_x uv - n_y v^2}{s_x} & 1 & 0 \end{array} \right\}. \tag{4.2.19}$$

The above one-dimensional nullspace is spanned by the vector

$$l = (\tfrac{2au_n + (\gamma-1)q}{2(\gamma-1)}, -\tfrac{n_x a + (\gamma-1)u}{\gamma-1}, -\tfrac{n_y a + (\gamma-1)v}{\gamma-1}, 1)^T.$$

Thus, there are three boundary condition on the adjoints on this segment, where there exists only one boundary condition for flow variables. The Lagrange multiplier μ is given by

$$\mu = -\sqrt{S_{21}^2 + S_{22}^2}(u_n - a)\hat{l}_2^T \Lambda, \tag{4.2.20}$$

where $\hat{l}_2 = (\frac{2(\gamma-1)}{s_2}, \frac{2(\gamma-1)(u-n_x a)}{s_2}, \frac{2(\gamma-1)(v-n_y a)}{s_2}, 1)^T$.

At the far-field boundary of the O-type grid system, there exist two points ξ_s, ξ_e where the freestream is tangent to the surface so that $u_n = 0$.[3] At each side of these points, different boundary conditions (4.2.17) and (4.2.19) should be imposed on the adjoints. However, boundary condition (4.2.19) can reduce to be identical to (4.2.17) by applying $u_n = 0$. To this purpose, we can rewrite (4.2.13) as

$$\begin{aligned} & (0,0,0,\mu)R^{-1} + \sqrt{S_{21}^2 + S_{22}^2}\Lambda^T R D R^{-1} \\ = \quad & ((0,0,0,\mu) + \sqrt{S_{21}^2 + S_{22}^2}\Lambda^T R D)R^{-1} = \mathbf{0} \end{aligned}$$

[3]We assume that the outer boundary in the physical domain is strictly convex.

by augmenting W to R^{-1}. Meanwhile, $\Lambda^T RD = (a\hat{l}_1^T\Lambda, 0, 0, a\hat{l}_2^T\Lambda)$, so we should have

$$\hat{l}_1^T\Lambda = 0,$$

which is identical to (4.2.17). Therefore, the same boundary conditions should be imposed at these two points, and there will be no jump in adjoints on either side of these two points.

4.2.4 Adjoint boundary conditions on the surfaces Ω_1, Ω_2

From (4.2.14), we have

$$\Lambda(0,\eta) = \omega(\eta)(\frac{\partial\hat{F}}{\partial Q})^{-T}|_{\xi=0},$$

and

$$\Lambda(\xi_{max},\eta) = \omega(\eta)(\frac{\partial\hat{F}}{\partial Q})^{-T}|_{\xi=\xi_{max}}.$$

Since Q is periodic across the surface Ω_1 and Ω_2, the RHS of the above two expressions are the same, and

$$\Lambda(0,\eta) = \Lambda(\xi_{max},\eta). \tag{4.2.21}$$

4.2.5 Reduced gradients of functionals

The reduced gradient of functionals can be obtained by differentiating (4.2.9) with respect to the angle of attack α

$$\begin{aligned}\frac{\partial\mathcal{I}}{\partial\alpha} &= \int_{\mathcal{B}_\mathcal{W}}\frac{\partial I}{\partial\alpha}d\xi \\ &+ \frac{\partial}{\partial\alpha}\int_{\Gamma_1(\mathcal{G},\alpha)}(\nu^T P(Q-Q_\infty)+\Lambda^T\hat{G})d\xi \\ &+ \frac{\partial}{\partial\alpha}\int_{\Gamma_2(\mathcal{G},\alpha)}(\mu W(Q-Q_\infty)+\Lambda^T\hat{G})d\xi,\end{aligned} \tag{4.2.22}$$

and with respect to the geometric parameters $\mathcal{G}$

$$\begin{aligned}
\frac{\partial \mathcal{I}}{\partial \mathcal{G}} &= \int_{\mathcal{B}_\mathcal{W}} \big(\frac{\partial I}{\partial \mathcal{G}} - \Lambda^T \frac{\partial \hat{G}}{\partial \mathcal{G}} + \psi(u\frac{\partial S_{21}}{\partial \mathcal{G}} + v\frac{\partial S_{22}}{\partial \mathcal{G}})\big) d\xi \\
&- \int\int_{\mathcal{D}} (\frac{\partial \Lambda^T}{\partial \xi}\frac{\partial \hat{F}}{\partial \mathcal{G}} + \frac{\partial \Lambda^T}{\partial \eta}\frac{\partial \hat{G}}{\partial \mathcal{G}}) d\xi d\eta \\
&+ \frac{\partial}{\partial \mathcal{G}} \int_{\Gamma_1(\mathcal{G},\alpha)} (\nu^T P(Q - Q_\infty) + \Lambda^T \hat{G}) d\xi \\
&+ \frac{\partial}{\partial \mathcal{G}} \int_{\Gamma_2(\mathcal{G},\alpha)} (\mu W(Q - Q_\infty) + \Lambda^T \hat{G}) d\xi, \qquad (4.2.23)
\end{aligned}$$

where

$$\begin{aligned}
\frac{\partial \hat{F}}{\partial \mathcal{G}} &= \frac{\partial S_{11}}{\partial \mathcal{G}} F + \frac{\partial S_{12}}{\partial \mathcal{G}} G \\
\frac{\partial \hat{G}}{\partial \mathcal{G}} &= \frac{\partial S_{21}}{\partial \mathcal{G}} F + \frac{\partial S_{22}}{\partial \mathcal{G}} G.
\end{aligned}$$

Note that this differentiation operation only concerns *explicit* dependence on the design variables. The implicit dependence of states on the design variables has already been accounted for via adjoining the governing equations and boundary conditions to the cost functional to derive the adjoint equations. Therefore, when C_D is under consideration, we have

$$\begin{aligned}
\frac{\partial I}{\partial p} &= \frac{-1}{\frac{1}{2}\rho V_\infty^2 c}(S_{21}\cos\alpha + S_{22}\sin\alpha) \\
\frac{\partial I}{\partial \mathcal{G}} &= \frac{-p}{\frac{1}{2}\rho V_\infty^2 c}(\frac{\partial S_{21}}{\partial \mathcal{G}}\cos\alpha + \frac{\partial S_{22}}{\partial \mathcal{G}}\sin\alpha) \\
\frac{\partial I}{\partial \alpha} &= \frac{-p}{\frac{1}{2}\rho V_\infty^2 c}(-S_{21}\sin\alpha + S_{22}\cos\alpha),
\end{aligned}$$

and when C_L is under consideration, we have

$$\begin{aligned}
\frac{\partial I}{\partial p} &= \frac{1}{\frac{1}{2}\rho V_\infty^2 c}(-S_{22}\cos\alpha + S_{21}\sin\alpha) \\
\frac{\partial I}{\partial \mathcal{G}} &= \frac{p}{\frac{1}{2}\rho V_\infty^2 c}(-\frac{\partial S_{22}}{\partial \mathcal{G}}\cos\alpha + \frac{\partial S_{21}}{\partial \mathcal{G}}\sin\alpha) \\
\frac{\partial I}{\partial \alpha} &= \frac{p}{\frac{1}{2}\rho V_\infty^2 c}(S_{22}\sin\alpha + S_{21}\cos\alpha),
\end{aligned}$$

where c is chord length. Careful attention should be paid to the last two terms of (4.2.22) and (4.2.23), since the integration region itself depends on the geometric variables. Let

$\Gamma_1 = [\xi_s(\mathcal{G},\alpha), \xi_e(\mathcal{G},\alpha)]$, then we have,

$$\begin{aligned}
& \frac{\partial}{\partial\alpha}\int_{\Gamma_1(\mathcal{G},\alpha)} (\nu^T P(Q-Q_\infty) + \Lambda^T \hat{G}) d\xi \\
= & \int_{\xi_s(\mathcal{G},\alpha)}^{\xi_e(\mathcal{G},\alpha)} \frac{\partial}{\partial\alpha}(\nu^T P(Q-Q_\infty) + \Lambda^T \hat{G}) d\xi \\
& + \frac{\partial \xi_e(\mathcal{G},\alpha)}{\partial\alpha}\big(\nu^T P(Q-Q_\infty) + \Lambda^T \hat{G}\big)|_{\xi_e} \\
& - \frac{\partial \xi_s(\mathcal{G},\alpha)}{\partial\alpha}\big(\nu^T P(Q-Q_\infty) + \Lambda^T \hat{G}\big)|_{\xi_s} \\
= & \int_{\xi_s(\mathcal{G},\alpha)}^{\xi_e(\mathcal{G},\alpha)} -\nu^T P\frac{\partial Q_\infty}{\partial\alpha} d\xi \\
& + \frac{\partial \xi_e(\mathcal{G},\alpha)}{\partial\alpha}\big(\nu_-^T P(Q-Q_\infty) + \Lambda_-^T \hat{G}\big)|_{\xi_e} \\
& - \frac{\partial \xi_s(\mathcal{G},\alpha)}{\partial\alpha}\big(\nu_+^T P(Q-Q_\infty) + \Lambda_+^T \hat{G}\big)|_{\xi_s},
\end{aligned}$$

where $(\cdot)_-$ represents approaching from the left, and $(\cdot)_+$ from the right. Similarly, with $\Gamma_2 = [\xi_e(\mathcal{G},\alpha), \xi_s(\mathcal{G},\alpha)]$, we find

$$\begin{aligned}
& \frac{\partial}{\partial\alpha}\int_{\Gamma_2(\mathcal{G},\alpha)} (\mu W(Q-Q_\infty) + \Lambda^T \hat{G}) d\xi \\
= & \int_{\xi_e(\mathcal{G},\alpha)}^{\xi_s(\mathcal{G},\alpha)} \frac{\partial}{\partial\alpha}(\mu W(Q-Q_\infty) + \Lambda^T \hat{G}) d\xi \\
& + \frac{\partial \xi_s(\mathcal{G},\alpha)}{\partial\alpha}(\mu W(Q-Q_\infty) + \Lambda^T \hat{G})|_{\xi_s} \\
& - \frac{\partial \xi_e(\mathcal{G},\alpha)}{\partial\alpha}(\mu W(Q-Q_\infty) + \Lambda^T \hat{G})|_{\xi_e} \\
= & \int_{\xi_e(\mathcal{G},\alpha)}^{\xi_s(\mathcal{G},\alpha)} -\mu W\frac{\partial Q_\infty}{\partial\alpha} d\xi \\
& + \frac{\partial \xi_s(\mathcal{G},\alpha)}{\partial\alpha}(\mu_- W(Q-Q_\infty) + \Lambda_-^T \hat{G})|_{\xi_s} \\
& - \frac{\partial \xi_e(\mathcal{G},\alpha)}{\partial\alpha}(\mu_+ W(Q-Q_\infty) + \Lambda_+^T \hat{G})|_{\xi_e}.
\end{aligned} \tag{4.2.24}$$

Hence,

$$\begin{aligned}
& \frac{\partial}{\partial\alpha}\int_{\Gamma_1(\mathcal{G},\alpha)} (\nu^T P(Q-Q_\infty) + \Lambda^T \hat{G}) d\xi \\
+ & \frac{\partial}{\partial\alpha}\int_{\Gamma_2(\mathcal{G},\alpha)} (\mu W(Q-Q_\infty) + \Lambda^T \hat{G}) d\xi \\
= & \int_{\xi_s(\mathcal{G},\alpha)}^{\xi_e(\mathcal{G},\alpha)} -\nu^T P\frac{\partial Q_\infty}{\partial\alpha} d\xi + \int_{\xi_e(\mathcal{G},\alpha)}^{\xi_s(\mathcal{G},\alpha)} -\mu W\frac{\partial Q_\infty}{\partial\alpha} d\xi
\end{aligned}$$

$$+\frac{\partial \xi_e(\mathcal{G},\alpha)}{\partial \alpha}\big((\nu_-^T P-\mu_+ W)(Q-Q_\infty)+(\Lambda_-^T-\Lambda_+^T)\hat{G}\big)|_{\xi_e}$$
$$-\frac{\partial \xi_s(\mathcal{G},\alpha)}{\partial \alpha}\big((\nu_+^T P-\mu_- W)(Q-Q_\infty)+(\Lambda_+^T-\Lambda_-^T)\hat{G}\big)|_{\xi_s}. \tag{4.2.25}$$

At $\xi=\xi_e$ and $\xi=\xi_s$, $u_n=0$, then from (4.2.18) we have $\nu_1=0,\nu_2=0$ and

$$\nu_3 = a\sqrt{S_{21}^2+S_{22}^2}\hat{l}_2^T\Lambda,$$

and from (4.2.20), we have

$$\mu = a\sqrt{S_{21}^2+S_{22}^2}\hat{l}_2^T\Lambda.$$

Therefore,

$$\begin{aligned}
&\frac{\partial}{\partial\alpha}\int_{\Gamma_1(\mathcal{G},\alpha)}(\nu^T P(Q-Q_\infty)+\Lambda^T\hat{G})d\xi+\frac{\partial}{\partial\alpha}\int_{\Gamma_2(\mathcal{G},\alpha)}(\mu W(Q-Q_\infty)+\Lambda^T\hat{G})d\xi\\
=\;&\int_{\xi_s(\mathcal{G},\alpha)}^{\xi_e(\mathcal{G},\alpha)}-\nu^T P\frac{\partial Q_\infty}{\partial\alpha}d\xi+\int_{\xi_e(\mathcal{G},\alpha)}^{\xi_s(\mathcal{G},\alpha)}-\mu W\frac{\partial Q_\infty}{\partial\alpha}d\xi\\
&+\frac{\partial \xi_e(\mathcal{G},\alpha)}{\partial\alpha}\big(a\sqrt{S_{21}^2+S_{22}^2}\hat{l}_2^T(\Lambda_--\Lambda_+)W(Q-Q_\infty)+(\Lambda_-^T-\Lambda_+^T)\hat{G}\big)|_{\xi_e}\\
&+\frac{\partial \xi_s(\mathcal{G},\alpha)}{\partial\alpha}\big(a\sqrt{S_{21}^2+S_{22}^2}\hat{l}_2^T(\Lambda_--\Lambda_+)W(Q-Q_\infty)+(\Lambda_-^T-\Lambda_+^T)\hat{G}\big)|_{\xi_s}\\
=\;&\int_\Gamma\Lambda^T\frac{\partial\hat{G}}{\partial Q}\frac{\partial Q_\infty}{\partial\alpha}d\xi+\frac{\partial\xi_e(\mathcal{G},\alpha)}{\partial\alpha}\big(a\sqrt{S_{21}^2+S_{22}^2}\hat{l}_2^T(\Lambda_--\Lambda_+)W(Q-Q_\infty)+(\Lambda_-^T-\Lambda_+^T)\hat{G}\big)|_{\xi_e}\\
&+\frac{\partial \xi_s(\mathcal{G},\alpha)}{\partial\alpha}\big(a\sqrt{S_{21}^2+S_{22}^2}\hat{l}_2^T(\Lambda_--\Lambda_+)W(Q-Q_\infty)+(\Lambda_-^T-\Lambda_+^T)\hat{G}\big)|_{\xi_s}.
\end{aligned}$$

Similarly,

$$\begin{aligned}
&\frac{\partial}{\partial\mathcal{G}}\int_{\Gamma_1(\mathcal{G},\alpha)}(\nu^T P(Q-Q_\infty)+\Lambda^T\hat{G})d\xi+\frac{\partial}{\partial\mathcal{G}}\int_{\Gamma_2(\mathcal{G},\alpha)}(\mu W(Q-Q_\infty)+\Lambda^T\hat{G})d\xi\\
=\;&\int_{\xi_s(\mathcal{G},\alpha)}^{\xi_e(\mathcal{G},\alpha)}\frac{\partial}{\partial\mathcal{G}}(\nu^T P(Q-Q_\infty)+\Lambda^T\hat{G})d\xi+\int_{\xi_e(\mathcal{G},\alpha)}^{\xi_s(\mathcal{G},\alpha)}\frac{\partial}{\partial\mathcal{G}}(\mu W(Q-Q_\infty)+\Lambda^T\hat{G})d\xi\\
&+\frac{\partial \xi_e(\mathcal{G},\alpha)}{\partial\mathcal{G}}\big(a\sqrt{S_{21}^2+S_{22}^2}\hat{l}_2^T(\Lambda_--\Lambda_+)W(Q-Q_\infty)+(\Lambda_-^T-\Lambda_+^T)\hat{G}\big)|_{\xi_e}\\
&+\frac{\partial \xi_s(\mathcal{G},\alpha)}{\partial\mathcal{G}}\big(a\sqrt{S_{21}^2+S_{22}^2}\hat{l}_2^T(\Lambda_--\Lambda_+)W(Q-Q_\infty)+(\Lambda_-^T-\Lambda_+^T)\hat{G}\big)|_{\xi_s}\\
=\;&\int_\Gamma\Lambda^T\frac{\partial\hat{G}}{\partial\mathcal{G}}d\xi+\frac{\partial\xi_e(\mathcal{G},\alpha)}{\partial\mathcal{G}}\big(a\sqrt{S_{21}^2+S_{22}^2}\hat{l}_2^T(\Lambda_--\Lambda_+)W(Q-Q_\infty)+(\Lambda_-^T-\Lambda_+^T)\hat{G}\big)|_{\xi_e}\\
&+\frac{\partial \xi_s(\mathcal{G},\alpha)}{\partial\mathcal{G}}\big(a\sqrt{S_{21}^2+S_{22}^2}\hat{l}_2^T(\Lambda_--\Lambda_+)W(Q-Q_\infty)+(\Lambda_-^T-\Lambda_+^T)\hat{G}\big)|_{\xi_s}.
\end{aligned}$$

Since the adjoints are continuous at ξ_e and ξ_s where $u_n = 0$, the gradients of a functional with respect to the angle of attack and the geometric parameters are given by,

$$\frac{\partial \mathcal{I}}{\partial \alpha} = \int_{\mathcal{B}_\mathcal{W}} \frac{\partial I}{\partial \alpha} d\xi + \int_\Gamma \Lambda^T \frac{\partial \hat{G}}{\partial Q} \frac{\partial Q_\infty}{\partial \alpha} d\xi \tag{4.2.26}$$

and

$$\begin{aligned} \frac{\partial \mathcal{I}}{\partial \mathcal{G}} &= \int_{\mathcal{B}_\mathcal{W}} \big(\frac{\partial I}{\partial \mathcal{G}} - \Lambda^T \frac{\partial \hat{G}}{\partial \mathcal{G}} + \phi(u \frac{\partial S_{21}}{\partial \mathcal{G}} + v \frac{\partial S_{22}}{\partial \mathcal{G}}) \big) d\xi \\ &\quad - \int \int_\mathcal{D} (\frac{\partial \Lambda^T}{\partial \xi} \frac{\partial \hat{F}}{\partial \mathcal{G}} + \frac{\partial \Lambda^T}{\partial \eta} \frac{\partial \hat{G}}{\partial \mathcal{G}}) d\xi d\eta + \int_\Gamma \Lambda^T \frac{\partial \hat{G}}{\partial \mathcal{G}} d\xi \end{aligned} \tag{4.2.27}$$

respectively. The first term of the RHS of (4.2.26) is the integration on the airfoil at the *frozen* states. The effect of changing states is mapped onto the outer boundary, which is accounted for by the second term. The gradient with respect to the geometric parameters consists of three parts, the integration on the airfoil, on the far-field boundary and on the flow domain, since a perturbation in the airfoil shape will lead to perturbations in the whole grid system. Therefore, we can calculate reduced gradients via adjoint methods. These gradients, as well as the values of cost/constraint functionals, can be provided to a numerical optimizer for the purpose of optimization design.

