Towards Exploratory Aerodynamic Design using the Reynolds-Averaged Navier-Stokes Equations

by

David Tai Shun Koo

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

Abstract

Towards Exploratory Aerodynamic Design using the Reynolds-Averaged Navier-Stokes Equations

David Tai Shun Koo

Master of Applied Science

Graduate Department of Aerospace Science and Engineering

University of Toronto

2015

The aerodynamic optimization framework Jetstream is applied to problems involving lift-constrained drag minimization using the Reynolds-averaged Navier-Stokes equations. A parallel Newton-Krylov algorithm is used to solve the governing equations on multiblock structured meshes; gradients are computed using the discrete-adjoint method. Geometry parameterization and mesh movement are integrated using B-spline control volumes. Drag minimization studies from past works are revisited and strategies are devised to improve optimization convergence. These strategies include linear constraints for geometric feasibility, robust flow solver parameters, and meshing with an O-O topology. The single-point and multi-point optimization of the NASA Common Research Model (CRM) wing geometry is presented. A rectangular NACA0012 wing is optimized with planform design variables, enabling significant changes in span, sweep, taper, and airfoil section. To demonstrate Jetstream's flexibility, a wing based on the B737-900 is optimized with nonplanar winglets, split-tip, and wingtip fence configurations. Finally, the box-wing optimization in subsonic flow is revisited.

Acknowledgements

I would like to thank my supervisor, Professor Zingg, for the support and advice he has provided me during my studies at UTIAS. He has helped keep me on track on numerous occasions while giving me the freedom to pursue my research interests. I am also thankful to the members of my research assessment committee, Professor Steeves and Professor Nair.

I extend my gratitude to the students at UTIAS for making my time here enjoyable. I'll make special mention to my labmates Shahriar, Jenmy, Chris, Julia, Tom, Hugo, Lana, and Howard for the many insightful discussions and debates. I also have to mention my old classmates from Engineering Science who have kept in touch - though I won't name you individually, your friendship has helped keep my spirits high throughout these two years.

Finally, I thank the Natural Sciences and Engineering Research Council of Canada, the Government of Ontario, and the University of Toronto for financial support during my studies.

David Tai Shun Koo

University of Toronto Institute for Aerospace Studies

September 2015

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

The two biggest hurdles facing the aviation industry in the 21^{st} century are peak oil production and the phenomenon of anthropogenic global warming. Peak oil refers to the impending problem of global oil demand outpacing the production of fossil fuels. The resulting scarcity has led to rising fuel prices, providing a financial incentive for the aviation industry to develop more efficient aircraft. Anthropogenic global warming is the gradual rise in global temperatures due to the production of CO_2 and other greenhouse gases from human activity. The resulting change in the global climate may have devastating consequences, including species extinction, destructive weather events, and coastal flooding. At the same time, the growth in demand for aviation is compounding these two problems. It is evident that the aircraft of the future must be more fuel efficient than ever before.

As prototyping and wind tunnel testing can be slow and costly, aerodynamic shape optimization (ASO) has emerged as powerful new tool for aerodynamicists. While conventional methods of aircraft design rely on engineering intuition and iteration, aerodynamic optimization generates the best geometry based on the results of computational fluid dynamics. Numerical optimization is well suited for tasks such as tailoring wing surfaces to reduce wave drag or to perform better at off-design flight conditions. However, the more exciting aspect of numerical optimization is in the design of novel, unconventional configurations where there is less engineering experience in industry. Examples of unconventional geometries include closed wings,[8] nonplanar wingtips,[25] and blended wing bodies.[22] To perform this sort of exploratory design, an optimization tool must have a flexible geometry parameterization, a robust flow solver, and an efficient optimization algorithm. This thesis focuses on the development of our

in-house optimization methodology Jetstream and its application to a variety of aerodynamic optimization problems. Jetstream is a gradient-based, three-dimensional optimizer which employs a parallel Newton-Krylov algorithm to solve the equations governing compressible, turbulent flow.

1.2 Aerodynamic Shape Optimization

While computational fluid dynamics has a long history, aerodynamic shape optimization saw rapid development with the advent of adjoint methods by Pironneau,[30] and later Jameson.[15] These methods enable the sensitivity of the lift and drag forces with respect to an aerodynamic surface to be calculated almost independently of the number of design variables used to parameterize the geometry. As a result, it becomes more computationally efficient to design a wing with many variables, as finite-difference approximations require for too many function evaluations for just one optimization iteration. Examples of gradient-based methods include quasi-Newton BFGS or the sequential quadratic programming (SQP) methods. In contrast, genetic algorithms (GAs), mimic the natural selection process and evaluate many candidate designs before choosing one with the best aerodynamics. At the same time, the emergence of parallel computing has made it possible for optimization algorithms to scale up to improve efficiency.