In Jameson's approach [72, 73], the adjoint variables were set to zero, identically, at the far-field boundary. Even though the vanishing adjoint variables satisfy the boundary conditions (4.2.17) and (4.2.19), they are over-specified in this case. Furthermore, this treatment leads to disappearance of the last term in (4.2.27). As a compensation, a specific grid system as described in [50] was employed, in which the grid lines at the far-field boundary remained unchanged during the optimization procedures. In this situation, $\frac{\partial \hat{G}}{\partial \mathcal{G}} = 0$ and the last term in (4.2.27) can be neglected.

Before we develop the numerical methods to solve the adjoint equations discussed above, it is worthwhile to explore the super-reduced optimization problem (2.5.9).

4.3 Super-Reduced Design Structure

In the problem of minimizing C_D at fixed C_L, the lift constraint is just a scalar equation. This special structure can be exploited and leads to the discussions in this section. The angle of attack is treated as an additional state variable and the lift constraint is an additional state equation. In this way, the constrained optimization problem (2.5.8) reduces to a super-reduced problem (2.5.9). In this super-reduced problem, the gradient to be supplied to an optimizer is the geometric gradient of C_D subject to the lift constraint, and is termed as the *super-reduced* gradient. In Section 4.3.1, the adjoint equations with non-local boundary conditions are derived to obtain the super-reduced gradient of C_D. Another approach, using reduced gradients obtained in Section 4.2 to obtain super-reduced gradients, is introduced in Section 4.3.2.

4.3.1 Adjoint system for super-reduced problem

The new states, including flow variables and the angle of attack, can be solved from the Euler equations and the lift constraint for given geometric design variables $\mathcal{G}$; these constraints can be viewed as the constraint H in the Lagrange Multiplier Theorem (A.3.1). Therefore, besides the Euler equations and the boundary conditions, the aerodynamic constraint $C_l - C_{l_0} = 0$ should be adjoined to the augmented Lagrangian functional, which now takes the following form,

$$\begin{aligned}
\mathcal{I}(Q,\alpha;\mathcal{G};\mathcal{L},\sigma) &= \int_{\mathcal{B}_\mathcal{W}} I_D(\xi;p(Q),\alpha;\mathcal{G})d\xi \\
&+ \int\int_{\mathcal{D}} \Lambda^T(\frac{\partial \hat{F}}{\partial \xi}+\frac{\partial \hat{G}}{\partial \eta})d\xi d\eta \\
&+ \int_{\mathcal{B}_\mathcal{W}} \psi(S_{21}u+S_{22}v)d\xi \\
&+ \int_{\Gamma_1(\mathcal{G},\alpha)} \nu^T P(Q-Q_\infty(\alpha))d\xi \\
&+ \int_{\Gamma_2(\mathcal{G},\alpha)} \mu W(Q-Q_\infty(\alpha))d\xi
\end{aligned}$$

$$
\begin{aligned}
&+ \int_{\eta_1}^{\eta_{max}} \omega^T (Q|_{\Omega_1} - Q|_{\Omega_2}) d\eta \\
&+ \sigma(\int_{\mathcal{B}_\mathcal{W}} I_L(\xi; p(Q), \alpha; \mathcal{G}) d\xi - C_{L_0}),
\end{aligned}
$$

where

$$
\begin{aligned}
I_D &= \frac{-p}{\frac{1}{2}\rho V_\infty^2 c}(S_{21}\cos\alpha + S_{22}\sin\alpha) \\
I_L &= \frac{-p}{\frac{1}{2}\rho V_\infty^2 c}(-S_{21}\sin\alpha + S_{22}\cos\alpha).
\end{aligned}
$$

Now differentiating the augmented Lagrangian functional with respect to the states not only includes differentiating it with respect to the flow variables Q, but also with respect to the angle of attack. Following the same procedures outlined in the Section 4.2, we arrive at,

$$
\begin{aligned}
\frac{\partial \mathcal{I}}{\partial Q} &= \int_{\mathcal{B}_\mathcal{W}} \big(\frac{\partial (I_D + \sigma I_L)}{\partial p}\frac{\partial p}{\partial Q} - \Lambda^T \frac{\partial \hat{G}}{\partial Q} + \psi(S_{21}\frac{\partial u}{\partial Q} + S_{22}\frac{\partial v}{\partial Q})\big) d\xi \\
&- \int\int_{\mathcal{D}} (\frac{\partial \Lambda^T}{\partial \xi}\frac{\partial \hat{F}}{\partial Q} + \frac{\partial \Lambda^T}{\partial \eta}\frac{\partial \hat{G}}{\partial Q}) d\xi d\eta \\
&+ \int_{\Gamma_1(\mathcal{G},\alpha)} (\nu^T P + \Lambda^T \frac{\partial \hat{G}}{\partial Q}) d\xi \\
&+ \int_{\Gamma_2(\mathcal{G},\alpha)} (\mu W + \Lambda^T \frac{\partial \hat{G}}{\partial Q}) d\xi \\
&+ \int_{\Omega_1} (\omega^T - \Lambda^T \frac{\partial \hat{F}}{\partial Q}) d\eta + \int_{\Omega_2} (-\omega^T + \Lambda^T \frac{\partial \hat{F}}{\partial Q}) d\eta. \qquad (4.3.28)
\end{aligned}
$$

and

$$
\begin{aligned}
\frac{\partial \mathcal{I}}{\partial \alpha} &= \int_{\mathcal{B}_\mathcal{W}} \frac{\partial (I_D + \sigma I_L)}{\partial \alpha} d\xi + \int_{\Gamma_1(\mathcal{G},\alpha)} -\nu^T P \frac{\partial Q_\infty(\alpha)}{\partial \alpha} d\xi + \int_{\Gamma_2(\mathcal{G},\alpha)} -\mu W \frac{Q_\infty(\alpha)}{\partial \alpha} d\xi \\
&= C_L - \sigma C_D + \int_{\Gamma_1(\mathcal{G},\alpha)} -\nu^T P \frac{\partial Q_\infty(\alpha)}{\partial \alpha} d\xi + \int_{\Gamma_2(\mathcal{G},\alpha)} -\mu W \frac{Q_\infty(\alpha)}{\partial \alpha} d\xi. \qquad (4.3.29)
\end{aligned}
$$

In the previous treatment in Section 4.2, α was a control variable and we used the requirement that $\frac{\partial \mathcal{I}}{\partial Q}$ in (4.3.28) vanish, along with $\sigma = 0$ to arrive at the adjoint boundary-value problem. In the present study, we add the requirement that (4.3.29) vanish and also solve for the additional scalar multiplier, σ.

The PDE and the far-field boundary conditions for the distributed adjoint $\Lambda(\xi, \eta)$ are the same as (4.2.10) and (4.2.12, 4.2.13) in the previous case. On the airfoil the boundary

condition (4.2.11) is replaced with

$$\frac{\partial(I_D+\sigma I_L)}{\partial p}\frac{\partial p}{\partial Q}-\Lambda^T\frac{\partial \hat{G}}{\partial Q}+\psi(S_{21}\frac{\partial u}{\partial Q}+S_{22}\frac{\partial v}{\partial Q})=0. \tag{4.3.30}$$

The scalar Lagrange multiplier σ in the above expression can be obtained from (4.3.29),

$$\sigma=\frac{1}{C_D}(C_L+\int_{\Gamma_1+\Gamma_2}\Lambda^T\frac{\partial \hat{G}}{\partial Q}\frac{\partial Q_\infty}{\partial \alpha}d\xi). \tag{4.3.31}$$

Note that since the scalar Lagrange multiplier σ depends on values of the distributed adjoint (Λ) integrated on the far-field, the boundary condition (4.3.30) is non-local. After solving this non-local boundary value problem, the super-reduced gradient of cost functional is given by,

$$\begin{aligned}\frac{\partial \mathcal{I}}{\partial \mathcal{G}} &= \int_{\mathcal{B}_\mathcal{W}}\big(\frac{\partial(I_D+\sigma I_L)}{\partial \mathcal{G}}-\Lambda^T\frac{\partial \hat{G}}{\partial \mathcal{G}}+\phi(u\frac{\partial S_{21}}{\partial \mathcal{G}}+v\frac{\partial S_{22}}{\partial \mathcal{G}})\big)d\xi \\ &\quad -\int\int_{\mathcal{D}}(\frac{\partial \Lambda^T}{\partial \xi}\frac{\partial \hat{F}}{\partial \mathcal{G}}+\frac{\partial \Lambda^T}{\partial \eta}\frac{\partial \hat{G}}{\partial \mathcal{G}})d\xi d\eta+\int_\Gamma\Lambda^T\frac{\partial \hat{G}}{\partial \mathcal{G}}d\xi.\end{aligned} \tag{4.3.32}$$

A Gauss-Seidel iterative method can be employed to solve this adjoint system. First a guess for the scalar Lagrange multiplier σ is substituted into boundary condition (4.3.30). The resulting *BVP* for adjoints involves only local data and is numerically solved. Then σ is calculated from (4.3.31) with adjoints supplied from the numerical adjoint solutions. This σ value is again substituted into (4.3.30), and so on, until convergence. In this method, the adjoint system must be solved many times [4] to find the super-reduced geometric gradients. For this reason, we do not adopt this method in the present studies.

At the same time, the state α can be obtained from the aerodynamic constraint. To this purpose, we first linearize the constraint about the current parameters to arrive at,

$$C_{L_c}+\frac{\partial C_L}{\partial Q}\frac{\partial Q}{\partial \mathcal{G}}\Delta\mathcal{G}+\frac{\partial C_L}{\partial \mathcal{G}}\Delta\mathcal{G}+\frac{\partial C_L}{\partial Q}\frac{\partial Q}{\partial \alpha}\Delta\alpha+\frac{\partial C_L}{\partial \alpha}\Delta\alpha-C_{L_0}=0$$

where $\frac{\partial C_L}{\partial \alpha}$ involves only the explicit dependence of C_L on α, so $\frac{\partial C_L}{\partial \alpha}=-C_D$. Finally we get,

$$\Delta\alpha=\frac{(C_{L_0}-C_{L_c})-(\frac{\partial C_L}{\partial Q}\frac{\partial Q}{\partial \mathcal{G}}\Delta\mathcal{G}+\frac{\partial C_L}{\partial \mathcal{G}}\Delta\mathcal{G})}{\frac{\partial C_L}{\partial Q}\frac{\partial Q}{\partial \alpha}-C_D}.$$

[4] In one typical optimization iteration for the reduced problem (2.5.8), the adjoint equations should be solved twice, one associated with C_D and the other with C_L.

4.3.2 Direct Lagrange multiplier approach

In this approach, the Lagrange Multiplier Theorem is directly applied to the augmented Lagrangian functional

$$\mathcal{I} = C_D + \sigma(C_L - C_{L_0})$$

to obtain

$$\frac{\partial \mathcal{I}}{\partial \alpha} = \frac{\partial C_D}{\partial \alpha}|_{\mathcal{G}} + \sigma \frac{\partial C_L}{\partial \alpha}|_{\mathcal{G}} = 0 \tag{4.3.33}$$

and

$$\frac{\partial \mathcal{I}}{\partial \mathcal{G}}|_{C_L} = \frac{\partial C_D}{\partial \mathcal{G}}|_{\alpha} + \sigma \frac{\partial C_L}{\partial \mathcal{G}}|_{\alpha}, \tag{4.3.34}$$

where $|_{(\cdot)}$ denotes the partial derivatives at the frozen quantity indicated by $(\cdot)$. From (4.3.33), the Lagrange multiplier σ can be obtain as

$$\sigma = -\frac{\frac{\partial C_D}{\partial \alpha}|_{\mathcal{G}}}{\frac{\partial C_L}{\partial \alpha}|_{\mathcal{G}}}.$$

Substituting the above formulation to (4.3.34) yields,

$$\frac{\partial \mathcal{I}}{\partial \mathcal{G}}|_{C_L} = \frac{\partial C_D}{\partial \mathcal{G}}|_{\alpha} - \frac{\frac{\partial C_D}{\partial \alpha}|_{\mathcal{G}}}{\frac{\partial C_L}{\partial \alpha}|_{\mathcal{G}}} \frac{\partial C_L}{\partial \mathcal{G}}|_{\alpha}. \tag{4.3.35}$$

The gradients at frozen $\mathcal{G}$ and α have already been obtained in the Section 4.2; thus, the super-reduced geometric gradients of cost functional C_D at frozen C_L (subject to the lift constraint) can be constructed in this way.

In Jameson's approach, the drag coefficient and lift coefficient are first expressed in terms of the normal force coefficient C_N and the axial force coefficient C_A,

$$C_D(\alpha, \mathcal{G}) = C_N(\alpha, \mathcal{G}) \sin\alpha + C_A(\alpha, \mathcal{G}) \cos\alpha$$
$$C_L(\alpha, \mathcal{G}) = C_N(\alpha, \mathcal{G}) \cos\alpha - C_A(\alpha, \mathcal{G}) \sin\alpha.$$

The notation here obviously implies that the flow variables Q have been eliminated by solving the Euler flow equations. The first variation for the C_D is

$$\delta C_D = (\frac{\partial C_N}{\partial \mathcal{G}} \sin\alpha + \frac{\partial C_A}{\partial \mathcal{G}} \cos\alpha)\Delta\mathcal{G} + (\frac{\partial C_N}{\partial \alpha} \sin\alpha + \frac{\partial C_A}{\partial \alpha} \cos\alpha)\Delta\alpha + (C_N \cos\alpha - C_A \sin\alpha)\Delta\alpha. \tag{4.3.36}$$

In the above expression, changes in geometric design variables, $\Delta\mathcal{G}$ and changes in angle of attack, $\Delta\alpha$ are independent. The first part of δC_D, $(\frac{\partial C_N}{\partial \mathcal{G}}\sin\alpha+\frac{\partial C_A}{\partial \mathcal{G}}\cos\alpha)\Delta\mathcal{G}$ accounts for the variation due to geometric changes at fixed α; the second and third term in (4.3.36) together account for the variation of C_D due to the change of angle of attack at fixed geometric parameters. These variations (or gradients) can be obtained through the adjoint methods outlined in Section 4.2. The lift constraint is linearized to obtain,

$$\begin{aligned} C_{L_c} &+ (\frac{\partial C_N}{\partial \mathcal{G}}\cos\alpha - \frac{\partial C_A}{\partial \mathcal{G}}\sin\alpha)\Delta\mathcal{G} + (\frac{\partial C_N}{\partial \alpha}\cos\alpha - \frac{\partial C_A}{\partial \alpha}\sin\alpha)\Delta\alpha \\ &- (C_N\sin\alpha + C_A\cos\alpha)\Delta\alpha - C_{L_0} = 0, \end{aligned} \tag{4.3.37}$$

from which $\Delta\alpha$ can be solved

$$\Delta\alpha = \frac{(C_{L_0} - C_{L_c}) - (\frac{\partial C_N}{\partial \mathcal{G}}\cos\alpha - \frac{\partial C_A}{\partial \mathcal{G}}\sin\alpha)\Delta\mathcal{G}}{-C_D + \frac{\partial C_N}{\partial \alpha}\cos\alpha - \frac{\partial C_A}{\partial \alpha}\sin\alpha}. \tag{4.3.38}$$

This formulation is identical to what has been obtained in the previous section. Substituting the above formulation for $\Delta\alpha$ in (4.3.36) yields,

$$\begin{aligned} \delta C_D &= (\frac{\partial C_N}{\partial \mathcal{G}}\sin\alpha + \frac{\partial C_A}{\partial \mathcal{G}}\cos\alpha)\Delta\mathcal{G} \\ &+ \left[\frac{C_L + \frac{\partial C_N}{\partial \alpha}\sin\alpha + \frac{\partial C_A}{\partial \alpha}\cos\alpha}{-C_D + \frac{\partial C_N}{\partial \alpha}\cos\alpha - \frac{\partial C_A}{\partial \alpha}\sin\alpha}\right]\big(C_{L_0} - C_{L_c} - (\frac{\partial C_N}{\partial \mathcal{G}}\cos\alpha - \frac{\partial C_A}{\partial \mathcal{G}}\sin\alpha)\Delta\mathcal{G}\big). \end{aligned}$$

If the design is started from $C_L = C_{L_0}$, i.e., the initial point is feasible , we will have $C_{L_c} = C_{L_0}$; thus

$$\frac{\partial C_D}{\partial \mathcal{G}} = (\frac{\partial C_N}{\partial \mathcal{G}}\sin\alpha + \frac{\partial C_A}{\partial \mathcal{G}}\cos\alpha) - \Omega(\frac{\partial C_N}{\partial \mathcal{G}}\cos\alpha - \frac{\partial C_A}{\partial \mathcal{G}}\sin\alpha)$$

where

$$\Omega = \frac{C_L + \frac{\partial C_N}{\partial \alpha}\sin\alpha + \frac{\partial C_A}{\partial \alpha}\cos\alpha}{-C_D + \frac{\partial C_N}{\partial \alpha}\cos\alpha - \frac{\partial C_A}{\partial \alpha}\sin\alpha}.$$

By recognizing

$$\begin{aligned} \frac{\partial C_D}{\partial \mathcal{G}}|_\alpha &= (\frac{\partial C_N}{\partial \mathcal{G}}\sin\alpha + \frac{\partial C_A}{\partial \mathcal{G}}\cos\alpha) \\ \frac{\partial C_L}{\partial \mathcal{G}}|_\alpha &= (\frac{\partial C_N}{\partial \mathcal{G}}\cos\alpha - \frac{\partial C_A}{\partial \mathcal{G}}\sin\alpha) \\ \frac{\partial C_D}{\partial \alpha}|_\mathcal{G} &= C_L + \frac{\partial C_N}{\partial \alpha}\sin\alpha + \frac{\partial C_A}{\partial \alpha}\cos\alpha \\ \frac{\partial C_L}{\partial \alpha}|_\mathcal{G} &= -C_D + \frac{\partial C_N}{\partial \alpha}\cos\alpha - \frac{\partial C_A}{\partial \alpha}\sin\alpha, \end{aligned} \tag{4.3.39}$$

we will arrive at the expression (4.3.35) for $\frac{\partial C_D}{\partial \mathcal{G}}|_{C_L}$. Again, this reduced gradient is the gradient of drag coefficient with respect to the geometric design parameters subject to the lift constraint, and is called super-reduced gradient. Therefore, we can construct the geometric C_D gradient at fixed C_L via geometric gradients of C_D and C_L at fixed α, and gradients of C_D and C_L with respect to α at fixed geometric parameters; these gradients have already obtained in Section 4.2. In Jameson and Reuther's approach [73], a finite difference method is employed to estimate Ω; and the Euler equations are augmented to C_D to derive the adjoint equations with boundary conditions on the airfoil affected by Ω.