The first applications of aerodynamic shape optimization were to two-dimensional airfoil design problems. Jameson and Reuther[17] developed an adjoint method to design airfoils using the Euler equations. Anderson and Bonhaus[1] incorporated the Spalart-Allmaras turbulence model to design airfoils on unstructured grids. Nemec et al.[23] presented a gradient-based Newton-Krylov method to design airfoils for multipoint and multi-objective optimization in turbulent flow. This was later extended by Driver and Zingg to model laminar-turbulent transition and design airfoils for natural laminar flow.[4]

Following the success of two-dimensional methods, the optimization of three-dimensional wings and full configurations became the next focus. Elliot and Peraire[6] were able to optimize three-dimensional wings as well as multi-element airfoils for the Euler equations, and later the laminar Navier-Stokes equations.[5] This was followed by the work of Nielson and Anderson,[24] who implemented the Spalart-Allamaras one-equation turbulence model to optimize on unstructured grids. At the same time, Jameson[16] developed the well-known code SYN107 for viscous aerodynamic optimization using his variation of the adjoint method. The three-dimensional Navier-Stokes code OPTIMAS, developed by Peigin and Epstein,[29] is an example of aerodynamic optimization method based on a genetic algorithm. A review published by Epstein et al.[7] compares some of these methods in the optimization of the geometry from the 3rd Drag Prediction Workshop.

For larger shape changes or unconventional configurations, aerodynamic shape optimization has

typically been performed using lower fidelity models. Jansen and Martins[18] used a panel method to compute induced drag for nonplanar lifting surfaces and used drag polar data to predict viscous drag. Ning and Kroo[25] used a vortex-lattice method for computing induced drag and assumed a parabolic relationship between viscous drag and lift in their numerical study on winglets. Using Jetstream, Hicken and Zingg[12] applied a high-fidelity Euler optimization to nonplanar wing geometries including the split tip configuration. This optimizer was later extended to include aerostructural constraints by Khosravi and Zingg[19] to study the net effect of winglets on a Boeing 737-900 wing.

Another topic that deserves mention is the possibility of a multimodal design space in aerodynamic optimization. Gradient-based methods offer the advantage of fast convergence to an optimum; however they may arrive at local minima since they attempt to satisfy the Karush-Kuhn-Tucker conditions. Genetic algorithms on the other hand, are more adept at locating global optima, but do so at the cost of much slower performance when there are many design variables.[33] In their study on multimodality of aerodynamic optimization, Chernuhkin and Zingg[3] showed genetic algorithms to be very impractical for three-dimensional design problems with high geometric flexibility. They developed a multi-start method combining the fast convergence capabilities of gradient methods with techniques to better explore the design space and identify global optima.

The work on which this thesis will build is the three-dimensional aerodynamic optimizer Jetstream, developed for the Euler equations by Hicken and Zingg[12] and later extended to RANS by Osusky et al.[26,27] By coupling a robust geometry parameterization and mesh movement scheme with an efficient RANS solver and adjoint code, Jetstream has the potential to perform high-fidelity aerodynamic design with significant geometry changes. In her thesis, Osusky applied the method to three main studies: the sectional optimization of the CRM wing geometry from the 5th Drag Prediction Workshop, the planform optimization of a rectangular NACA0012 wing, and to the optimization of winglets. The first of these cases was included as one of the benchmark cases for the Aerodynamic Design Optimization Discussion Group (ADODG) which convened first at the AIAA Scitech 2014 Conference,[31] and then again at Scitech 2015.[21]

Although each of the optimizations were successful in reducing drag, there were some issues found in the results. In the CRM case conducted by Osusky et al.,[26,27] some of the optimized sections showed a sharp hook-like feature at the leading edge - an unusual feature in airfoils which can lead to flow separation at flight conditions other than the optimal one. The second issue was the optimizations were plagued by software errors including mesh movement and solver problems. As a result, the gradient-based optimizer had difficulty exploring the design space and converging to an optimum solution. The final issue is that in the optimization of the CRM wing, optimizations with small variations of the initial

geometry converged to different geometries but with similar drag counts. This suggests the existence of either a multimodal or flat design space which warrants further investigation. In the optimization of the rectangular NACA0012 wing, Osusky was successful in showing the optimizer's ability to produce large scale planform changes when accounting for viscous and turbulent effects. However, these cases also showed poor convergence histories. Finally, the RANS optimization of the non-planar geometries - wingtip devices and box-wing configurations - showed small reductions in drag and unusual geometric features. Despite these issues, Osusky's work is extremely promising and warrants further development.

1.3 Objectives

The work of this thesis is an extension of the work of Osusky[27] to further the development of a high-fidelity numerical aerodynamic shape optimization algorithm based on the three-dimensional RANS equations and the Spalart-Allmaras turbulence model. To this end, the specific objectives of this thesis are

- to improve the performance of the CRM section optimization as well as the NACA0012 planform optimization, in other words, to obtain a further reduction in the optimality and objective function;
- to investigate the issue of multi-modality or flatness in the design space for a three-dimensional wing optimization based on the RANS equations;
- to apply the RANS-based optimizer to the design of non-planar geometries and wingtip devices.

1.4 Thesis Outline

Chapter 2 of this thesis will summarize the main theoretical concepts and algorithms used by Jetstream. The discretization and solution of the RANS equations is discussed, as well as B-spline volumes and gradient-based optimization. Chapter 3 will detail some of the changes and strategies implemented in this work to improve the performance of RANS optimization cases. Chapter 4 will outline the results of this thesis - each section will present the optimization problem and the initial and optimized geometries.

Chapter 2

Methodology

This chapter briefly summarizes the main theoretical concepts behind the optimization software Jetstream used in this thesis. The three main components discussed are the geometry parameterization, flow solver, and the gradient-based optimizer.

2.1 Geometry Parameterization and Mesh Movement

The integrated geometry parameterization and mesh movement scheme, developed originally by Hicken and Zingg,[12] uses B-spline volumes to form a control mesh for optimization. This section reviews some of the basic theory of B-spline volumes, as well as some of the additional steps required to fit meshes with turbulent node spacings. An overview of the geometry control methods is presented, followed by the mesh movement algorithm.

2.1.1 B-Spline Volumes

Prior to optimization, the multi-block computational mesh is fitted and approximated by a B-spline control mesh. The user specifies the number of B-spline points to use in the approximation in each direction for each particular block. The number of nodes in the control mesh is around two orders of magnitude fewer than the number of nodes in the computational mesh, allowing for a mesh deformation to be performed much more robustly and efficiently.

The B-spline volume maps the cubic domain in the computational space $D = \boldsymbol{\xi} = (\xi, \eta, \zeta) \in$