Up to now, we have discussed the optimality structure for the reduced problem and super-reduced problem. In the reduced problem, the gradients of cost/constraint functionals with respect to the design parameters are obtained from adjoint methods and then supplied to the constrained optimizer. In the super-reduced problem, the constrained optimization problem reduces to an unconstrained problem which requires super-reduced gradients of cost functional subject to the lift constraint. Such super-reduced gradients can be constructed either by solving new adjoint equations with non-local boundary conditions or by a direct Lagrange multiplier approach.

4.4 Numerical Solutions of Adjoint Systems

Just as the unsteady Euler equation is solved to find the asymptotic solution to the steady equation, the steady adjoint equation (4.2.10) is augmented to an unsteady hyperbolic equations to be numerically solved via marching in the time-direction. When augmenting an unsteady term to the adjoint equations, we should note that the characteristic adjoint boundary conditions demonstrate a reverse characteristic property when compared to the boundary conditions for the flow variables. On the far-field boundary Γ_1 only one characteristic is outgoing ($u_n + a > 0$), so only one boundary condition can be imposed on the flow solver in that part of boundary. However, in the same segment of boundary, we get three boundary

conditions for the adjoints, which means only one characteristic is incoming. Similar phenomenon are also found in the boundary Γ_2 where there is one characteristic incoming for flow variables and one characteristic outgoing for adjoints. Based on this reverse property of adjoints, the augmented unsteady equations for adjoints can be formed as

$$\frac{1}{J}\frac{\partial \Lambda}{\partial t} - (\frac{\partial \hat{F}}{\partial Q})^T \frac{\partial \Lambda}{\partial \xi} - (\frac{\partial \hat{G}}{\partial Q})^T \frac{\partial \Lambda}{\partial \eta} = \mathbf{0}. \tag{4.4.40}$$

Jameson's finite volume scheme [73] is adapted to solve the adjoint equations numerically. In principle, other schemes, such as Steger-Warming FVS approach, could be utilized. Jameson's scheme was chosen for ease of numerical implementations.

In this numerical approach, a second order central difference scheme is employed in discretizing the spatial derivatives and a 5-step Runge-Kutta scheme is applied for the time integration. Furthermore, fourth order dissipation is introduced to provide background damping to prevent odd-even oscillations in the smooth region. Since this background dissipation is not sufficient to smear undesired oscillations around regions with large flow gradients, second order dissipation is introduced around the shock region via a switch devised to capture the large flow gradients. Hence the semi-discretized form is

$$\frac{1}{J}(\frac{\partial \Lambda}{\partial t})_{j,k} = \mathbf{Q}_{j,k} + \mathbf{D}(\Lambda_{j,k})$$

where $\mathbf{Q}_{j,k}$ is given by

$$\begin{aligned} \mathbf{Q}_{j,k} &= \big(S_{11}(\frac{\partial \hat{F}}{\partial Q})^T + S_{12}(\frac{\partial \hat{G}}{\partial Q})^T\big)_{j,k} \frac{\Lambda_{j+1,k} - \Lambda_{j-1,k}}{2} \\ &+ \big(S_{21}(\frac{\partial \hat{F}}{\partial Q})^T + S_{22}(\frac{\partial \hat{G}}{\partial Q})^T\big)_{j,k} \frac{\Lambda_{j,k+1} - \Lambda_{j,k-1}}{2} \end{aligned} \tag{4.4.41}$$

and the artificial dissipations are composed of the second order dissipation D^2 and the fourth order dissipation D^4

$$\mathbf{D}_{j,k} = D^2_{j,k} - D^4_{j,k}$$

with

$$D^{(\cdot)}_{j,k} = d^{(\cdot)}_{j+\frac{1}{2},k} - d^{(\cdot)}_{j-\frac{1}{2},k} + d^{(\cdot)}_{j,k+\frac{1}{2}} - d^{(\cdot)}_{j,k-\frac{1}{2}}$$

and the first order and third order dissipation flux in ξ-direction are given by

$$d^2_{j+\frac{1}{2},k}=\varepsilon^2_{j+\frac{1}{2},k}(\Lambda_{j+1,k}-\Lambda_{j,k})$$
$$d^4_{j+\frac{1}{2},k}=\varepsilon^4_{j+\frac{1}{2},k}(\Lambda_{j+2,k}-3\Lambda_{j+1,k}+3\Lambda_{j,k}-\Lambda_{j-1,k}).$$

The dissipation in the η direction can be formulated in a similar way. The dissipation coefficient ε^2 behaves as a switch which triggers the second order dissipation in regions with large gradients. To this purpose, a switch function is devised to capture the shock by taking the second order difference of pressure, and is defined as

$$\nu^j_{j,k}=\left|\frac{p_{j+1,k}-2p_{j,k}+p_{j-1,k}}{p_{j+1,k}+2p_{j,k}+p_{j-1,k}}\right|$$

The dissipation coefficients are set as

$$\begin{aligned}\varepsilon^2_{j+\frac{1}{2},k} &= \frac{\mu^2 V_{j,k}}{\Delta t_{j,k}}\max(\nu^j_{j+2,k},\nu^j_{j+1,k},\nu^j_{j,k},\nu^j_{j-1,k})\\ \varepsilon^4_{j+\frac{1}{2},k} &= \max(0,[\frac{\mu^4 V_{j,k}}{\Delta t_{j,k}}-\varepsilon^2_{j+\frac{1}{2},k}]),\end{aligned}$$

where the local time step $\Delta t_{j,k}$ will be defined later, and $V_{j,k}$ is the volume of the grid cell. The dissipation coefficients μ^2, μ^4 are user-supplied parameters; [5] and dissipative terms in the other direction can be readily obtained by following the same procedures. The following modified 5-stage Runge-Kutta scheme is employed for the time integration,

$$\begin{aligned}\Lambda^0_{j,k} &= \Lambda^n_{j,k}\\ \Lambda^1_{j,k} &= \Lambda^0_{j,k}+\frac{1}{4}\frac{\Delta t_{j,k}}{V_{j,k}}(\mathbf{Q}(\Lambda^0_{j,k})+\mathbf{D}(\Lambda^0_{j,k}))\\ \Lambda^2_{j,k} &= \Lambda^0_{j,k}+\frac{1}{6}\frac{\Delta t_{j,k}}{V_{j,k}}(\mathbf{Q}(\Lambda^1_{j,k})+\mathbf{D}(\Lambda^0_{j,k}))\\ \Lambda^3_{j,k} &= \Lambda^0_{j,k}+\frac{3}{8}\frac{\Delta t_{j,k}}{V_{j,k}}(\mathbf{Q}(\Lambda^2_{j,k})+[.44\mathbf{D}(\Lambda^0_{j,k})+.56\mathbf{D}(\Lambda^2_{j,k})])\\ \Lambda^4_{j,k} &= \Lambda^0_{j,k}+\frac{1}{2}\frac{\Delta t_{j,k}}{V_{j,k}}(\mathbf{Q}(\Lambda^3_{j,k})+[.44\mathbf{D}(\Lambda^0_{j,k})+.56\mathbf{D}(\Lambda^2_{j,k})])\end{aligned}$$

[5] One of advantages of this scheme lies in the fact that the dissipation is not inherent in the scheme, instead it is controlled by the user-supplied parameters, leaving room for improving the accuracy. However this may mean a burden to find proper parameters for the artificial dissipation.

$$\begin{aligned}\Lambda^5_{j,k} &= \Lambda^0_{j,k} + \frac{\Delta t_{j,k}}{V_{j,k}}(\mathbf{Q}(\Lambda^4_{j,k}) + (.56[.44\mathbf{D}(\Lambda^0_{j,k}) + .56\mathbf{D}(\Lambda^2_{j,k})] + .44\mathbf{D}(\Lambda^4_{j,k})))\\ \Lambda^{n+1}_{j,k} &= \Lambda^5_{j,k}\end{aligned} \tag{4.4.42}$$

where the artificial dissipation is calculated at every other step and the weight coefficients for the dissipation were suggested by Martinelli [73]. To accelerate convergence a local time step is utilized and given by characteristic analysis,

$$\Delta t_{j,k} = V_{j,k}\big(\,|uS_{11} + vS_{12}| + |uS_{21} + vS_{22}| + a(\sqrt{S^2_{11} + S^2_{12}} + \sqrt{S^2_{21} + S^2_{22}})\big)^{-1}_{j,k}$$

where a is the local sound speed. Therefore, the transient solution is not time-accurate. Furthermore, implicit residual smoothing is also adopted in every other Runge-Kutta time step to accelerate convergence; i.e., the residual $W_{j,k}$ of the RHS of (4.4.42) is replaced by $\bar{W}_{j,k}$ given by

$$(1 - \varepsilon_\xi \delta_\xi)(1 - \varepsilon_\eta \delta_\eta)\bar{W}_{j,k} = W_{j,k}$$

where $\varepsilon_\xi, \varepsilon_\eta$ are smoothing coefficients in ξ, η directions, respectively, and δ denotes the central difference operator. The smoothed residual collects residual information from all points over the computational domain with a weighting coefficient scaled by the distance from the point of interests. Jameson [45, 42] has shown that the effect of residual smoothing is to stabilize the computations even when a large CFL number is adopted. Application of this technique involves successively solving two tri-diagonal systems. The Thomas algorithm [39] is employed in the present studies.

In a numerical method, the adjoint boundary conditions should be put in a convenient mathematical formulation. The boundary condition on the airfoil (4.2.15) only involves the second and the third components of adjoint variables. To avoid overspecification of the boundary condition on this surface, it can be formulated as follows

$$\begin{aligned}\Lambda_{1_{(j,0)}} &= \Lambda_{1_{(j,1)}}\\ \Lambda_{2_{(j,0)}} &= \Lambda_{2_{(j,1)}} + 2n_1[\frac{\partial I}{\partial p} - n_1\Lambda_{2_{(j,1)}} - n_2\Lambda_{3_{(j,1)}}]\\ \Lambda_{3_{(j,0)}} &= \Lambda_{3_{(j,1)}} + 2n_2[\frac{\partial I}{\partial p} - n_1\Lambda_{2_{(j,1)}} - n_2\Lambda_{3_{(j,1)}}]\end{aligned}$$

$$\Lambda_{4_{(j,0)}} = \Lambda_{4_{(j,1)}} \tag{4.4.43}$$

where subscript $(j,0)$ and $(j,1)$ represent the cells below and above the airfoil surface, respectively. The same method can be applied to the boundary condition (4.2.17) on the far-field boundary Γ_1 to give

$$\Lambda^- = \Lambda^+ - 2(\Lambda^+ \cdot \hat{l}_1)\hat{l}_1,$$

where Λ^+, Λ^- are the adjoints inside and outside the far-field boundary. For boundary condition (4.2.19) on the far-field boundary Γ_2, we set

$$\Lambda^- = -\Lambda^+ + 2\|\Lambda^+\|\hat{l}_2.$$

4.5 Hyperbolic Grid Generation

In the present studies, we adopt the same structured grid system in both the CFD flow solver and the adjoint solver. The derivatives of metric terms with respect to geometric design variables are calculated by finite differentiating two grid systems with slightly perturbed geometric parameters. Therefore, in the optimization procedures the grid generation code is invoked many times, which necessitates an efficient grid generator for the 2-D airfoil.

Some discussions about advantages and disadvantages of several kinds of grid generation methods can be found in the review articles [85, 88]. Although straightforward algebraic grid generation is a candidate, this method demands much experience in carefully choosing the distribution functions to achieve satisfactory grid density and smoothness, and can not guarantee the orthogonality of grid lines which is important in reducing truncation errors.

A popular approach is to numerically solve the $PDEs$ governing the transformation from the computational domain to the physical domain. According to the characteristics of these $PDEs$, the method may be classified into 3 types: elliptic, parabolic and hyperbolic grid generations. The elliptic grid generation dates back to Thompson [88], and properly is the

most popular technique in grid generations. The grid systems generated in this way are smooth even with some loss of orthogonality, especially in the vicinity of the boundaries. Moreover, in this method the grid density can be controlled by source terms. However, an iterative procedure is needed to solve elliptic $PDEs$ which is much more time-consuming than solving parabolic or hyperbolic equations by a marching method in a time-like direction.

Hyperbolic grid generation has been studied by Starius [81], Steger [82] and Alsalihi [3] and further developed by Chan & Steger [14] and Matsuno [58]. In this method, the governing equations are obtained by imposing the orthogonality condition to the grid lines and by specifying the cell areas to control the grid density. The hyperbolic governing equations are marched in the time-like direction of η with given initial condition (grids on the line $\eta = 0$), so the grid points on the outer boundaries can not be specified. This drawback limits its use to external flow problems with free outer boundaries. However, this shortcoming can be overcome by parabolic grid generation which reduces the hyperbolic governing equations (4.5.46) to the parabolic ones by replacing $(\cdot)_\eta$ by a difference integrated with information from the outer boundary. Nakamura [61] and Edwards [22] successfully employed this approach to generate grid system around 3-D fuselage-wing structure. Takanashi & Takeoto [86] also generated grid systems around complex aircraft configurations by solving parabolic equations. Since this approach can not guarantee the grid orthogonality, especially in the vicinity of the inner boundary, the hyperbolic and parabolic grid generations were combined to take advantages from both methods while avoiding their disadvantages. Nakamura & Suzuki [62] outlined this hybrid scheme and demonstrated its advantages over strictly hyperbolic and strictly parabolic grid generation methods.

In Jameson's adjoint method, it is required that the outer boundary remains unchanged while modifying the shape of airfoil [59] or the shape of wing [71]. However this limitation does not arise in the present study, since we have carefully included the outer boundary conditions in our adjoint formulation. In the present study, the outer boundary can be free. Based on the above observations, the hyperbolic grid generation method is employed here to provide fast grid generation.

In hyperbolic grid generation, the first governing equation comes from specifying the cell areas which can control the grid density,

$$x_\xi y_\eta - y_\xi x_\eta = V(\xi, \eta). \tag{4.5.44}$$

The second equation can be derived from the orthogonality of grid lines. Along $\xi = const.$ contour, we have

$$d\xi = \xi_x dx + \xi_y dy = 0,$$

which means

$$\frac{dy}{dx}|_{\xi=const.} = -\frac{\xi_x}{\xi_y} = \frac{y_\eta}{x_\eta}.$$

Similarly on line $\eta = const.$, we have

$$\frac{dy}{dx}|_{\eta=const.} = -\frac{\eta_x}{\eta_y} = \frac{y_\xi}{x_\xi}.$$

Thus, the orthogonality of grid lines requires

$$x_\xi x_\eta + y_\xi y_\eta = 0. \tag{4.5.45}$$

Eq. (4.5.44) and (4.5.45) comprise the governing equations for hyperbolic grid generation [85]. To solve them numerically, the equations are first linearized about the known grid point $(\tilde{x}, \tilde{y})$, therefore we arrive at the linearized governing equations,

$$r_\eta + Cr_\xi = RHS, \tag{4.5.46}$$

where

$$r = \begin{Bmatrix} x \\ y \end{Bmatrix}, C = \frac{1}{\tilde{x}_\xi^2 + \tilde{y}_\xi^2} \begin{Bmatrix} a & b \\ b & -a \end{Bmatrix}, RHS = \begin{bmatrix} \tilde{x}_\xi & \tilde{y}_\xi \\ -\tilde{y}_\xi & \tilde{x}_\xi \end{bmatrix}^{-1} \begin{Bmatrix} 0 \\ \tilde{V} + V \end{Bmatrix}$$

with $a = \tilde{x}_\xi \tilde{x}_\eta - \tilde{y}_\xi \tilde{y}_\eta$, $b = \tilde{x}_\eta \tilde{y}_\eta + \tilde{x}_\eta \tilde{y}_\xi$. It can be easily shown that matrix C has two real distinct eigenvalues $s_{1,2} = \pm\sqrt{\frac{a^2+b^2}{\tilde{x}_\xi^2+\tilde{y}_\xi^2}}$, so that Eq. (4.5.46) is a hyperbolic PDE and can be

solved by marching in the direction of the time-like variable η. Thus, we specify the initial grid points on the airfoil (inner boundary) and then solve Eq. (4.5.46) numerically to find the grid points on the next η line, and so on, up to the outer boundary. Typically, Eq. (4.5.46) can be discretized by approximating r_η by forward finite difference and r_ξ by central finite difference at $\eta = k+1$ level, then we have

$$-\frac{C}{2}r_{j-1,k+1} + r_{j,k+1} + \frac{C}{2}r_{j+1,k+1} = RHS + r_{j,k}$$

or in Δ form,

$$-\frac{C}{2}\Delta r_{j-1,k} + \Delta r_{j,k} + \frac{C}{2}\Delta r_{j+1,k} = RHS + \frac{C}{2}(r_{j-1,k} - r_{j+1,k}), \tag{4.5.47}$$

with $\Delta r_{j,k} = r_{j,k+1} - r_{j,k}$. It is the nature of hyperbolic $PDEs$ that the discontinuities in the initial boundary conditions (discontinuities in the slopes at the trailing edge of the airfoil) will propagate into the grid system, so either artificial dissipations should be introduced or the upwind scheme with flux-vector-splitting should be employed to damp the travelling effects of discontinuities. Matsuno [58] employed high-order upwind scheme with the $MUSCL$ approach and the *minmod* limiter. Tai, Yin and Soong developed Roe's, Osher's and Stege-Warming's splitting formulation in [85] and compared their effects on grid generation around a cross section of delta-wing with a leading-edge vortex-flap. In these splitting upwind schemes, dissipation is inherent in the formulation. On the other hand, different artificial dissipation was introduced by Alsalihi [3], and Thompson [88] and Tai, et.al. in [85]. In general, high-order dissipations should be employed since low-order ones offer too large dissipation to maintain orthogonality. Here we adopt Tai *et al.*'s approach, i.e., 2^{nd} order dissipation with adaptive coefficients,

$$\mathbf{D} = \frac{s_j}{2}(\Delta_j \nabla_j) r_{j,k},$$

where s_j is the positive eigenvalue of matrix C at the current point and Δ, ∇ are forward and backward difference operators, respectively. By means of the adaptive coefficient s_j, large dissipation is introduced where necessary and excess dissipation is avoided. Then Eq.

(4.5.47) can be rewritten as,

$$-\frac{C}{2}\Delta r_{j-1,k} + \Delta r_{j,k} + \frac{C}{2}\Delta r_{j+1,k} = RHS + \frac{C}{2}(r_{j-1,k} - r_{j+1,k}) + \mathbf{D},$$

which is actually a tri-diagonal linearly system with 2×2 matrix entries and can be solved by the standard method. Boundary conditions, including periodicity for the O-type grid, constant Cartesian plane for the C-type grid, and some others, were discussed in detail by Chan and Steger in [14]. For the O-type grid, the periodicity conditions should be imposed on $j = 0$ and $j = j_{max}$ lines, i.e.,

$$\Delta r_{0,k} = \Delta r_{j_{max},k}.$$

The grid density is controlled by specifying cell areas $V_{j,k}$. As outlined in [34] three approaches have been employed by different authors to specify the cell areas. Thompson [88] suggested a method via constructing a set of concentric circles with radii obeying an algebraic distribution function which controlled the grid line spacing in η -direction. The innermost circle bears the same perimeter and the same grid point distribution as the inner boundary (the airfoil in the present study). The cell areas computed from this model were supplied to the grid generation. Another approach by Steger, *et al.* [83] employed a straight line instead of a circle to model the inner boundary. Nonuniform parallel lines were drawn in η-direction to generate a rectangular mesh. The $V_{j,k}$ was then set to the cell areas of this mesh. In Hoffmann *et al.*'s approach [35], a numerical procedure was first employed to generate a grid system for the airfoil and the cell areas were computed from this system. In the present study, the approach by Thompson is first adopted to find $\bar{V}_{j,k}$; then the supplied cell areas are set to be

$$V_{j,k} = t(k)\bar{V}_{j,k} + (1 - t(k))V_0,$$

where V_0 is a constant, and $0 \leq t(k) \leq 1$ assumes the value of 1 for $k = K_{max}$, and the value of 0 for $k = 0$. The purpose of introducing this interpolation function is to distribute grid point uniformly on the outer boundary.

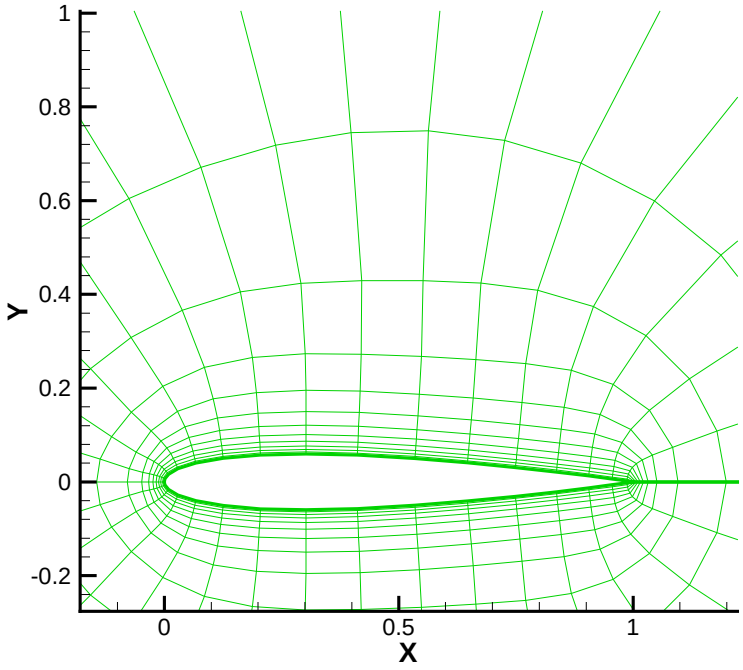

Figure 4.3: Close-up view of 97×49 O-type grid around $NACA$ 0012 airfoil shown in every fourth line

A close-up view of the O-type grid system around a $NACA$ 0012 airfoil is depicted in Fig. 4.3 where results are shown for every fourth line. The orthogonality is well achieved in the vicinity of the airfoil even around the leading edge and trailing edge. The robustness of this hyperbolic grid generation method is demonstrated by generating grid system around an unrealistic $NACA$ 4-digit airfoil with $m = 0.3, p = 0.5, t = 0.06$. In Fig. 4.4 the overall view of the grid system is presented. Fig. 4.5 and 4.6 show the details in the vicinity of the trailing edge and leading edge, respectively. As seen from these figures, the density of grid lines in η-direction in the vicinity of the airfoil is much greater than required by inviscid Euler solver. This is required by the adjoint properties.

Grid convergence studies are conducted using 5 different grid systems as shown in Fig. 4.7 and 4.8[6]. The Mach number of the freestream is 0.8 and the attack of angle is 1.25°.

[6]The numbers shown in the figures indicate the number of panels.

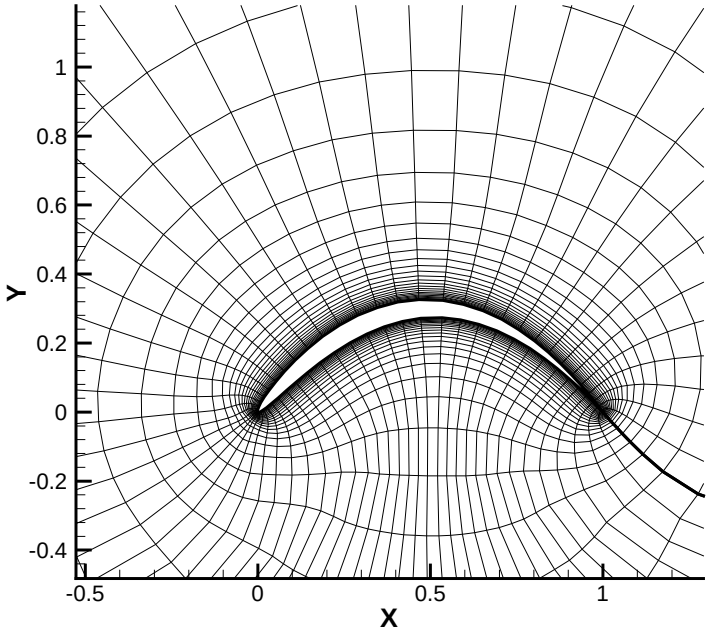

Figure 4.4: Overall view of 65×33 O-type grid around airfoil $m = 0.3, p = 0.5, t = 0.06$

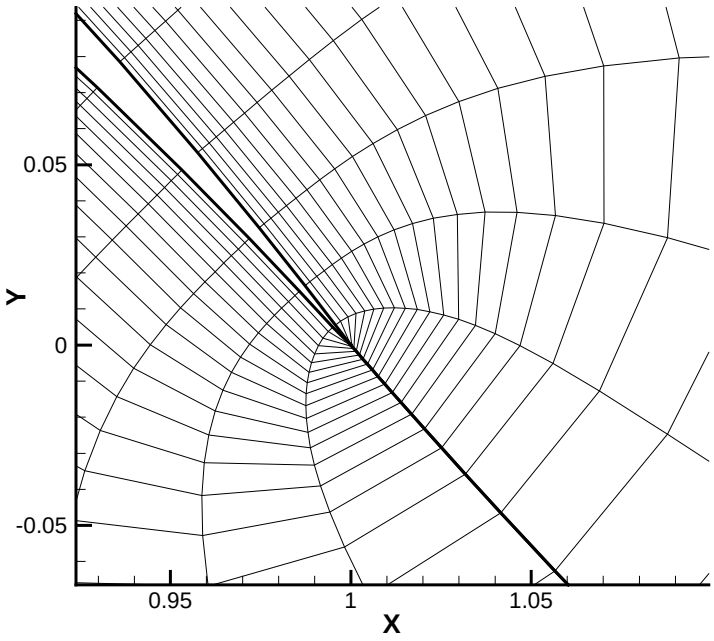

Figure 4.5: Close-up view of trailing edge vicinity

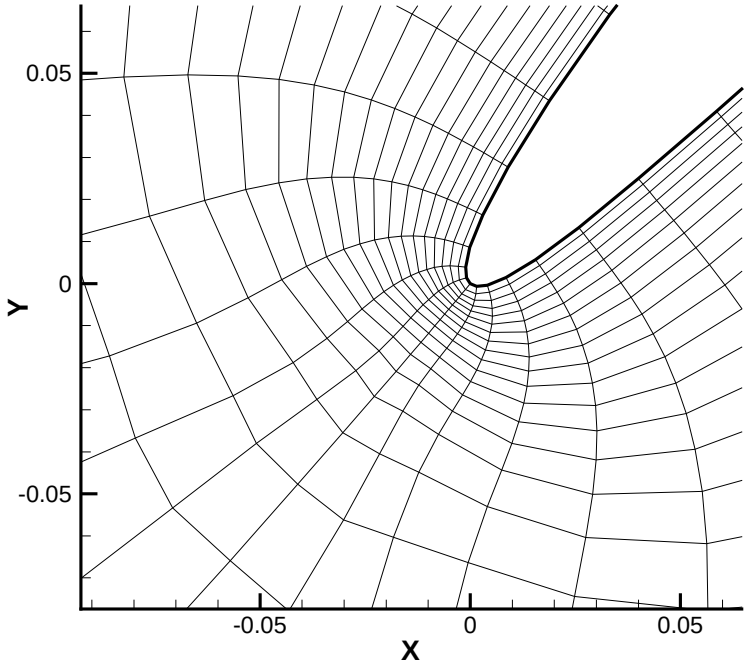

Figure 4.6: Close-up view of the leading edge vicinity

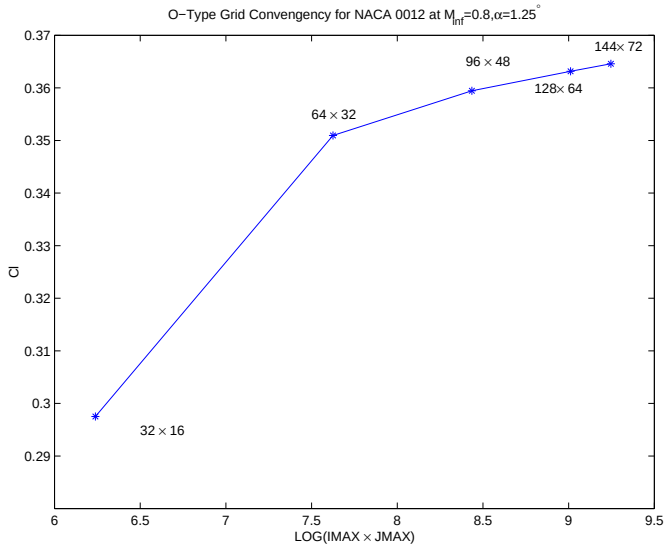

Figure 4.7: Grid convergence study on O-type grid system: C_L

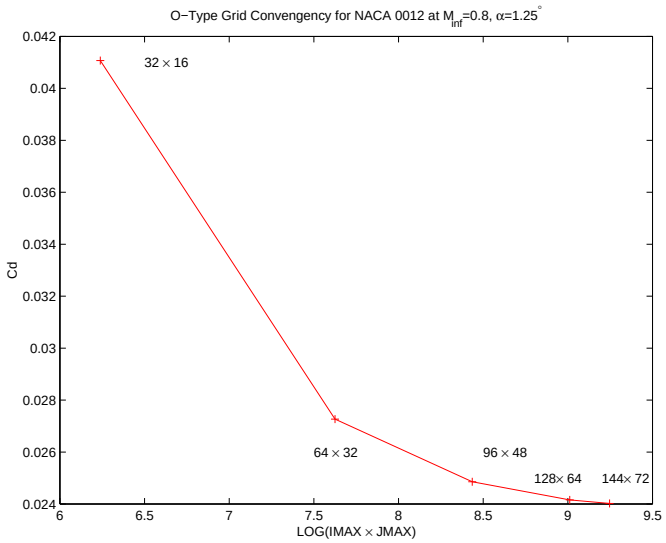

Figure 4.8: Grid convergence study on O-type grid system: C_D

4.6 Geometric Design Variables

The choice of geometric design parameters remains an open issue in optimization design studies. As early as in 1970's, Vanderplatts and Hicks used polynomials to define an airfoil shape, taking the coefficients as the geometric design variables. In [90], Vanderplaats adopted 6 basic shapes to span a design space. This approach was also employed by Shenoy in his thesis [78]. In Jameson's studies of adjoint methods, the coordinates of every surface point were taken as design parameters, leading to a very large design space. Although this approach provides great freedom to change the airfoil shape, it may produce discontinuities on gradients which can be smoothed by an implicit smoothing technique. Another choice of taking B-spline control points as design parameters was explored by Burgeen *et al.* [12]. Reuther [73] discussed the advantages and limitations of these choices. In the present studies, two approaches of parameterizing the airfoil are utilized. One approach employs the shape functions given by Hicks & Henne [33] to parameterize shape changes in the airfoil; the other one uses the definition of the $NACA$ 4-digit airfoil series [1].

4.6.1 Hicks & Henne shape perturbations

In this approach, the shape function introduced by Hicks and Henne [33] are employed to parameterize the changes of an airfoil shape (the upper and/or lower surface) from a basic $NACA$ 0012 airfoil,

$$\Delta y = \Sigma_{k=1}^{N} \delta_k f_k(x) \tag{4.6.48}$$

where N is the number of geometric design parameters, δ_k are chosen design parameters, and shape functions $f_k(x)$ are given by,

$$f_k(x) = \begin{cases} \sin(\pi(1-x)^{e_k}) & k = 1, 2, \\ \sin^3(\pi x^{e_k}) & k = 3, 4, 5, 6, \\ \sin(\pi x^{e_k}) & k = 7, 8, \end{cases} \tag{4.6.49}$$

with

$$e_k = \begin{cases} \frac{\log(0.5)}{\log(1-x_k)} & k = 1, 2 \\ \frac{\log(0.5)}{\log(x_k)} & \text{otherwise} \end{cases}$$

and $x_k = (0.06, 0.13, 0.2, 0.4, 0.6, 0.8, 0.87, 0.94)$. The shape functions are depicted in Fig. 4.9, showing that each shape function f_k has its maximum at the point x_k.

Tables 4.1-4.4 compare the FDM [7] and ADJ results of derivatives of C_L and C_D with respect to each geometric design parameter and angle of attack using four different grid systems (with 8 design parameters modifying the upper surface). The airfoil is an $NACA$ 0012; and the flow condition is $M_\infty = 0.8$ and $\alpha = 1.25°$. It is shown that the finer the grid system is, the better agreement we get. The largest errors exist for the first geometric design parameter, which peaks at $x = 0.06$. In the subsequent numerical studies, we employed the grid system 97×49.

[7] FDM involves finite differencing two Euler flow solutions with slightly different design parameters.

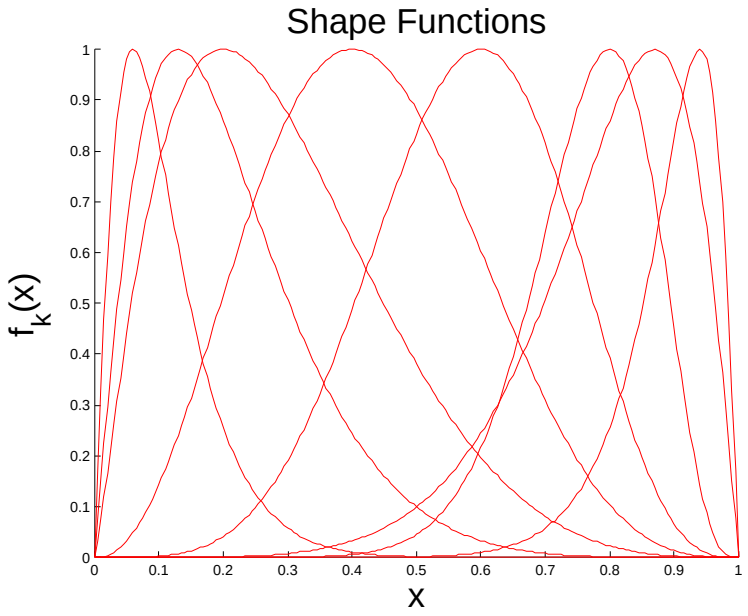

Figure 4.9: Shape functions

		1	2	3	4	5	6	7	8	AoA
$\frac{\partial C_l}{\partial g_i}$	FDM	2.6339	4.0384	4.1620	3.9603	8.4599	7.4531	22.6257	32.4284	13.9977
	ADJ	3.4090	4.4147	4.9062	5.8164	11.3125	5.7613	26.9139	38.2098	19.2985
$\frac{\partial C_d}{\partial g_i}$	FDM	0.05321	1.3610	1.5993	0.7043	-0.2928	0.0866	0.8440	1.5988	1.0212
	ADJ	0.3563	1.3589	1.6044	0.8677	-0.0229	0.0427	1.2393	2.1370	1.4190

Table 4.1: Comparisons of gradients of C_L and C_D from FDM and ADJ on O-grid 65×33

		1	2	3	4	5	6	7	8	AoA
$\frac{\partial C_l}{\partial g_i}$	FDM	3.3070	4.3152	3.8675	2.9966	8.8746	8.4136	23.6978	32.6534	14.5927
	ADJ	4.0478	4.8121	4.4250	3.8810	10.4111	8.4677	24.5309	34.7020	17.7081
$\frac{\partial C_d}{\partial g_i}$	FDM	0.03963	1.4404	1.6515	0.6756	-0.2678	0.1344	0.9324	1.6680	1.1099
	ADJ	0.2177	1.4427	1.6527	0.7347	-0.1683	0.1571	1.0263	1.8444	1.3113

Table 4.2: Comparisons of gradients of C_L and C_D from FDM and ADJ on O-grid 97×49

		1	2	3	4	5	6	7	8	AoA
$\frac{\partial C_l}{\partial g_i}$	FDM	3.4831	4.3412	3.8880	3.1896	9.9922	9.1986	25.2465	34.2343	15.2930
	ADJ	3.8906	4.7853	4.2757	3.3613	9.7333	8.7674	23.0953	30.8504	14.7752
$\frac{\partial C_d}{\partial g_i}$	FDM	0.05626	1.4773	1.6825	0.7003	-0.2092	0.1623	1.0045	1.7511	1.1564
	ADJ	0.13239	1.4714	1.6800	0.7120	-0.2177	0.1500	0.9133	1.5981	1.2353

Table 4.3: Comparisons of gradients of C_L and C_D from FDM and ADJ on O-grid 129×65

		1	2	3	4	5	6	7	8	AoA
$\frac{\partial C_l}{\partial g_i}$	FDM	3.8481	4.3767	3.5942	2.5287	9.6119	9.4005	25.1853	33.7639	15.1831
	ADJ	3.9994	4.9098	4.2787	3.1678	9.8554	9.2343	23.7044	30.9670	14.8117
$\frac{\partial C_d}{\partial g_i}$	FDM	0.0811	1.4959	1.6807	0.6704	-0.2269	0.1667	1.0024	1.7331	1.1634
	ADJ	0.1070	1.4973	1.6979	0.7033	-0.2125	0.1676	0.9475	1.6152	1.2404

Table 4.4: Comparisons of gradients of C_L and C_D from FDM and ADJ on O-grid 161×81

4.6.2 A $NACA$ 4-digit airfoil family

In $NACA$ 4-digit airfoil series, three parameters, m (the maximum mean camber), p (the chordwise position of the maximum mean camber) and t (maximum thickness of the airfoil) are used to define the airfoil shape [1], in which the thickness distribution is given by,

$$\pm y_t = \frac{t}{0.20}(0.29690\sqrt{x} - 0.12600x - 0.35160x^2 + 0.28430x^3 - 0.10150x^4). \tag{4.6.50}$$

The mean line for locations where $x \leq p$ is taken to be

$$y_c = \frac{m}{p^2}(2px - x^2), \tag{4.6.51}$$

and for $x > p$ the mean line is given by

$$y_c = \frac{m}{(1-p)^2}(1 - 2p + 2px - x^2). \tag{4.6.52}$$

The airfoil shape can be obtained by combining the mean line with the thickness distribution. The points on the upper surface are defined as

$$\begin{aligned} x_U &= x_c - y_t \sin\theta \\ y_U &= y_c + y_t \cos\theta \end{aligned}$$

	$\frac{\partial C_L}{\partial m}$	$\frac{\partial C_L}{\partial t}$	$\frac{\partial C_L}{\partial \alpha}$
FDM	29.6471	0.2217	14.5927
ADJ	28.8304	0.2749	13.5726

Table 4.5: Comparisons of gradients of C_L from FDM and ADJ

	$\frac{\partial C_D}{\partial m}$	$\frac{\partial C_D}{\partial t}$	$\frac{\partial C_D}{\partial \alpha}$
FDM	1.7042	0.5509	1.1099
ADJ	1.6838	0.5475	1.1973

Table 4.6: Comparisons of gradients of C_D from FDM and ADJ

where θ is the angle between the tangent of the mean line and the chord line. The corresponding points on the lower surface are be expressed as [8]

$$\begin{aligned} x_L &= x_c + y_t \sin\theta \\ y_L &= y_c - y_t \cos\theta. \end{aligned}$$

In our numerical studies for this parameterization, m, t, α are taken as design variables and p is frozen at 0.45. The reduced gradients of C_L and C_D from the adjoint method (Section 4.2) and a finite difference method are listed in Tables 4.5 and 4.6, respectively. The results agree well except there exists 25% disagreement in $\frac{\partial C_L}{\partial t}$. However, since $\frac{\partial C_L}{\partial t}$ is less significant than the other derivatives $\frac{\partial C_L}{\partial m}$ and $\frac{\partial C_L}{\partial \alpha}$, this disagreement will not considerably affect the optimization procedures.