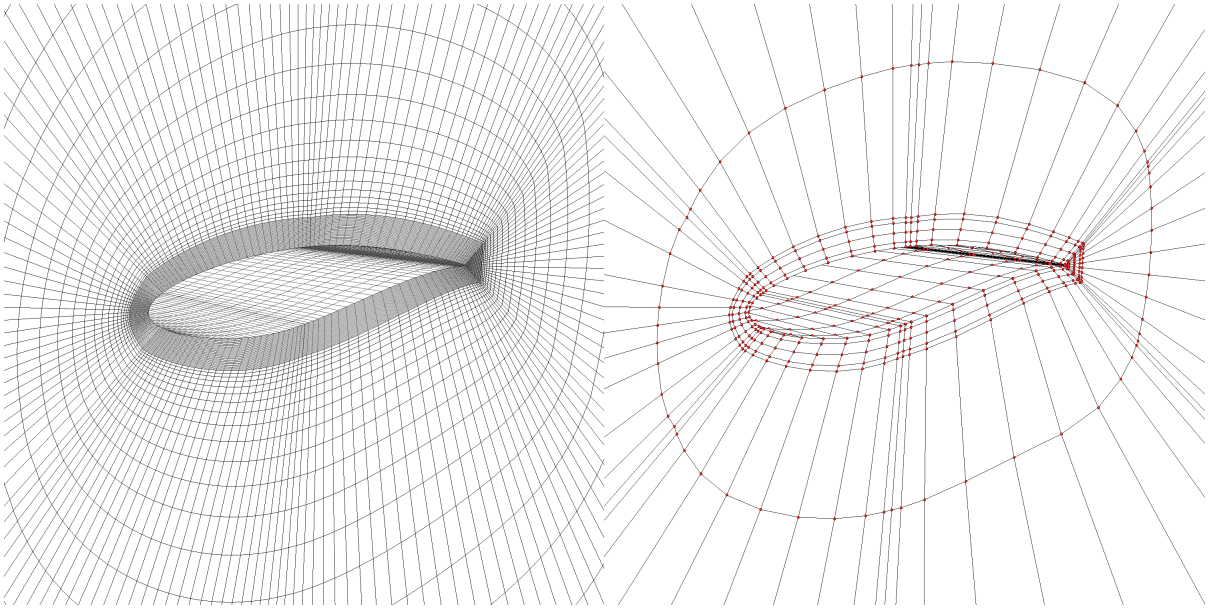


Figure 2.1: Example of B-spline mapping on a grid with RAE2822 sections. The computational grid is shown on the left, and the B-spline control mesh is on the right

$\mathbb{R}^3|\xi, \eta, \zeta \in [0, 1]$ to the physical space $P \subset \mathbb{R}^3$. It is defined as

$$\mathbf{x}(\boldsymbol{\xi}) = \sum_{i=1}^{N_i} \sum_{j=1}^{N_j} \sum_{k=1}^{N_k} \mathbf{B}_{ijk} \mathcal{N}_i(\xi) \mathcal{N}_j(\eta) \mathcal{N}_k(\zeta). \tag{2.1}$$

The points $\mathbf{B}_{ijk}$ are the coordinates of the control points, $\mathcal{N}$ are the B-spline basis polynomials of order p. The basis functions are joined at knot functions T_i and are C^{p-2} continuous at the knots. In this application, the B-spline polynomials used are 4th order ($p = 4$). In the ξ direction, the basis functions can be expressed as

$$\mathcal{N}_i^{(1)}(\xi; \eta, \eta) = \begin{cases} 1 \text{ if } T_i(\eta, \zeta) \leq \xi < T_{i+1}(\eta, \zeta), \\ 0 \text{ otherwise} \end{cases} \tag{2.2}$$

$$\mathcal{N}_i^{(p)}(\xi; \eta, \eta) = \frac{\xi - T_i(\eta, \zeta)}{T_{i+p-1}(\eta, \zeta) - T_i(\eta, \zeta)} \mathcal{N}_i^{p-1}(\xi; \eta, \zeta) + \frac{T_{i+p}(\eta, \zeta) - \xi}{T_{i+p}(\eta, \zeta) - T_{i+1}(\eta, \zeta)} \mathcal{N}_{i+1}^{p-1}(\xi; \eta, \zeta)$$

with similar expressions for the basis functions in the η and ζ directions. Another main advantage of creating a B-spline control mesh is that it preserves an analytical description of the aerodynamic surface, no matter how it is deformed. Figure 2.1 shows an example of a computational grid fitted with B-spline control points.

2.1.2 Fitting Meshes for Turbulent Flows

The B-spline volume fitting uses spatially varying knot vectors, causing the distribution of points in the B-spline control volume to resemble the distribution of nodes in the computational mesh. While this is typically a desired characteristic in meshes for inviscid flow, the clustering of the control points presents a problem when fitting to meshes with fine offwall spacings required to resolve the boundary layer for turbulent flows. When fitting grids with nodes between 10^{-7} to 10^{-6} reference units, the B-spline points have a tendency to cluster too closely, creating cells in the control mesh that have negative Jacobians.

To address this issue, a method known as parameter extraction is employed.[26] For optimization, two meshes are supplied to Jetstream. Both meshes are identical in terms of blocking, connectivity, and the number of nodes. The first mesh, known as the fitting grid, is used by the optimizer to fit B-spline volumes and has a minimum spacing of no less than 10^{-4} reference units. The second mesh, known as the fine spacing grid, has node spacings refined for turbulent flows. The fine spacing grid is read by the optimizer and used to extract parameter values along each edge. The parameter values are then applied to the B-spline volume mapping created from the fitting mesh. This approach avoids the difficulties associated with fitting excessively fine spacings as well as providing the user with an extra level of control by separating the clustering of the B-spline control points from the clustering of the computational grid nodes. While requiring two manually created meshes is more demanding on the user, in practice the fitting mesh can be created fairly easily. Once a turbulent mesh has been created and shown to be viable through some trial flowsolves, the fitting mesh can made in ICEM CFD simply by redistributing the nodes in the off-wall direction near the wing surface.