[8]The airfoil obtained in this way is not closed, since there is a small gap at the trailing edge. In the numerical grid generation, this gap is eliminated by slightly modifying the airfoil shape.

4.7 Optimal Design for Transonic Airfoils

In this section, computational tests are presented for the reduced problem (2.5.8) and the super-reduced problem (2.5.9). The reduced (super-reduced) gradients of cost/constraint functionals are utilized by an optimizer to perform gradient-based optimum aerodynamic design; the procedure is illustrated in Fig. 4.10. The design starts from the initial parameters (including shape parameters and flow-related parameters), the grid system is then generated followed by running the flow solver to obtain the cost/constraint functionals, and by running the adjoint solver to obtain the reduced (super-reduced) gradients. If the convergence criterion is not satisfied, the optimization program will update the design parameters (for super-reduced problem, the angle of attack should also be updated according to (4.3.38).). The grid system is generated using updated parameters to run the flow solver and adjoint solver, and so on; otherwise the design procedure stops. In the following numerical tests, the optimization process starts from a $NACA$ 0012 airfoil and proceeds to minimize the drag coefficient while maintaining the lift coefficient at a specific value at transonic freestream Mach number $M_\infty = 0.8$.

4.7.1 Reduced problem

Design on $NACA$ 4-digit airfoil (Test case 0)

A constrained optimal design is conducted to minimize the drag coefficient subject to the lift constraint $C_{L_0} = 0.5$. Fig. 4.11 shows the comparisons of pressure distribution $(1-p/p_0)$ and airfoil shape for the airfoil $NACA$ 0012 ($\alpha = 1.25°$) and the optimal airfoil with $m = 0.01072, t = 0.05(\alpha = 1.2362°)$. In the optimization procedure, the thickness t quickly hits its lower bound (0.05) and remains unchanged in the following iterations to make the optimal airfoil much thinner than the starting airfoil. The optimal airfoil is no longer symmetric, with a small maximum camber $m = 0.01072$. The 3^{rd} row of the Table 4.8 summarizes the

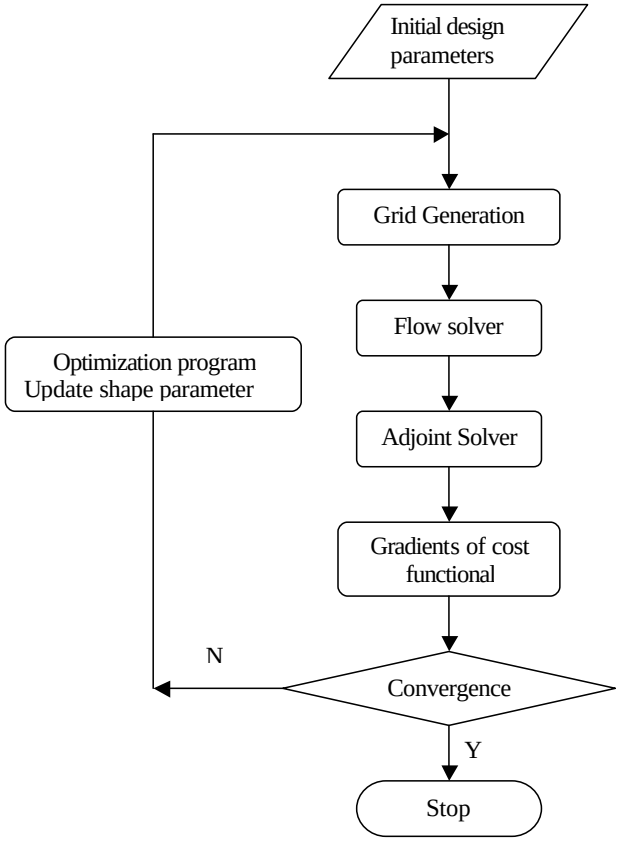

Figure 4.10: Gradient-based optimization procedures using adjoint method

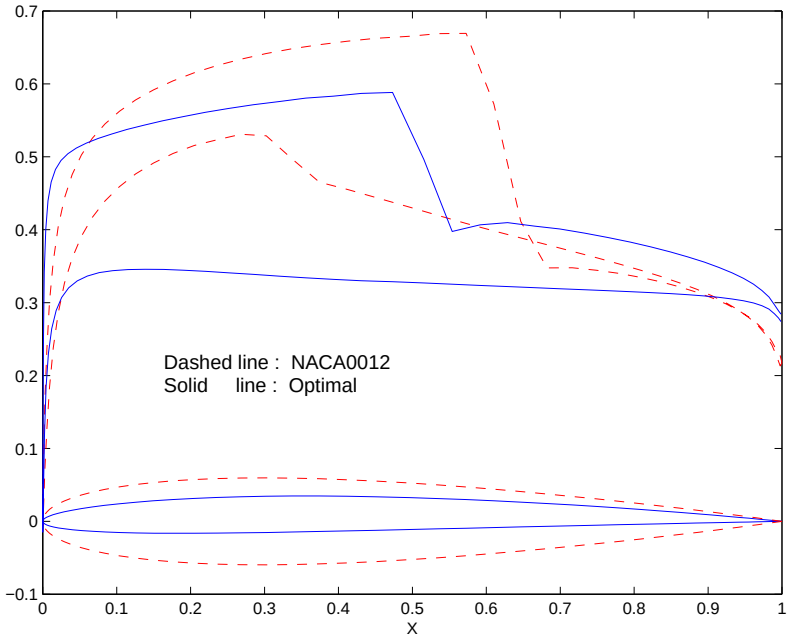

Figure 4.11: Comparisons of airfoil shapes and pressure distributions of $NACA$ 0012 and optimal airfoil in Test Case 0, $C_{L_0} = 0.5$

results of the optimization procedure starting from the airfoil $NACA$ 0012 at $\alpha = 1.25°$ and shows that the drag coefficient decreases from 0.02216 to 0.005186. The optimal pressure distribution, denoted by the solid line in Fig. 4.11, reveals that the wave drag is decreased through weakening the strength of the shock on the upper surface.

Design using Hicks & Henne shape functions (Test case 1, 2)

In these two test cases, the design variables include 8 geometric variables (to modifying the upper surface) and the angle of attack. Table 4.7 displays starting design variables and their lower bound (LB), upper bound (UB) and optimal values for two test cases. The drag and lift coefficients are given in Table 4.8 for the starting and optimal airfoils, as well as the number of iterations and functional evaluations in the optimization procedures. In Fig. 4.12 (4.13), the shape of the starting $NACA$ 0012 airfoil and its pressure distribution are

DVs	Test Case 1				Test Case 2			
	$Start$	LB	UB	$Opt.$	$Start$	LB	UB	$Opt.$
1	0	-0.005	0.005	-0.002308	0	-0.005	0.005	-0.002091
2	0	-0.005	0.005	-0.005	0	-0.01	0.005	-0.007828
3	0	-0.005	0.005	-0.005	0	-0.01	0.005	-0.005148
4	0	-0.005	0.005	-0.005	0	-0.01	0.005	-0.008303
5	0	-0.005	0.005	0.005	0	-0.005	0.01	0.008537
6	0	-0.005	0.005	0.005	0	-0.005	0.01	0.004080
7	0	-0.005	0.005	0.005	0	-0.005	0.01	0.01
8	0	-0.005	0.005	$8.9607e-4$	0	-0.005	0.005	-0.002609
9	1.25°	0°	5°	1.2121°	1.25°	0°	5°	1.2277°

Table 4.7: Optimization results on O-Type grid 97×49

represented by the dashed lines and the solid lines represent the optimized airfoil and the corresponding pressure distribution in test case 1 (test case 2). In test case 1, the upper and lower bound for the design parameters are set to ± 0.005. The computational results reveal that six parameters hit the lower or upper bounds, which means the proposed bounds set limits on the optimization. That is the reason why the shock on the upper surface is not eliminated for the optimal airfoil. In test case 2, the lower bound are change to -0.01 for those parameters hitting the lower bounds in test case 1, and the upper bounds are change to 0.01 for those parameters hitting the upper bounds in test case 1. In this case, the shock on the upper surface almost disappears for the optimal airfoil as shown in Fig 4.13, therefore the drag coefficient of the optimal airfoil is decreased from 0.008943 to 0.004576. However, more iterations and more Euler solutions are required for the test case 2.

Two additional geometric design parameters are introduced to change the lower surface of the airfoil; thus, we have 11 design variables, 10 geometric variables and 1 flow-related variable, the angle of attack. All the computational cases start from a $NACA$ 0012 airfoil at $AoA = 3^\circ$.

	C_L	C_D	$Iter.$	$FunCount$	Normof derivative
$NACA$ 0012	0.3275	0.02216			
Test Case 0	0.49990	0.005185	4	9	1.85×10^{-5}
Test Case 1	0.49998	0.008943	5	11	-5.37×10^{-6}
Test Case 2	0.49980	0.004576	15	39	-0.000117

Table 4.8: Optimization results for Test Case 0, 1 ,2

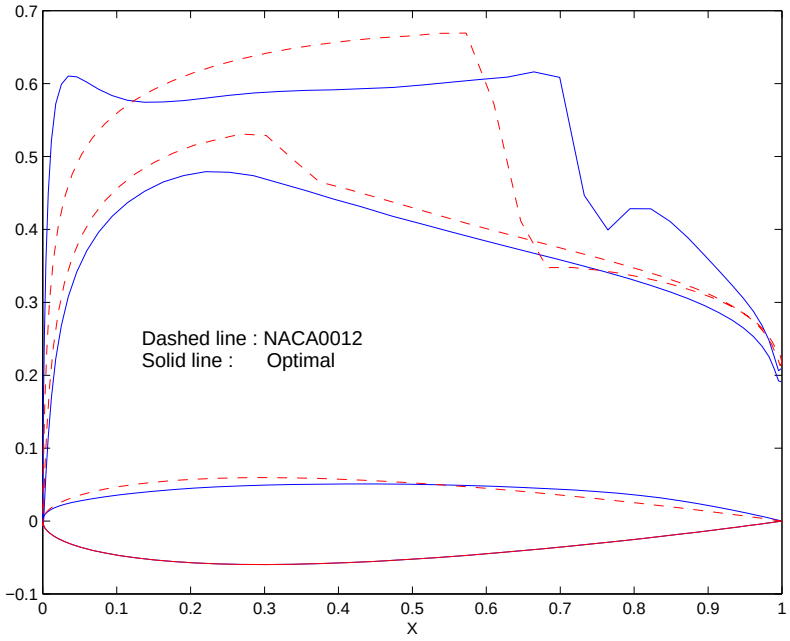

Figure 4.12: Comparisons of airfoil shapes and pressure distributions of $NACA$ 0012 and optimal airfoil in Test Case 1, $C_{L_0} = 0.5$

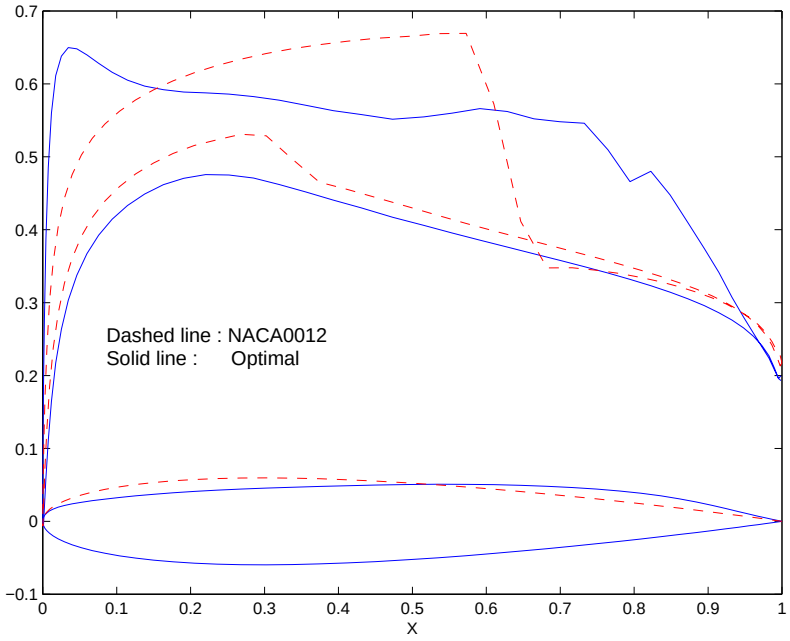

Figure 4.13: Comparisons of airfoil shapes and pressure distributions of $NACA$ 0012 and optimal airfoil in Test Case 2, $C_{L_0} = 0.5$

Fig. 4.14-4.18 show the optimization results at different specified C_{L_0}. The shock on the upper surface disappears for the cases of low lift coefficients ($C_{L_0} = 0.3, 0.4, 0.5$); for high lift coefficients ($C_{L_0} = 0.6, 0.7$), this shock is greatly weakened and pushed downstream. In this way the wave drag is considerably decreased. In all cases, the upper surface of the airfoil is flattened to allow for the 'aft-load', which contributes a considerable fraction of lift for cases with high lift coefficients. In contrast to the considerable modifications to the upper surface, only small changes have been made to the lower surface. An imperfection for these designs lies in the fact that the optimized pressure distributions are not very smooth for the cases with relatively low lift coefficients. This phenomenon can be avoided by adding more design parameters for the upper surface. Table 4.9 summarizes the optimal drag and lift coefficients and angles of attack for all design cases. The closeness of the optimal lift coefficients to the specified lift coefficients implies that CFD can not only be utilized as the analysis tool to find the flow solution under given conditions, it can also be employed as a design tool to find the

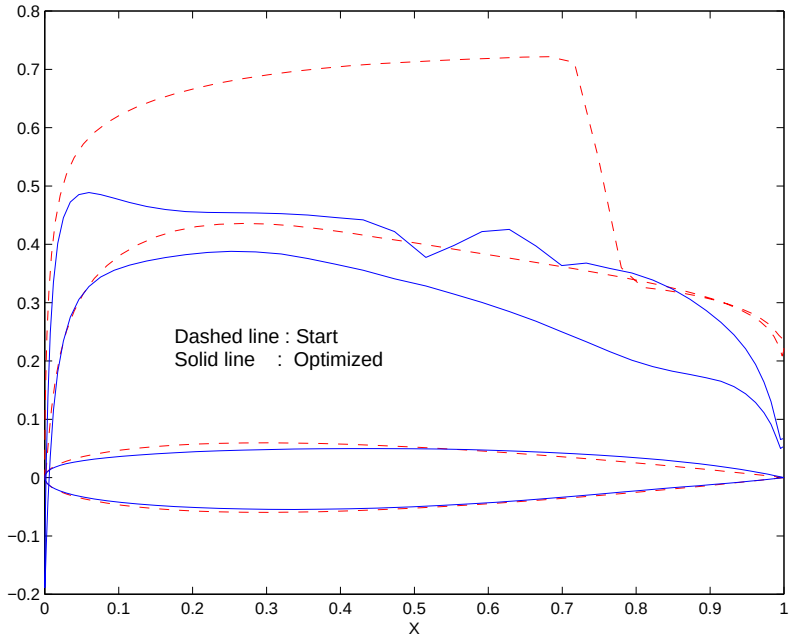

Figure 4.14: Comparisons of airfoil shapes and pressure distributions of $NACA$ 0012 and optimal airfoil, $C_{L_0} = 0.3$

geometric and freestream conditions which satisfied the specified requirements. Also listed in this table are the numbers of iterations and functional evaluations in the optimization procedures. As shown in Fig. 4.19 which depicts the $C_L - C_D$ polar for the $NACA$ 0012 (dashed line) and optimal airfoils (solid line), the optimization designs are successful in accomplishing the design objective. (Note that the geometry of the airfoil in this polar for optimal airfoils is not fixed.)

4.7.2 Super-reduced problem

In this part the super-reduced gradients of C_D are calculated from the DLM (stands for Direct Lagrange Multiplier method) and FDM for $NACA$ 0012 airfoil at $\alpha = 1.25°$ and $M_\infty = 0.8$, at which $C_L = 0.32748$ $C_D = 0.022161$. In Table 4.10, each of the geometric design variables is perturbed by what is listed in the second row with others set to zero. The

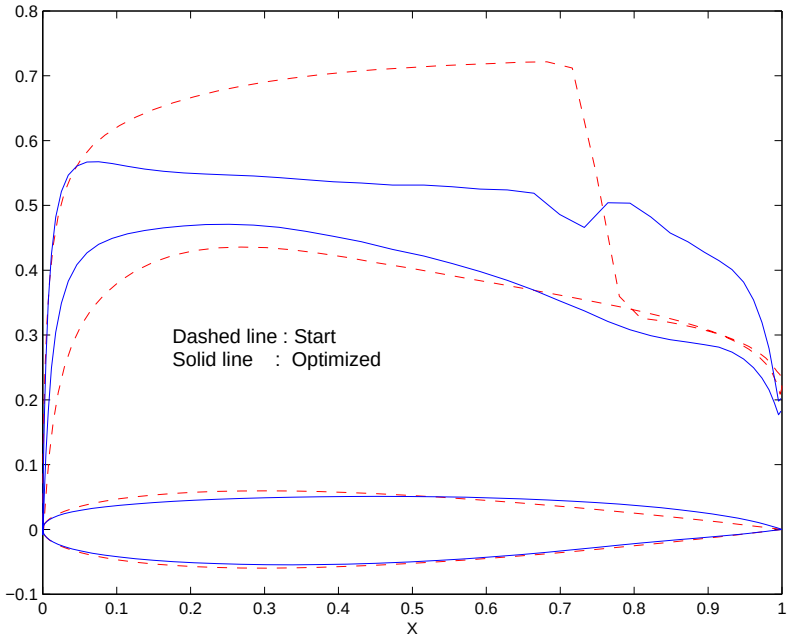

Figure 4.15: Comparisons of airfoil shapes and pressure distributions of $NACA$ 0012 and optimal airfoil, $C_{L_0} = 0.4$

perturbations in angle of attack are calculated according to (4.3.38) to maintain $C_L = C_{L_c}$. The drag and lift coefficients at perturbed geometric variables and the angle of attack are also given in the table. It can seen that the relative changes in C_L from the nominal value $C_{L_c} = 0.32748$ is significantly smaller than those in C_D from its nominal value $C_D = 0.022161$. The gradients from DLM are calculated from (4.3.35), which agree well with the results from FDM.

This methodology is then applied to the unconstrained optimization design with simple box bounds (upper and lower bounds on the design variables), in which the airfoil shape is parameterized by m and t. The optimization procedure starts by $NACA$ 0012 airfoil with $m = 0$ and $t = 0.12$ at angle of attack $\alpha = 3°$, freestream Mach number $M_\infty = 0.8$. In each iteration of the optimization procedure, the optimizer determines the changes in the geometric design variables, i.e., $\Delta\mathcal{G} = (\Delta m, \Delta t)^T$. Then the change in α, $\Delta\alpha$, is obtained from (4.3.38) to maintain the lift coefficient. The next iteration begins at $\mathcal{G}+\Delta\mathcal{G}$ and $\alpha+\Delta\alpha$.

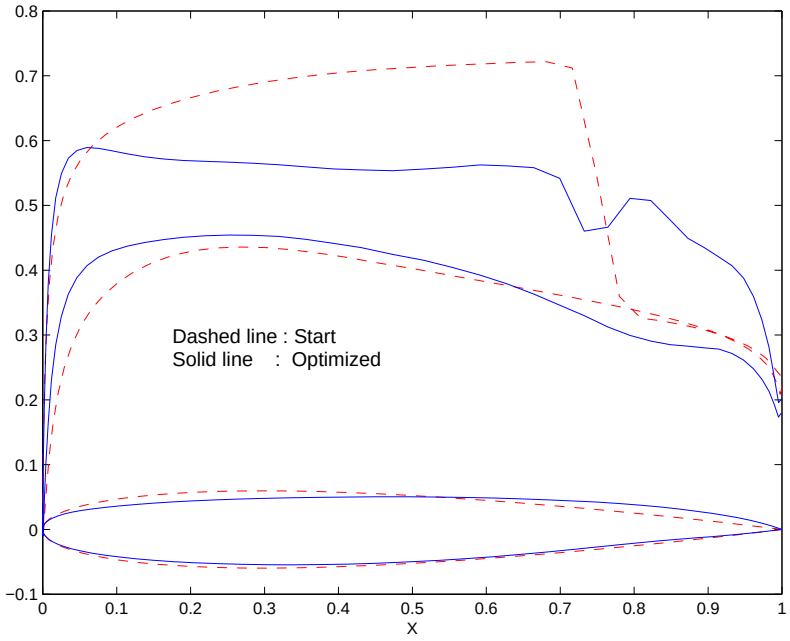

Figure 4.16: Comparisons of airfoil shapes and pressure distributions of $NACA$ 0012 and optimal airfoil, $C_{L_0} = 0.5$

The pressure distributions and the airfoil shapes for $NACA$ 0012 and optimized airfoil are presented in Fig. 4.20. The drag and lift coefficients for these two airfoils are given in Table 4.11. By contrast to very small change (0.7459 to 0.7452) in C_L, the drag coefficient decreases from 0.06940 to 0.01793. As pointed out in Section 4.7.1, this reduction results from weakening the shock to decrease the wave drag. In the optimization design, the thickness hits the low bound (0.005), so we arrive at a very thin airfoil, which may not be realistic. However, this computational test shows that the super-reduced methodology is successful in terms of reducing the drag coefficient while maintaining the lift coefficient.

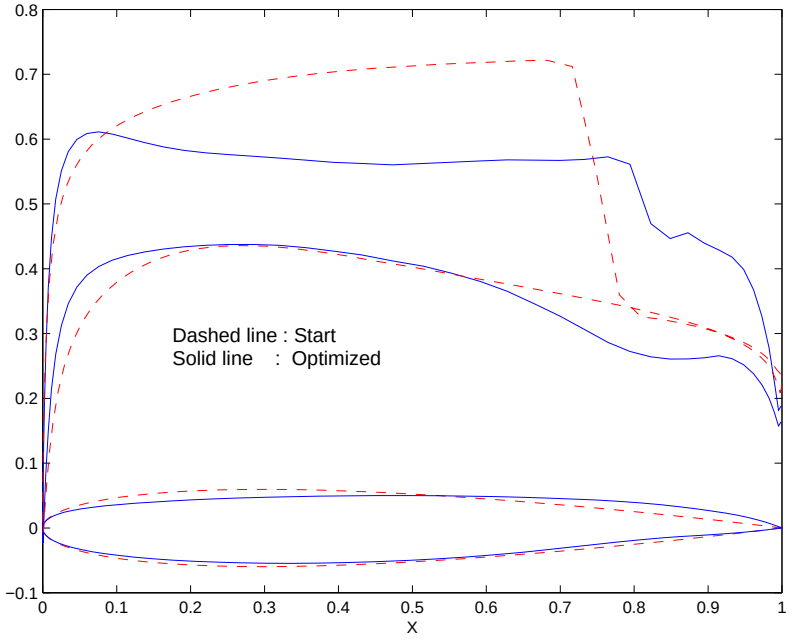

Figure 4.17: Comparisons of airfoil shapes and pressure distributions of $NACA$ 0012 and optimal airfoil, $C_{L_0} = 0.6$

C_{L_0}	.3	.4	.5	.6	.7
C_L	.30001	.40004	.50078	.59977	.70014
C_D	2.3484e-3	3.0834e-3	4.4007e-3	5.7889e-3	1.1212e-2
AoA	.3316°	.1347°	.3605°	.3843°	.7874°
$Iters$	17	18	11	8	12
$funcCount$	46	42	28	21	34

Table 4.9: Optimal C_L, C_D and AoA for different C_{L_0}

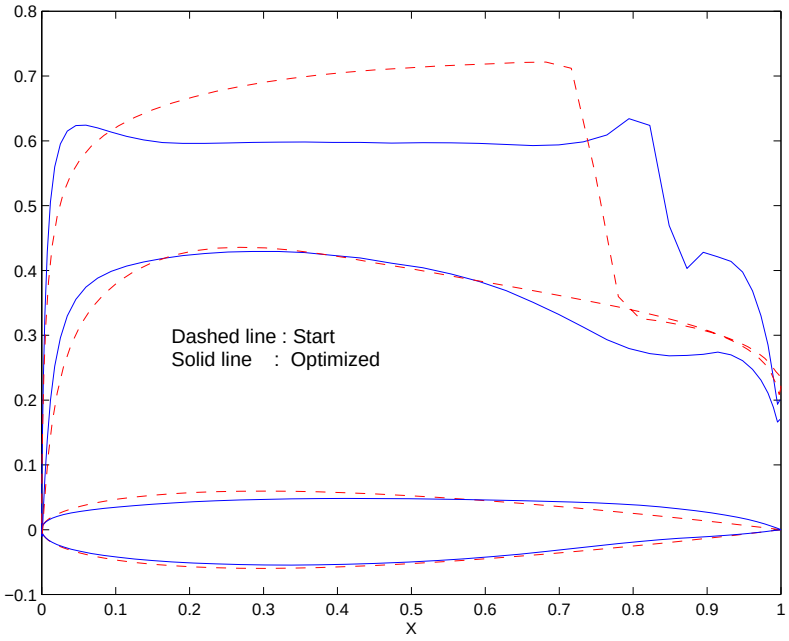

Figure 4.18: Comparisons of airfoil shapes and pressure distributions of $NACA$ 0012 and optimal airfoil, $C_{L_0} = 0.7$

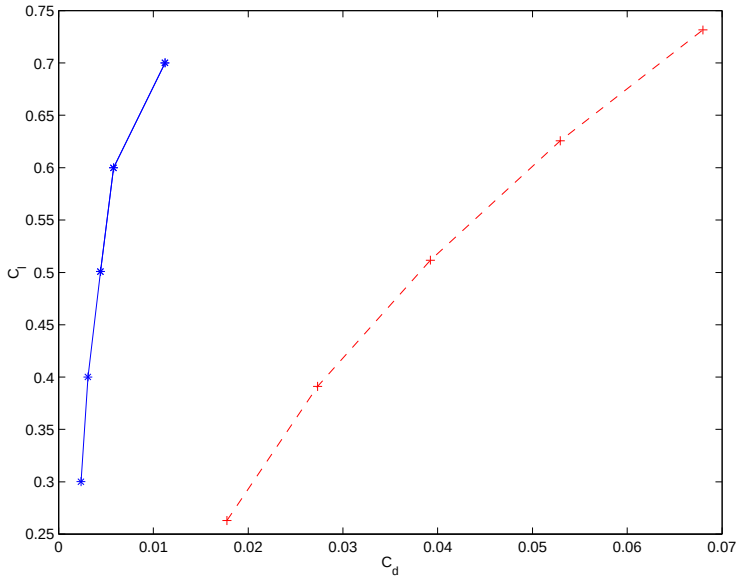

Figure 4.19: $C_L - C_D$ comparisons for $NACA0012$ and optimal airfoils

Variable		1	2	3	4	5	6	7	8
$\Delta\mathcal{G}_i$		.001	.002	.002	.002	.002	.002	.002	.001
$\Delta\alpha(^\circ)$		-.01542	-.03669	-.03383	-.02928	-.07765	-.06302	-.1829	-.1295
C_D		2.1989e-2	2.4414e-2	2.4854e-2	2.3134e-2	2.0191e-2	2.1224e-2	2.0540e-2	2.1352e-2
C_L		3.2709e-1	3.2659e-1	3.2673e-1	3.2613e-1	3.2538e-1	3.2834e-1	3.2890e-1	3.2746e-1
$\frac{\partial C_D}{\partial \mathcal{G}_i}$	DLM	-0.1244	1.0363	1.2802	0.4115	-1.0388	-0.5550	-1.0382	-1.0776
	FDM	-0.1717	1.1265	1.3465	0.4580	-0.9850	-0.4685	-0.8105	-0.8090

Table 4.10: Comparisons of geometric gradients of C_D at frozen C_L from FDM and DLM

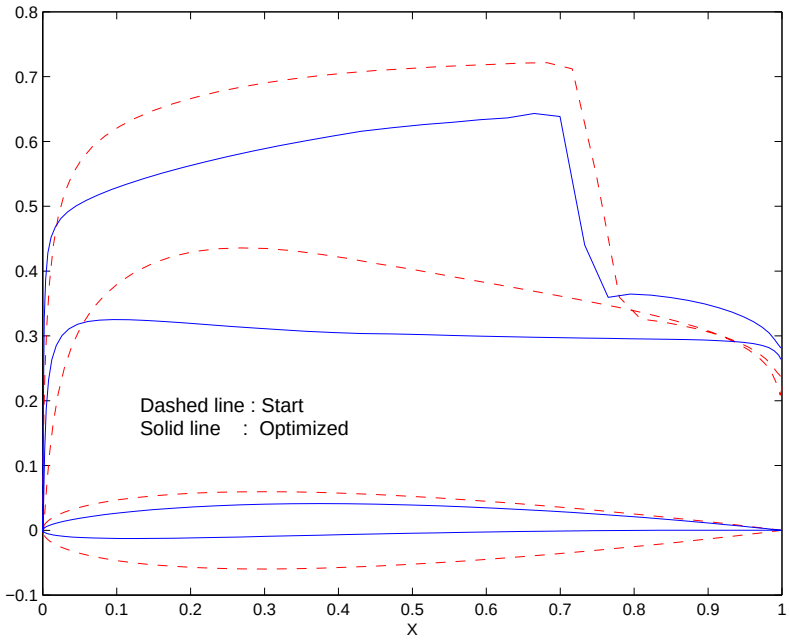

Figure 4.20: Comparisons of airfoil shapes and pressure distributions of $NACA$ 0012 and optimal airfoil, super-reduced problem

	C_L	C_D	α
$NACA$ 0012	0.7459	0.06940	3°
Optimal	0.7452	0.01793	1.4143°

Table 4.11: Starting and optimal C_L and C_D, super-reduced problem

Chapter 5

Concluding Remarks

The continuous adjoint method has been investigated in the present research to show its reliability and efficiency in calculating the gradients of cost/constraint functionals with respect to the design parameters (geometric and flow-related parameters both included). This method and an Euler flow solver have been incorporated into an optimization program to conduct optimum aerodynamic designs.

In Chapter 3 the adjoint method is applied to the quasi 1-D nozzle flow. Three flow conditions are considered, namely the isentropic subsonic, isentropic transonic flows and the shock flow, to show the properties of adjoints at the sonic throat and the shock location. The adjoints are proven to be continuous at the shock location. The singularity of the adjoints at the sonic throat is associated with the vanishing characteristic $u - a$. The non-uniform grid system can better resolve the logarithmic singularity in the adjoint solutions. This singularity rarely happens in 2-D or 3-D case since in general the streamlines are not perpendicular to the sonic line. The proposed shock-capturing numerical method to solve the adjoint equations works well for three cases, and predicts good agreement between the numerical results and results from FDM. This method is combined with an optimization code to conduct a pressure-matching inverse design for the nozzle flow.

The gradient-based optimum aerodynamic design for 2-D airfoils using adjoint methods is addressed in Chapter 4. A Lagrange Multiplier Theorem is applied to derive the ad-

joint equations and the characteristic far-field boundary conditions. By adjoining all the boundary conditions to the cost functional, the adjoint method is extended to calculate the gradients of cost/constraint functionals with respect to the flow-related variables, the angle of attack. Two ways of parameterizing the airfoil shape are explored, one of which employs Hicks and Henne's shape functions, and the other of which utilizes the $NACA$ 4-digit airfoil formulation. Computational experiments are performed both on the reduced problem and super-reduced problem. In the former problem, the reduced gradients of the cost functional are obtained through adjoint methods. These reduced gradients can be employed to construct super-reduced gradients, the gradients subject to the lift constraint. In this way, the super-reduced problem is solved as an optimization design without the nonlinear lift constraint.

Based on the numerical tests conducted in the current research, we can confidently state that the adjoint method is an efficient way of calculating reduced (super-reduced) gradients of functionals, especially in design problems with many design variables but with relatively few cost/constraint functionals. The numerical method to solve the flow problem can also be suitably modified to solve the linear adjoint equations with the point in mind that the characteristic waves of adjoints travel in the opposite direction to that of the flow variables. However, since adjoints display different properties from the flow solution, different grid system can be adopted for the adjoint solver to obtain more accurate gradient results even though this different grid system may require interpolation of flow solutions.

Although the numerical properties of adjoints and characteristic boundary conditions are examined in this study, an issue remained to be explored in the future is the physical meaning of adjoints. For example, the flow variables can be integrated to reveal some conservation laws, such as mass conservation, energy conservation and momentum conservation. Are there similar laws for adjoints? The exploration in this direction will deepen our insights into the adjoint method and help develop new numerical techniques to solve the adjoint equation in a more efficient way. Other research may focus on integrating the grid adaption techniques to the design procedures, since the location and the strength of the shock change during

the optimization procedures. The adapted grid technique will provide high resolution to the shock and therefore is expected to improve the accuracy of gradients, which is critical in gradient-based optimizations.

Appendix A

Mathematical Basis for Adjoint Method

This chapter briefly reviews some mathematical basis for the adjoint method. Most the materials are excerpted from [56].

A.1 Definition of Adjoint

Optimality conditions for many constrained optimization problems can be conveniently stated in terms of adjoint operators. This review starts from normed vector spaces and introduces the ideas underlying adjoint operators.

Definition A.1.1. *A normed vector space is a vector space X on which a norm is provided which maps each element $x \in X$ into a real number $\|x\|$ and satisfies the following axioms:*

- $\|x\| \geq 0$ *for all* $x \in X$, $\|x\| = 0$ *if and only if* $x=\boldsymbol{0}$.
- $\|x + y\| \leq \|x\| + \|y\|$ *for each* $x, y \in X$.
- $\|\alpha x\| =\mid \alpha \mid \|x\|$ *for all scalars* α *and each* $x \in X$.

A complete normed linear vector space is named as a *Banach space*. A Banach space equipped with an inner product is a *Hilbert space*. Both the Euclidean n-space R^n, and $L_2[a,b]$ space are Hilbert spaces. For the space R^n, the inner product of two vectors $x = (x_1, x_2, \cdots, x_n)^T$ and $y = (y_1, y_2, \cdots, y_n)^T$ is defined as $\langle x, y\rangle = \sum_{i=1}^{n} x_i y_i$; and the induced norm for vector x is $\|x\| = \sqrt{\langle x, x\rangle}$. For the real space $L_2[a,b]$, the inner product is defined as

$$\langle x, y\rangle = \int_a^b x(t)y(t)dt.$$

Detailed discussions may be found in [54].

By *linear operator* we mean a linear transformation from a vector space to another; and denote it by $A : X \longmapsto Y$. The range of operator A, i.e., the set of images of A in Y, is denoted by $\Re(A)$. The set in X satisfying $Ax = \mathbf{0}$ is called nullspace of A and denoted by $\aleph(A)$. A *functional* is actually an operator which maps a vector space into the space of real (complex) scalars. The normed dual space of a vector space X is the space of all bounded linear functionals on X, and is denote by X^*. Then any linear functional can be regarded as the element of dual space X^*, and can be denoted by $x^* \in X^*$. This point of view then justifies the notation $x^*(x)$ and $< x, x^* >$ which means the value of functional x^* evaluated at x. All bounded linear operators from the normed space X into the normed space Y compose another normed space, and denoted by $B(X,Y)$. Based on the concept of the dual space, the adjoint can be defined as:

Definition A.1.2. *Let X and Y be normed spaces and $A \in B(X,Y)$. The adjoint operator $A^* : Y^* \longmapsto X^*$ is defined by $< x, A^*y^* > = < Ax, y^* >$.*

Furthermore, the dual space of a Hilbert space is itself. Thus, when X and Y in the above definition are Hilbert spaces, the adjoint operator A^* is acting on Y with image in X; therefore the relationship between the linear operator and its adjoint only involves two Hilbert spaces, and can be cast into

$$\langle Ax, y\rangle = \langle x, A^*y\rangle.$$

A.2 An Implicit Function Theorem

The concept of a differential in a normed vector space is generalized in definition (A.2.1) and (A.2.2). These concepts are important in the optimization theories.

Definition A.2.1. *Let $x \in D \subset X$, a vector space and h be arbitrary in X and let $T : D \subset X \longmapsto R \subset Y$, a normed space and α be a scalar. If the limit*

$$\delta T(x; h) = \lim_{\alpha \to 0} \frac{1}{\alpha}(T(x + \alpha h) - T(x)) \tag{A.2.1}$$

exists, it is called the Gateaux differential of T at x with increment h. If the limit exists for each $h \in X$, the transformation T is said to be Gateaux differentiable at x.

In the definition of *Gateaux differential*, the concept of directional derivative in finite-dimensional space is generalized; its definition requires no norm on X. When the vector space X is equipped with a norm, we will have the following definition which is stronger than *Gateaux differential*.

Definition A.2.2. *Let T be a transformation defined on an open domain D in a normed space X and having range in a normed space Y. If for fixed $x \in D$ and each $h \in X$ there exists $\delta T(x; h) \in Y$ which is linear and continuous with respect to h such that*

$$\lim_{\|h\| \to \boldsymbol{0}} \frac{\|T(x + h) - T(x) - \delta T(x; h)\|}{\|h\|} = 0$$

Then T is said to be Fréchet differentiable at x and $\delta T(x; h)$ is said to be the Fréchet differential of T at x with increment h.