2.1.3 Surface-based and Volume-based Geometry Control

Jetstream uses two different methods of geometry control: surface-based and volume-based. The surface-based parameterization involves designating the coordinates of the B-spline control points as design variables. When a wing surface is subdivided over several grid blocks, the block faces corresponding to the surface are referred to as surface patches. When wing optimization is confined to manipulating twist and sectional shape variables, it is sufficient to designate the z-coordinates of the patch B-spline points as variables for the optimizer. This approach provides fine and localized surface control, but it has several drawbacks. First, surface optimization requires significant programming of linear constraints in order for the optimizer to maintain smooth B-spline surface transition across blocks. Furthermore, the programming of continuity and other constraints requires the user to have detailed knowlege of the wing topology, for example the H-H topology. As a result, it is difficult to generalize a particular approach to

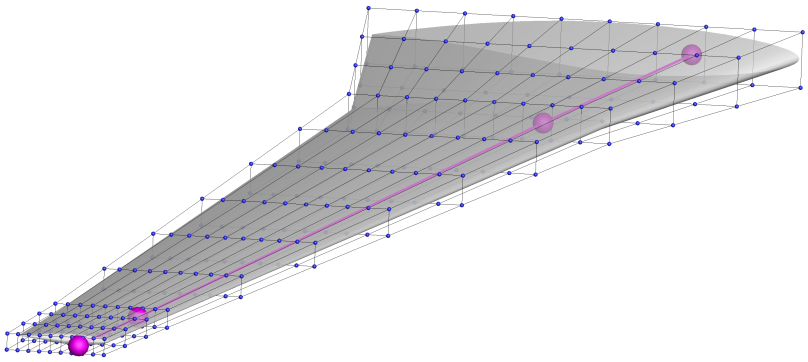


Figure 2.2: Example of wing surface embedded within an axial-curve-driven FFD volume. The FFD lattice points are shown in blue and the axial curve and its points are shown in magenta.

more complex geometries and blocking topologies. Finally, the surface-based approach requires extensive coupling and constraints in order to make more significant changes to wing planform such as span and dihedral.

Volume-based geometry control, borrowed from the animation industry, has been shown to work well for aerodynamic design.[2] Implemented into Jetstream by Gagnon and Zingg,[9] it addresses some of the shortcomings of the surface-based approach. The B-spline control points on the wing surface, regardless of topology, are embedded into a NURBS or Free-Form Deformation (FFD) volume. The FFD volume has its own lattice of control points that is user specified and can be created independently of how the multi-block wing mesh is created. As the outer lattice points of the FFD volume move, the embedded wing surface deforms intuitively and maintains its topological continuity. Another important characteristic is that the sensitivity of the aerodynamic surface has an exact mapping to the sensitivity with respect to the FFD control points, meaning that it does not introduce any gradient error into the optimization. The volume-based deformation method is further enhanced by the use of axial curves to drive larger changes in the FFD volume. A NURBS curve is defined as an axial curve and positioned within an FFD volume, for example at the quarter chord. Each FFD lattice point is then associated with a point on the axial curve. The axial curve can be thought of as the driving curve of the wing, controlling its span, sweep, and dihedral. By moving the axial points, larger deformations in the FFD volume can be achieved through a relatively small and intuitive set of design variables. For finer control of airfoil section, section twist, and taper, the FFD lattice points can move along their local z-axis orthogonal to the axial curve. Figure 2.2 shows a wing parameterized by an axial curve.

2.1.4 Mesh Movement Algorithm

Once the optimizer has updated the design variables, be it through surface-based or volume-based geometry control, the locations of the B-spline control points on the aerodynamic surface are updated. The changes in the surface control points are then propagated into the interior of the computational mesh using the linear elasticity deformation.[12] The control mesh is modelled as a linear elastic solid with a spatially varying Young's modulus, which increases the stiffness of small or skewed cells to maintain mesh quality. The elasticity model produces a linear algebraic system of equations which can be solved to obtain the interior B-spline control point displacements. When the control mesh is undergoing significant deformation, the mesh movement is split into increments, where the intermediate locations of the surface control points are linearly interpolated between their initial and desired location. The user can control the number of increments used and adjust it if large deformations are involved. For a mesh deformation with m increments, each increment gives rise to a set of equations

$$\mathbf{K}^{(i)}(\mathbf{b}^{(i)} - \mathbf{b}^{(i-1)}) - \mathbf{f}^{(i)} = 0 \quad \text{for } i = 1....m, \tag{2.3}$$

where $\mathbf{K}^{(i)}$ is the symmetric positive-definite stiffness matrix which must be computed at each increment, $\mathbf{b}^{(i)}$ is the vector of control point locations at each increment, and $\mathbf{f}^{(i)}$ is the discrete forcing vector defined by the displacements at the surface and the far field. The mesh movement equations must be accounted for in the computation of the objective function gradient, as will be discussed in Section 2.3. For simplicity, they are written as

$$\boldsymbol{\mathcal{M}}^{(i)}(\mathbf{b}^{(i)}, \mathbf{b}^{(i-1)}) = 0 \quad \text{for } i = 1....m. \tag{2.4}$$

2.2 Flow Solution

This section will briefly summarize the flow solution discretization and strategy used by Jetstream. The solver was developed by Hicken and Zingg[12] for the Euler equations and was extended to include viscosity and turbulence by Osusky and Zingg.[28]

2.2.1 Governing Equations

This thesis is focused on the application of the Reynolds-averaged Navier-Stokes equations, which govern the conservation of mass, momentum and energy in turbulent, viscous flows. The equations are given in

nondimensional form by

$$\frac{\partial}{\partial t}\mathbf{Q} + \frac{\partial}{\partial x}\mathbf{E} + \frac{\partial}{\partial y}\mathbf{F} + \frac{\partial}{\partial z}\mathbf{G} = \frac{1}{Re}\left(\frac{\partial}{\partial x}\mathbf{E}_v + \frac{\partial}{\partial y}\mathbf{F}_v + \frac{\partial}{\partial z}\mathbf{G}_v\right), \tag{2.5}$$

where

$$\mathbf{Q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix} \qquad \mathbf{E} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (e+p)u \end{bmatrix} \qquad \mathbf{F} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ (e+p)v \end{bmatrix} \qquad \mathbf{G} = \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ (e+p)w \end{bmatrix}$$

$$\mathbf{E}_v = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ E_{v,5} \end{bmatrix} \qquad \mathbf{F}_v = \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ F_{v,5} \end{bmatrix} \qquad \mathbf{G}_v = \begin{bmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ G_{v,5} \end{bmatrix} \qquad Re = \frac{\rho_\infty a_\infty l}{\mu_\infty}$$

$[u, v, w]^T$ is the velocity in Cartesian coordinates, e is the total energy per unit volume, and τ is the Newtonian stress tensor. The Reynolds number Re is a function of the freestream density ρ_∞, the freestream sound speed a_∞, freestream viscosity μ_∞, and the chord length l. The pressure p is related to the flow variables using the perfect gas equation

$$p = (\gamma - 1)(e - \frac{1}{2}\rho(u^2 + v^2 + w^2)), \tag{2.6}$$

where the ratio specific heats is $\gamma = 1.4$. Turbulence is modelled using the Spalart-Allmaras one-equation turbulence model, given by

$$\frac{\partial \tilde{\nu}}{\partial t} = M(\tilde{\nu})\tilde{\nu} + P(\tilde{\nu})\tilde{\nu} - D(\tilde{\nu})\tilde{\nu} + T, \tag{2.7}$$

where $M(\tilde{\nu})\tilde{\nu}$ is advection and diffusion, $P(\tilde{\nu})\tilde{\nu}$ is the production source term, $D(\tilde{\nu})\tilde{\nu}$ is the wall destruction source term, and T is the transition trip function, which is not used when the flow is fully turbulent, as is assumed in all cases here. The no-slip boundary condition is enforced on all aerodynamic surfaces, meaning $u, v, w = 0$. The farfield solution is dictated by the specified freestream conditions. The initial conditions of the flow are set to the freestream values.

2.2.2 Solution Discretization and Strategy

The flow is solved on multi-block structured meshes. The governing equations are mapped from physical coordinates (x, y, z) to a computational coordinate system (ξ, η, ζ) where the grid is uniformly spaced. The summation-by-parts (SBP) operators provide centered finite difference approximations to both the first and second derivative terms in the Navier-Stokes equations. They have the property that the energy method can be used to prove stability for the linearized Navier-Stokes equations. The computational domain is split into multiple domains, and simultaneous approximation terms (SATs) are used to enforce both the boundary conditions and solution continuity at block interfaces. In the SAT methodology, penalty terms are computed at block interface nodes based on the differences between the flow quantities computed on the adjacent blocks. This approach reduces the amount of communication required between processors, allowing for a highly parallel and scalable algorithm. Osusky and Zingg[28] have shown this discretization to scale well on up to 6000 processors. The SATs also allow for the solver to handle a wide variety of grid topologies and geometries as mesh lines do not need to be continuous across block interfaces.