It can be easily proven that if the *Fréchet differential* exits, it is unique; and the *Fréchet differential* and *Gateaux differential* are related by the proposition that the *Gateaux differential* exists if the *Fréchet differential* exists, and they are equal.

Theorem A.2.1. *Let X and Y be Banach spaces and let T be a continuously Fréchet differentiable transformation from an open set D in $X \times Y$ with values in X. Let (x_0, y_0) be a point in D for which $T(x_0, y_0) = \boldsymbol{0}$ and for which $[T_x(x_0, y_0)]^{-1}$ exists. Then there is a neighborhood N of y_0 and a Fréchet differentiable transformation F mapping N into X such that $F(y_0) = x_0$ and $T(F(y), y) = \boldsymbol{0}$ for all $y \in N$.*

A.3 A Lagrange Multiplier Theorem

The theorem of Lagrange multipliers states the necessary conditions for an optimization problem where a real-valued functional f on a Banach space X is to be minimized subject to an equality constraint $H(x) = \mathbf{0}$. H is a mapping from X into a Banach space Z; it can be a set of algebraic equations, ordinary differential equations, or partial differential equations. In our applications, the constraint commonly takes the form of a boundary value problem (BVP).

Theorem A.3.1. *If the continuously Fréchet differentiable functional f has a local extremum under the constraint $H(x) = \boldsymbol{0}$ at the regular point x_0, then there exists an element $z^* \in Z^*$ such that the Lagrangian functional*

$$L(x) = f(x) + z_0^* H(x)$$

is stationary at x_0, i.e., $f_x(x_0) + z_0^* H_x(x_0) = \boldsymbol{0}$, where $f_{(\cdot)}, H_{(\cdot)}$ are the Fréchet derivatives of the cost functional and the constraints, respectively, with respect to the subscripted variables.

In most optimization problems (just as in the optimization problem stated in the PMP), the *Banach* space X can be partitioned as $X = \mathcal{X} \times \mathcal{U}$, where $\mathcal{X}$ is *state* space, and $\mathcal{U}$ is *control* space. In optimal design applications the variables in $\mathcal{U}$ are sometimes called

design variables, or design parameters. The intuition is that the constraint $H = 0$ uniquely determines the state for given controls. However, when the constraints are adjoined to the cost functional to construct the Lagrangian functional, the dependence of the *states* on the *controls* should be disconnected and we view them as independent variables. The stationary conditions stated in Theorem (A.3.1) can be rewritten as

$$f_{\mathcal{X}}(\mathcal{X}_0;\mathcal{U}_0) + z_0^* H_{\mathcal{X}}(\mathcal{X}_0;\mathcal{U}_0) = \mathbf{0} \tag{A.3.2}$$

and

$$\frac{\partial L}{\partial \mathcal{U}} = f_{\mathcal{U}}(\mathcal{X}_0;\mathcal{U}_0) + z_0^* H_{\mathcal{U}}(\mathcal{X}_0;\mathcal{U}_0) = \mathbf{0}. \tag{A.3.3}$$

Furthermore, when the constraints $H(\mathcal{X};\mathcal{U})$ can be viewed as the governing equations from which the states $\mathcal{X}$ can be uniquely determined for a given set of design variables $\mathcal{U}$, and under the conditions that the constraint $H(x) = \mathbf{0}$ is *regular* at the optimal point x_0, the Jacobian $H_{\mathcal{X}}$ is invertible; thus the Lagrange multiplier z_0^* can be solved from Eq. (A.3.2),

$$z_0^* = -f_{\mathcal{X}}(\mathcal{X}_0;\mathcal{U}_0)[H_{\mathcal{X}}(\mathcal{X}_0;\mathcal{U}_0)]^{-1}.$$

In many cases, z^* is also called the adjoint, and Eq. (A.3.2) is therefore called the adjoint equation. Then from Eq. (A.3.3), we have

$$\frac{\partial L}{\partial \mathcal{U}} = f_U(\mathcal{X}_0;\mathcal{U}_0) - f_{\mathcal{X}}(\mathcal{X}_0;\mathcal{U}_0)[H_{\mathcal{X}}(\mathcal{X}_0;\mathcal{U}_0)]^{-1} H_{\mathcal{U}}(\mathcal{X}_0;\mathcal{U}_0) = \mathbf{0}.$$

The above necessary stationary conditions are formulated at the local extremum point.

When an iterative approach is employed to find the local extremum point, the stationary conditions and the constraint may or may not be satisfied at intermediate points. At these points, we assume that the adjoints can be solved from

$$f_{\mathcal{X}}(\mathcal{X};\mathcal{U}) + z H_{\mathcal{X}}(\mathcal{X};\mathcal{U}) = \mathbf{0},$$

and yields

$$z = -f_{\mathcal{X}}(\mathcal{X};\mathcal{U})[H_{\mathcal{X}}(\mathcal{X};\mathcal{U})]^{-1}.$$

and the reduced gradient of the cost functional with respect to the controls is given by,

$$\frac{\partial L}{\partial \mathcal{U}} = f_{\mathcal{U}}(\mathcal{X};\mathcal{U}) - f_{\mathcal{X}}(\mathcal{X};\mathcal{U})[H_{\mathcal{X}}(\mathcal{X};\mathcal{U})]^{-1} H_{\mathcal{U}}(\mathcal{X};\mathcal{U}). \tag{A.3.4}$$

In the above formulation, $f_{\mathcal{U}}(\mathcal{X};\mathcal{U})$ is the derivatives of the cost functional with respect to the controls at $frozen$ states; $\frac{\partial L}{\partial \mathcal{U}}$ is the gradient of the cost functional with respect to the controls at constant value of H. This is called the *reduced* gradient. Note that we have

$$\delta H = H_{\mathcal{X}} \delta \mathcal{X} + H_{\mathcal{U}} \delta \mathcal{U}.$$

To maintain constant H, the changes in states, to the first order, are related to changes in controls by

$$\delta \mathcal{X} = -H_{\mathcal{X}}^{-1} H_{\mathcal{U}} \delta \mathcal{U}.$$

Thus, we have

$$\begin{aligned} \delta f &= f_{\mathcal{U}} \delta \mathcal{U} + f_{\mathcal{X}} \delta \mathcal{X} \\ &= (f_{\mathcal{U}} - f_{\mathcal{X}} H_{\mathcal{X}}^{-1} H_{\mathcal{U}}) \delta \mathcal{U}, \end{aligned} \tag{A.3.5}$$

or

$$\frac{\partial f}{\partial \mathcal{U}} = f_{\mathcal{U}} - f_{\mathcal{X}} H_{\mathcal{X}}^{-1} H_{\mathcal{U}}.$$

This result is identical to what we have obtained in (A.3.4), and therefore justifies the interpretation of the reduced gradient as the gradient of the cost functional with respect to the control variables subject to the constraint.

A.4 Pontryagin's Minimum Principle for Parameter Optimization

This principle states the necessary conditions for a optimal control problem governed by a system of first-order ordinary differential equations [1],

$$\dot{x}(t) = f(x(t), u(t), w),$$

where $x(t) \in R^n$ is the *state* vector, $u(t) \in R^m$ is the *control* vector, $w \in R^l$ is a constant parameter vector. An initial set Θ^0 and a target set Θ^1 are given by the intersection of p, q smooth surfaces, respectively, i.e.

$$\Theta^0 = \{x \in R^n | \phi_i^0(x) = 0, i = 1, 2, \cdots, p\}$$
$$\Theta^1 = \{x \in R^n | \phi_i^1(x) = 0, i = 1, 2, \cdots, q\}$$

such that $x(t_0) \in \Theta^0$ and $x(t_1) \in \Theta^1$. In addition, the control should be piecewise continuous, with $u(t) \in \Omega \subset R^m$ for a given fixed set Ω. Given this mathematical structure, we are to minimize the cost functional

$$J(u(\cdot), x(\cdot), w) = \int_{t_0}^{t_1} L(x(t), u(t), w)dt$$

where $L : R^n \times R^m \times R^l \longmapsto R$ is the *Lagrange* function . This indicates that we need to find an admissible control $u^*(\cdot)$, a parameter vector w^* and the corresponding state-trajectory $x^*(\cdot)$ to yield a minimum value of the cost functional. Leitmann [53] cast this parameter optimization to a standard structure via treating the parameter as additional states, then applying a standard version of the Minimum Principle to it. Here we only refer to the necessary conditions derived by Leitmann.

Before the necessary conditions are stated, it is convenient to introduce the *variational Hamiltonian*

$$H(\lambda_0, \lambda, x, u, w) = \lambda_0 L(x, u, w) + \lambda^T f(x, u, w).$$

[1] In many treatments, the principle is referred to as Pontryagin's Maximum Principle. This follows from a different choice of sign on λ.

The real-valued functional H involves the inner product of adjoint vector λ and the RHS of the dynamic system; and the term $\lambda_0 L$ can be viewed as the zeroth coordinate in an $(n+1)$ dimensional space. Then we have :

If $(x^*(\cdot), u^*(\cdot), w^*)$ is optimal for the control problem, then there exists a real scalar λ_0 and a vector function $\lambda(\cdot) : [t_0, t_1] \longmapsto R^n$, such that:

- $\lambda_0 \geqslant 0$
- $\dot{\lambda}(t) = -(\frac{\partial H}{\partial x})^T$
- $\lambda(t_0) \perp \Theta^0 \mid_{x(t_0)}$
- $\lambda(t_1) \perp \Theta^1 \mid_{x(t_1)}$
- $H(\lambda_0, \lambda(t), x^*(t), w^*, v) \geq H(\lambda_0, \lambda(t), x^*(t), w^*, u^*(t))$, for all $v \in \Omega$
- $H(\lambda_0, \lambda(t), x^*(t), w^*, u^*(t)) = 0$
- $\int_{t_0}^{t_1} \frac{\partial H}{\partial w} dt = \mathbf{0}$

In the last necessary condition, $\int_{t_0}^{t_1} \frac{\partial H}{\partial w} dt$ is actually the reduced gradient of the cost functional with respect to the parameter w. The initial *transversality* condition $\lambda(t_0) \perp \Theta^0 \mid_{x(t_0)}$ is essentially a geometric presentation, and can be interpreted as requiring the existence of scalars $\mu_i, i = 1, \cdots, p$ such that

$$\lambda(t_0) = \mu_1 \nabla \phi_1^0(x) + \cdots + \mu_1 \nabla \phi_p^0(x).$$

Similarly, the analytic statement to the terminal *transversality* condition $\lambda(t_1) \perp \Theta^1 \mid_{x(t_1)}$ is that there exist scalars $\nu_i, i = 1, \cdots, q$ such that

$$\lambda(t_1) = \nu_1 \nabla \phi_1^1(x) + \cdots + \nu_1 \nabla \phi_p^1(x).$$

In many cases, the constant $\lambda_0 \neq 0$ and therefore we can take $\lambda_0 = 1$. For the *pathological problems* in which $\lambda_0 = 0$ the reader is referred to [55].

Appendix B

Recursive Projection Method

In this method, the time integration scheme (3.4.29) is interpreted as a fixed point iteration

$$\Lambda^{n+1} = \mathcal{R}(\Lambda^n). \tag{B.0.1}$$

The convergence of this fixed point method depends crucially on the eigenvalues of Jacobian $\mathcal{R}_\Lambda$. Slow convergence rate or even divergence of (B.0.1) is accredited to one or more eigenvalues of $\mathcal{R}_\Lambda$ approaching the unit disk or lying outside of the unit disk. In this case, Shroff and Keller [79] split the whole space into two subspaces $\mathcal{P}$ and $\mathcal{Q}$, one spanned by those directions in which $\mathcal{R}$ is either slowly contracting of not contracting at all, the other orthogonal to the former:

$$\mathcal{P} \equiv \text{maximal invariant subspace of } \mathcal{R}_\Lambda \text{ belonging to unstable eigenvalues}$$

and

$$\mathcal{Q} \equiv \text{orthogonal complement of } \mathcal{P}.$$

Denote orthogonal projectors of $\Re^N$ onto the subspaces $\mathcal{P}$,$\mathcal{Q}$ by $\mathbf{P}$ and $\mathbf{Q}$, then

$$\Re^N = \mathcal{P} \oplus \mathcal{Q} = \mathbf{P}\Re^N \oplus \mathbf{Q}\Re^N.$$

Therefore, we get a unique decomposition for $\Lambda^{n+1} \in \Re^N$,

$$\begin{aligned} \Lambda^{n+1} &= \mathbf{P}\mathcal{R}(\Lambda^n) + \mathbf{Q}\mathcal{R}(\Lambda^n) \\ &= r(\mathbf{p}^n, \mathbf{q}^n) + s(\mathbf{p}^n, \mathbf{q}^n) \\ &= \mathbf{p}^{n+1} + \mathbf{q}^{n+1}. \end{aligned} \tag{B.0.2}$$

Shroff & Keller suggest that one use a fixed-point scheme in $\mathcal{Q}$

$$\mathbf{q}^{n+1} = s(\mathbf{p}^n, \mathbf{q}^n) \tag{B.0.3}$$

and use Newton's method to solve $\mathbf{p} = r(\mathbf{p}, \mathbf{q})$ to stabilize the iteration in $\mathcal{P}$

$$(I - r_{\mathbf{p}}(\mathbf{p}, \mathbf{q}))(\mathbf{p}^{n+1} - \mathbf{p}^n) = r(\mathbf{p}^n - \mathbf{q}^n) - \mathbf{p}^n. \tag{B.0.4}$$

To estimate the projectors $\mathbf{P}$ and $\mathbf{Q}$, we monitor the rate of convergence of the iterations $\mathbf{q}^n \in \mathcal{Q}$ which are approximately power iterations to the second-order accuracy. To do this, a small window of r difference vectors

$$\Delta Q = \{\Delta \mathbf{q}^i\}_n^{n-r+1}$$

are accumulated while carrying out iterations. Then we find the singular value decomposition to $\Delta Q \in \Re^{N \times r}$,

$$\Delta Q = \mathbf{U}\Sigma\mathbf{V}^T, \tag{B.0.5}$$

where $\mathbf{U} = [\mathbf{U}_1, \ldots, \mathbf{U}_N] \in \Re^{N \times N}$ and $\mathbf{V} = [\mathbf{V}_1, \ldots, \mathbf{V}_r] \in \Re^{r \times r}$ are orthogonal matrices, and non-negative diagonal matrix $\Sigma \in \Re^{N \times r}$ is given by

$$\Sigma = \begin{pmatrix} \sigma_1 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ 0 & \sigma_2 & \ldots & 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & \sigma_r & \ldots & \ldots & 0 \\ 0 & 0 & \ldots & \ldots & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \ldots & \ldots & \ldots & 0 \end{pmatrix} \tag{B.0.6}$$

with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$. ($\sigma_i, i = 1, 2, \ldots, r$ are called singular values.) If σ_1 is significantly larger than σ_2 ($\sigma_1 \geq 10\sigma_2$ in practice), we choose $Z = \mathbf{U}_1$ as the orthogonal basis for subspace $\mathcal{P}$, otherwise we let $Z = [\mathbf{U}_1, \mathbf{U}_2]$. Thus, we get the corresponding projectors:

$$\mathbf{P} = ZZ^T \text{ and } \mathbf{Q} = I - ZZ^T.$$

The inverse of $I - r_{\mathbf{p}}(\mathbf{p}, \mathbf{q}))$ needed in Newton's method (B.0.4) can be shown to be

$$(I - r_{\mathbf{p}}(\mathbf{p}, \mathbf{q}))^{-1} = Z(I - Z^T\mathcal{R}_\Lambda Z)^{-1}Z^T.$$

After introducing the coordinate variable $z = Z^T\mathbf{p}$ to represent $\mathbf{p} \in \mathcal{P}$ in the basis Z, the iterations in $\mathcal{P}$ can be reduced to

$$z^{n+1} = z^n + (I - Z^T\mathcal{R}_\Lambda Z)^{-1}(Z\mathcal{R}(\Lambda^n) - z^n).$$

The finite difference method is applied to approximate $\mathcal{R}_\Lambda Z$ needed in the above equation as following

$$\mathcal{R}_\Lambda Z_j \approx \tfrac{1}{\epsilon}[\mathcal{R}(\Lambda + \epsilon Z_j) - \mathcal{R}(\Lambda)],$$

where Z_j is the j^{th} column of Z.

B.1 Numerical Experiments

The RPM is applied in numerically solving the adjoint equation (3.2.19) on a non-uniform grid system with 81 grid points. Fig. (B.1) and (B.2) show the comparisons of convergence histories with and without RPM for the two cases of different CFL number; $CFl = 0.95$ and $CFL = 1.15$. In both cases RPM is only applied at three given time steps $Iter = 1000, 1500, 2150$, at each of which basis Z is calculated independently. As displayed in these

figures, the slop of the residue is larger for the numerical method with RPM than without it; therefore the convergence is speeded up by this scheme. The following is the RPM algorithm employed in the numerical experiments.

Step 1: Initialize: n=0, Λ^0

Initilize Qs : Matrix to save previous q

Specify IterRPM: Time step to apply RPM

Step 2: while n $\leq IterMAX$

$Z = \varnothing$

if n == IterRPM, apply RPM

$[\mathbf{U}, \Sigma, \mathbf{V}] = svd(\Delta Q)$

Construct basis Z according to σ_1, σ_2

if$\sigma_1 \geq 10\sigma_2$

$Z = U_1$

else

$Z = [U_1 \quad U_2]$

endif

Using FDM to approximate$[\mathcal{R}_\Lambda Z]$

end if

if$Z == \varnothing$:Fixed Point Iterations

$\Lambda^{n+1} = \mathcal{R}(\Lambda^n)$

else: RPM

$\hat{H} = Z^T \mathcal{R}_\Lambda Z$

$z = Z^T \Lambda^n$

$q = \Lambda^n - Zz$

$\xi = Z' \mathcal{R}(\Lambda^n)$

$z = z + (I - \hat{H})^{-1}(\xi - z)$

$q = \mathcal{R}(\Lambda^n) - Z\xi$

$\Lambda^{n+1} = Zz + q$

end if

modify Qs, n = n+1

Convergence tests

end while

Step 3: end

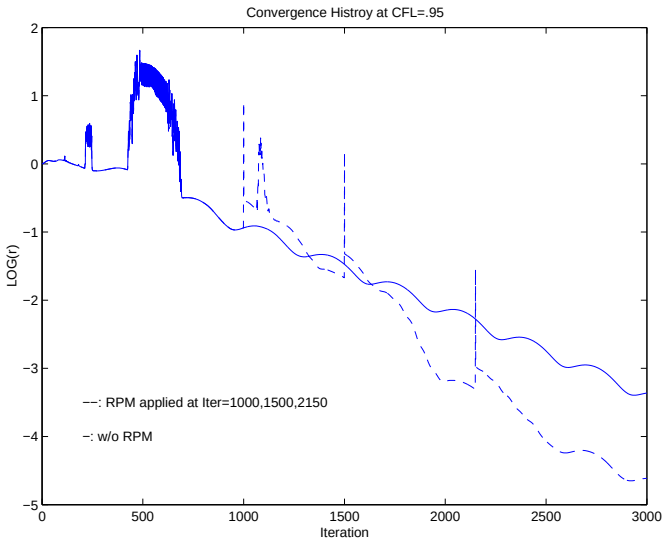

Figure B.1: Convergence history at $CFL = 0.95$

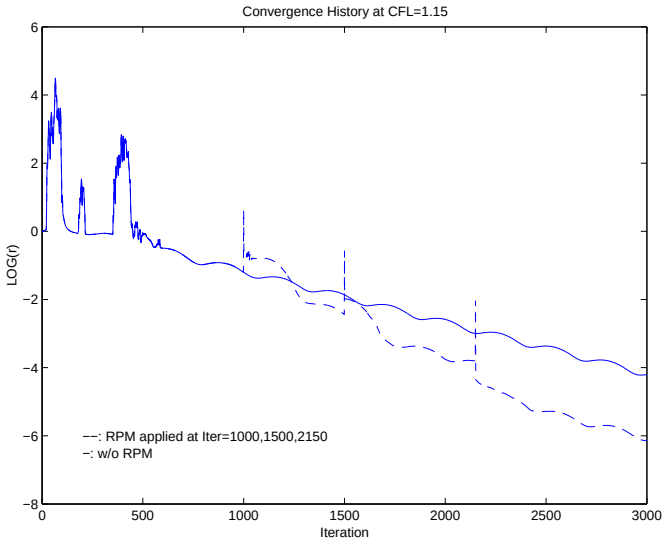

Figure B.2: Convergence history at $CFL = 1.15$

Bibliography

[1] I. H. Abbott and A. E. Doenhoff, *Theory of wing sections*, Dover Publications, Inc., New York, 1959.