The Reynolds-averaged Navier-Stokes equations are solved to steady state. Neglecting the time derivative term, the nonlinear discretized system of equations can be written as

$$\boldsymbol{\mathcal{R}}(\mathbf{q}) = 0. \tag{2.8}$$

This system of equations is solved iteratively using a two-phase Newton-Krylov Algorithm. The first stage is an approximate-Newton phase which employs pseudo-transient continuation. The approximate-Newton method is similar to the implicit Euler time marching scheme with local time linearization and uses a spatially varying time step with an approximate Jacobian matrix. Once the flow residual has been reduced by several orders of magnitude, the solution is determined to be a suitable iterate for the second stage of the solver. In the second phase, an inexact form of Newton's method is used to converge the flow rapidly to the desired solution tolerance using a second-order Jacobian. The flexible generalized minimal residual method (FGMRES) with approximate-Schur preconditoning is used to solve the linear system arising from each iteration of the approximate and inexact Newton phases. The solution algorithm terminates once the flow residual has been reduced to a desired relative tolerance. In the drag minimization studies presented in this thesis, the relative solution tolerance is 10^{-8}.

2.3 Gradient-Based Optimization

A gradient-based optimization algorithm is used to minimize the objective function, which in this thesis can be assumed to be drag. Optimization problems are posed as minimizing an objective function $\mathcal{J}(\mathbf{v})$ with respect to design variables $\mathbf{v}$. Due to the presence of intermediate variables in aerodynamic shape optimization, the task of calculating the gradient $\frac{d\mathcal{J}}{d\mathbf{v}}$ is quite involved.

2.3.1 Gradient Evaluation

The objective function depends on the design variables $\mathbf{v}$, the B-spline volume control points at the final mesh movement increment $\mathbf{b}^{(m)}$, and the flow variables $\mathbf{q}$. The design variables $\mathbf{v}$ are either the surface control points, or the axial curve control points and the FFD lattice points. The optimization is subject to the constraint that each mesh movement increment equation is satisfied, thus introducing a dependency on the control points at each mesh movement increment $\mathbf{b}^{(i)}$. It is also subject to the constraint that the flow residual equation is satisfied. Note that terms are now given as functions of the design variables and the intermediate variables which determine the grid geometry.

$$\text{minimize} \quad \mathcal{J}(\mathbf{v}, \mathbf{q}, \mathbf{b}^{(m)})$$

$$\text{w.r.t.} \quad \mathbf{v}$$

$$\text{s.t.} \quad \boldsymbol{\mathcal{M}}^{(i)}(\mathbf{v}, \mathbf{b}^{(i)}, \mathbf{b}^{(i-1)}) = 0. \quad \text{for } i = 1....m$$

$$\boldsymbol{\mathcal{R}}(\mathbf{v}, \mathbf{b}^{(m)}, \mathbf{q}) = 0.$$

For this optimization problem constrained with intermediate variables, the Lagrangian can be introduced with the form

$$\mathcal{L} = \mathcal{J} + \sum_{i=1}^{m} \left({\boldsymbol{\lambda}^{(i)}}^T \boldsymbol{\mathcal{M}}^{(i)} \right) + \boldsymbol{\psi}^T \boldsymbol{\mathcal{R}}, \tag{2.9}$$

where $\boldsymbol{\lambda^{(1)}}...\boldsymbol{\lambda^{(m)}}, \boldsymbol{\psi}$ are the Lagrange multipliers. The Karush-Kuhn-Tucker conditions for optimality state that the partial derivatives of $\mathcal{L}$ must be equal to zero. Setting the condition $\partial\mathcal{L}/\partial\mathbf{q} = 0$ gives the flow adjoint system:

$$\left(\frac{\partial \boldsymbol{\mathcal{R}}}{\partial \mathbf{q}} \right)^T \boldsymbol{\psi} = - \left(\frac{\partial \boldsymbol{\mathcal{J}}}{\partial \mathbf{q}} \right)^T, \tag{2.10}$$