[2] P. V. Aidala, W. H. Davis, and W. H. Mason, *Smart aerodynamic optimizaiton*, AIAA 83-1863, AIAA Applied Aerodynamics Conference, Danvers, Ma, July 13-15 1983.

[3] Z. Alsalihi, *Two dimensional hyperbolic grid generation*, Technical note 162, Von Karman Institute for Fluid Dynamics, Oct. 1987.

[4] W. K. Anderson, J. C. Newman, D. L. Whitfield, and E. J. Nielsen, *Sensitivity analysis for Navier-Stokes equations on unstructured meshes using complex variables*, AIAA Journal **39** (2001), no. 1, 56–63.

[5] W. K. Anderson, J. L. Thomas, and B. van Leer, *A comparison of finite volume flux vector splitting for the Euler equations*, AIAA 85-0122, AIAA 23rd Aerospace Sciences Meeting, Reno, Nevada, Jan. 14-17 1985.

[6] W. K. Anderson and V. Venkatakrishnan, *Aerodynamic design optimization on unstructured grids with continuous adjoint formulation*, AIAA 97-0643, 35th Aerospace Sciences Meeting, Jan. 1997.

[7] M. Asselin, *An introduction to aircraft performance*, Educational Series, AIAA, Inc., 1997.

[8] O. Baysal and K. Ghayour, *Continuous adjoint sensitivities for optimization with general cost functionals on unstructured meshes*, AIAA J. **39** (2001), no. 1, 48–55.

[9] J. Borggaard and J. Burns, *A PDE sensitivity equation method for optimal aerodynamic design*, J. Comput. Phys. **136** (1997), no. 2, 366–389.

[10] C. G. Broyden, *The convergence of a class of double-rank minimization algorithms*, J. Inst. Maths. Applics. **6** (1970), 76–90.

[11] A. E. Bryson and Y.-C. Ho, *Applied optimal control*, Hemisphere Publishing Corporation, 1975.

[12] G. W. Burgreen and O. Baysal, *Three-dimensional aerodynamic shape optimization of wings using sensitivity analysis*, AIAA 94-0094, 32nd Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 1994.

[13] R. L. Campbell and L. A. Smith, *A hybrid algorithm for transonic airfoil and wing design*, AIAA 87-2552, AIAA 5th Applied Aerodyamic Conference, Monterey, CA, Aug. 17-19 1987.

[14] W. M. Chan and J. L. Steger, *A generalized scheme for three-dimensional hyperbolic grid generation*, AIAA 91-1588, AIAA 10th Computational Fluid Dynamics Conference, June 1991.

[15] E. M. Cliff, M. Heinkenschloss, and A. Shenoy, *Optimal control for flows with discontinuities*, Tr 95-09-02, Interdisciplinary Center for Applied Mathematics, Virginia Tech, Blacksburg, VA, 1995.

[16] ———, *On the optimality system for a 1-D Euler flow problem*, AIAA 96-3993, 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Bellevue, WA, Sept. 1996.

[17] A. Dadone and B. Grossman, *Progressive optimization of inverse fluid dynamic design problems*, Computers & Fluids **29** (2000), 1–32.

[18] W. H. Davis, *Technique for developing design tools from the analysis methods of computational aerodynamics*, AIAA 79-1529, AIAA 12th Fluid and Plasma Dynamics Conference, Williamsburg, VA, 1979.

[19] J. E. Dennis and R. B. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[20] M. Van Dyke, *Perturbation methods in fluid mechanics*, annotated ed., The Parabolic Press, Standford, CA, 1975.

[21] A. Eberle, A. Rizzi, and E. H. Hirschel, *Numerical solutions of the Euler equations for steady flow problems*, Notes on Numerical Fluid Mechanics, vol. 34, Vieweg, 1992.

[22] T. A. Edwards, *Noniterative three-dimensional grid generation using parabolic partial differential equations*, AIAA 85-0485, AIAA 23rd Aerospace Sciences Meeting, Reno, Nevada, Jan. 14-17 1985.

[23] R. Fletcher, *A new approach to variable metric algorithms*, Computer J. **13** (1970), 317–322.

[24] P. D. Frank and G. R. Shubin, *A comparison of optimization-based approaches for a model computational aerodynamics design problem*, J. Comput. Phys. (1992), no. 98, 74–89.

[25] J. W. Gallman, J. J. Reuther, and et al., *Business jet wing design using aerodynamic shape optimization*, AIAA 96-0554, AIAA 34th Aerospace Sciences Meeting and Exhibit, Reno, NV, Jan. 15-18 1996.

[26] M. B. Giles and N. A. Pierce, *Adjoint equations in CFD: duality, boundary conditions and solutions behavior*, AIAA 97-1850, AIAA 13th Computational Fluid Dynamics Conference, Snowmass Village, CO, June 29-July 2 1997.

[27] ———, *On the properties of solution of the adjoint Euler equations*, Tech. report, Numerical methods for fluid dynamics VI, 1998.

[28] _____, *Analytic adjoint solution for the quasi-one-dimensional Euler equations*, J. Fluid Mech. **426** (2001), 327–345.

[29] P. Gill, W. Murray, and M. Wright, *Practical optimization*, Academic Press, New York, 1981.

[30] P. E. Gill and W. Murray (eds), *Numerical methods for constrained optimization*, Academic Press, 1974.

[31] D. Goldfarb, *A family of variable-metric methods derived by variational means*, Math. Comput. **24** (1970), 23–26.

[32] L. L. Green, P. A. Newman, and K. J. Haigler, *Sensitivity derivatives for advanced CFD algorithm and viscous modeling parameters via automatic differentiation*, J. Comput. Phys **125** (1996), 313–324.

[33] R. M. Hicks and P. A. Henne, *Wing design by numerical optimization*, AIAA 77-1247, AIAA Aircraft System and Technology Conference, Seattle, Wash., Aug. 22-24 1977.

[34] K. A. Hoffmann, *Computational fluid dynamics for engineers*, Engineering Education System, Austin, 1989.

[35] K. A. Hoffmann, W. H. Rutledge, and P. E. Rodi, *Hyperbolic grid generation techniques for blunt body*, Numerical grid generation in computational fluid dynamics, Pineridge Press, 1988.

[36] T. L. Holst and T. H. Pulliam, *Aerodynamic shape optimization using a real-number-encoded genetic algorithm*, AIAA 2001-2473, AIAA 19th Applied Aerodynamics Conference, Anaheim, CA, June 11-14 2001.

[37] A. H. Ibrahim, S. N. Tiwai, and R. E. Smith, *Stability analysis and aerodynamic design optimization of Euler equations using variational methods*, Int. J. Numer. Meth.Engng. **44** (1999), 1709–1726.

[38] AeroSoft Inc., $\mathcal{SENSE}$ *user's manual*, Blacksburg, VA.

[39] ———, *Aerosoft 3.0 user's manual*, Blacksburg, VA, 1996.

[40] A. Iollo, M. D. Salas, and S. Ta'asan, *Shape optimization governed by the Euler equations using adjoint method*, Tech. Report 93-78, ICASE, NASA Langley Research Center, Hampton, VA.

[41] M. Heinkenschloss J. E. Dennis and L. N. Vicente, *Trust-region interior-point SQP algorithm for a class of nonlinear programming problems*, SIAM J. Control Optim. **36** (1998), no. 5, 1750–1794.

[42] A. Jameson, *Numerical solution of the Euler equations for compressible inviscid fluid*, Numerical methods for the Euler equations of fluid dynamics, SIAM, 1985.

[43] ———, *Airfoils admitting non-unique solutions of the Euler equations*, AIAA 91-1625, AIAA 22nd Fluid Dynamics, Plasmadynamics & Lasers Conference, Honolulu, HI, June 24-26 1991.

[44] ———, *Optimum aerodynamic design using CFD and control theory*, AIAA 95-1729, 12th AIAA Computational Fluid Dynamic Conference, San Diego, CA, June 19-22 1995.

[45] A. Jameson and T. J. Baker, *Solution of the Euler equations for complex configurations*, AIAA 83-1929, AIAA 6th Computational Fluid Dynamics Conference, Danvers, MA, July 1983.

[46] A. Jameson, N. A. Pierce, and L. Martinelli, *Optimum aerodynamic design using the Navier-Stockes equations*, AIAA 97-0101, 35th Aerospace Sciences Meeting, Reno, NV, Jan. 1997.

[47] A. Jameson and J. C. Vassberg, *Computational fluid dynamics for aerodynamic design: its current and future impact*, AIAA 2001-0583, 39th AIAA Aerospace Sciences Meeting & Exhibit, Reno, NV, June 8-11 2001.

[48] J. E. Dennis, Jr., M. El-alem, and M. C. Maciel, *A global convergence theory for general trust-region-based algorithms for equality constrained optimization*, SIAM J. Optim. **7** (1997), no. 1, 177–207.

[49] H.-J. Kim and O.-H. Rho, *Dual-point design of transonic airfoils using the hybrid inverse optimization method*, Journal of Aircraft **34** (1997), no. 5, 612–618.

[50] S. Kim, J. J. Alonso, and A. Jameson, *Two-dimensional high-lift aerodynamic optimization using the continuous adjoint method*, AIAA 2000-4741.

[51] Th. W. Labrujère and J. W. Slooff, *Computational methods for the aerodynamic design of aircraft components*, Annu. Rev. Fluid Mech. **25** (1993), 183–214.

[52] K. D. Lee and S. Eyi, *Transonic airfoil design by constrained optimization*, AIAA 91-3287, AIAA 9th Applied Aerodynamics Conference, Baltimore, MD, Sept. 23-25 1991.

[53] G. Leitmann, *The calculus of variations and optimal control: An introduction*, Mathematical concepts and methods in science and engineering, vol. 24, Plenum Press, 1981.

[54] J. L. Lions, *Optimal control of systems governed by partial differential equations*, Springer-Verlag, 1971.

[55] A. Locatelli, *Optimal control: an introduction*, Birkhauser Verlag, 2001.

[56] D. G. Luenberger, *Optimization by vector space methods*, John Wiley & Sons, Inc., 1969.

[57] W. H. Mason, D. L. Knill, A. A. Giunta, and R. T. Haftka, *Getting the full benefits of CFD in conceprual design*, AIAA 98-2513, 16th AIAA Applied Aerodynamics Conference, Albuquerque, NM, June 15-18 1998.

[58] K. Matsuno, *High-order upwind method for hyperbolic grid generation*, Computers & Fluids **28** (1999), 825–851.

[59] S. Nadarajah and A. Jameson, *A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization*, AIAA 2000-0667, 38th Aerospace Sciences Meeting & Exhibit, Reno, NV, Jan. 10-13 2000.

[60] S. Nadarajah, A. Jameson, and J. Alonso, *An adjoint method for the calculation of remote sensitivities in supersonic flow*, AIAA 2002-0261, 40th Aerospace Sciences Meeting & Exhibit, Reno, NV, Jan 14-17 1996.

[61] S. Nakamura, *Marching grid generation using parabolic partial differential equations*, Symposium on the Numerical Generation of Curvilinear Coordinate Systems and their Use in the Numerical Solution of Partial Differential Equations, Nashville, Tenn., April 1982.

[62] S. Nakamura and M. Suzuki, *Noniterative three dimensional grid generation using a parabolic-hyperbolic hybrid scheme*, AIAA 87-0277, AIAA 25th Aerospace Sciences Meeting, Reno, Nevada, Jan. 12-15 1987.

[63] R. Narducci, B. Grossman, and R. T. Haftka, *Sensitivity algorithms for an inverse design problem involving a shock wave*, Inverse Problems in Engineering **2** (1995), 49–83.

[64] R. Narducci, B. Grossman, M. Valorani, A. Dadone, and R. T. Haftka, *Optimization methods for non-smooth or noisy objective functions in fluid design problems*, AIAA 95-1648, AIAA 12th Computational Fluid Dynamics Conference, San Diego, CA, June 19-22 1995.

[65] A. Oloso, *Aerodynamic design sensitivities by automatic differentiation*, AIAA 98-2536, AIAA 29th Fluid Dynamics Conference, Albuquerque, NM, Jun. 1998.

[66] N. A. Pierce and M. B. Giles, *Adjoint recovery of superconvergent functionals from PDE approximation*, SIAM Review **42** (2000), no. 2, 247–264.

[67] O. Pironneau, *On optimum profiles in Stokes flow*, J. Fluid Mech. **59** (1973), no. 1, 117–128.

[68] ———, *On optimum design in fluid mechanics*, J. Fluid Mech. **64** (1974), no. 1, 97–110.

[69] M. M. Putko, A. C. Taylor III, P. A. Newman, and L. L. Green, *Some advanced concepts in discrete aerodynamic sensitivity analysis*, AIAA 2001-2529, AIAA 15th Computational Fluid Dynamic Conference, Anahelm, CA, Jun. 11-14 2001.

[70] J. N. Reddy, *Applied functional analysis and variational methods in engineering*, McGraw-Hill Book Company.

[71] J. Reuther and A. Jameson, *Aerodynamic shape optimization of wing and wing-body configurations using control theory*, AIAA 95-0123, AIAA 33rd Aerospace Sciences Meeting and Exhibit, Reno, NV, Jan. 9-12 1995.

[72] J. Reuther, A. Jameson, J. J. Alonso, M. J. Rimlinger, and D. Saunders, *Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers*, AIAA 97-0103, 35th Aerospace Sciences Meeting & Exhibit, Reno, NV, Jan. 1997.

[73] J. J. Reuther, *Aerodynamic shape optimization using control theory*, Ph.D. thesis, University of California, Davis, 1996.

[74] L. C. Santos, *A hybrid inverse optimization method for the aerodynamic design of lifting surfaces*, AIAA 94-1895, AIAA 12th Applied Aerodynamics Conference, Colorado Springs, CO, 1994.

[75] D. F. Shanno, *Conditioning of quasi-Newton methods for function minimization*, Math. Comput. **24** (1970), 647–657.

[76] A. Shenoy and E. M. Cliff, *An optimal control formulations for a flow matching problem*, AIAA 94-4306, Proceedings from 5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Panama City Beach, Florida, Sept. 1994.

[77] A. Shenoy, M. Heinkenschloss, and E. M. Cliff, *Airfoil design by an all-at-once method*, IJCFD **11** (1998), 3–25.

[78] A. R. Shenoy, *Optimization techniques exploiting problems structure: application to aerodynamic design*, Ph.D. thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA 24060, 1998.

[79] G. M. Shroff and H. B. Keller, *Stabilization of unstable procedures: the recursive projection method*, SIAM J. Numer. Anal. **30** (1993), no. 4, 1099–1120.

[80] J. Sokolowski and J.-P. Zolesio, *Introduction to shape optimization*, Springer-Verlag, 1992.

[81] G. Starius, *Constructing orthogonal curvilinear meshes by solving initial value problems*, Numer. Math. **28** (1977), 25–48.

[82] J. L. Steger and D. S. Chaussee, *Generation of body-fitted coordinates using hyperbolic partial differential equations*, SIAM J. of Sci. Stat. & Comput. **1** (1980), 431–437.

[83] J. L. Steger and R. L. Sorenson, *Use of hyperbolic differential equations to generate body fitted coordinates*, Numerical grid generation techniques (1980).

[84] P. Sundaram, S. Agrawal, and J. O. Hager, *Aerospace vehicle MDO shape optimization using ADIFOR 3.0 gradients*, AIAA 2000-4733, 8th AIAA/USAF/NASA/ISSMO Symposium Multidisciplinary Analysis and Optimization, Long Beach, CA, Sept. 6-8 2000.

[85] C. H. Tai, S. L. Yin, and C. Y. Soong, *A novel hyperbolic grid generation procedure with inherent adaptive dissipation*, J. Comp. Phys. **116** (1995), 173–179.

[86] S. Takanashi and M. Takemoto, *An automatic grid generation procedure for complex aircraft configurations*, Computers & Fluids **24** (1995), no. 4, 393–400.

[87] J. C. Tannehill, D. A. Anderson, and R. H. Pletcher, *Computational fluid mechanics and heat transfer*, Taylor & Francis, 1997.

[88] J. F. Thompson, Z. Warsi, and C. W. Mastin, *Numerical grid generation: foundations and applications*, North-Holland, 1985.

[89] É. Turgeon, D. Pelletier, and J. Borggaard, *A general continuous sensitivity equation formulation for complex flows*, AIAA 2000-4732, 8th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, CA, Sept. 6-8 2000.

[90] G. N. Vanderplaats, *Efficient algorithm for numerical airfoil optimization*, J. Aircraft **16** (1979), no. 2, 842–847.

[91] ———, *Numerical optimization techniques for engineering design*, second ed., McGraw-Hill, New York, 1998.

[92] A. Vardi, *A trust region algorithm for equality constrained minimization: convergence problems and implementation*, SIAM J. Numer. Anal. **22** (1985), no. 3, 575–591.

[93] D. A. Venditti and D. L. Darmofal, *Adjoint error estimation and grid adaptation for functional outputs: Application to quasi-one-dimensional flow*, J. Compt. Phys. **164** (2000), 204–227.

[94] Y.-X. Yuan, *On the convergence of a new trust region algorithm*, Numer. Math. **70** (1995), 515–539.

Index

Vita

Lei Xie was born on Oct. 14, 1974 in a small town in China. He received a BS degree from the department of aircraft design and engineering mechanics at Beijing University of Aeronautics and Astronautics in July, 1995. Then he worked for Research & Design Institute of Unmanned Aerial Vehicle during the next three years. In Aug. 1998 he came to Aerospace & Ocean department of VPI&SU in pursuit of a Ph.D. degree in Aerospace Engineering.