where the term on the left-hand side is the Jacobian matrix of the discrete flow equations. Osusky and Zingg[27] extended the formulation developed by Hicken and Zingg[12] to include the viscous and turbulence

model terms into the Jacobian in the flow adjoint calculation. Setting the condition $\partial\mathcal{L}/\partial\mathbf{b}^{(i)} = 0$ yields the mesh adjoint system:

$$\left(\frac{\partial\boldsymbol{\mathcal{M}}^{(m)}}{\partial\mathbf{b}^{(m)}}\right)^T\boldsymbol{\lambda}^{(m)} = -\left(\frac{\partial\boldsymbol{\mathcal{J}}}{\partial\mathbf{b}^{(m)}}\right)^T - \left(\frac{\partial\boldsymbol{\mathcal{R}}}{\partial\mathbf{b}^{(m)}}\right)^T\boldsymbol{\psi} \tag{2.11}$$

for the last mesh movement increment, and

$$\left(\frac{\partial\boldsymbol{\mathcal{M}}^{(i)}}{\partial\mathbf{b}^{(i)}}\right)^T\boldsymbol{\lambda}^{(i)} = -\left(\frac{\partial\boldsymbol{\mathcal{M}}^{(i+1)}}{\partial\mathbf{b}^{(i)}}\right)^T\boldsymbol{\lambda}^{(i+1)} \quad \text{for } i = m-1...1 \tag{2.12}$$

for the preceding increments. In order to solve for the adjoint variables $\boldsymbol{\lambda^{(i)}}$ and $\boldsymbol{\psi}$, Equation 2.10 is solved first, followed by 2.11, and finally 2.12. The flow adjoint equations are solved using a flexible variant of the generalized conjugate residual with orthogonalization and truncation (GCROT).[14] In the drag minimization studies presented in this thesis, the flow adjoint system is solved to a relative tolerance of 10^{-8}. Since the mesh movement equations are governed by linear elasticity as in Equation 2.3, the right hand side matrices in the mesh adjoint system are symmetric positive definite and can be solved with a preconditioned conjugate gradient (PCG) method. Since the constraints are satisfied at optimality, we have $\mathcal{L} = \mathcal{J}$, and the final optimality condition $\partial\mathcal{L}/\partial\mathbf{v} = 0$ gives the equation for the desired gradient:

$$\frac{d\mathcal{J}}{d\mathbf{v}} = \frac{\partial\mathcal{J}}{\partial\mathbf{v}} + \sum_{i=1}^{m}\boldsymbol{\lambda}^{(i)^T}\frac{\partial\boldsymbol{\mathcal{M}}^{(i)}}{\partial\mathbf{v}} + \boldsymbol{\psi}^T\frac{\partial\boldsymbol{\mathcal{R}}}{\partial\mathbf{v}}. \tag{2.13}$$

2.3.2 Optimization Algorithm

The software package SNOPT,[10] which uses the quasi-Newton method and approximates the Hessian using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) update formula, is used to solve the constrained optimization problem. At each design iteration, SNOPT is provided with the objective function $\mathcal{J}$, its gradient with respect to design variables $\mathbf{v}$, nonlinear constraints, and nonlinear constraint gradients. Note that the constraints on the flow residual and mesh movement equations are already solved in the evaluation of the gradient.

Linear constraints can be specified at the beginning of the optimization and are provided as a sparse coefficient matrix. SNOPT restricts the design space to a subspace within the linear constraints, thus guaranteeing that they are satisfied throughout the optimization process. Constraints such as airfoil thickness can be formulated as a minimum difference between two points, and specified as linear inequality constraints. Examples of linear equality constraints include those specifying some sort of shape

symmetry or continuity. Nonlinear constraints are not satisfied exactly, but to a specified tolerance. At every design iteration, each nonlinear constraint is reevaluated and its gradient with respect to the design variables is computed. Examples of nonlinear constraints include aerodynamic forces such as lift and pitching moment, and nonlinear geometric functions such as volume.

SNOPT solves the constrained optimization problem by reducing the Langrangian merit function, which can be thought of as an augmented form of objective function. Convergence is determined by satisfying the first-order optimality or Karush-Kuhn-Tucker (KKT) conditions. The conditions require that the feasibility, a measure of nonlinear constraint violation, and the optimality or objective gradient norm, are below a specified tolerance. In this thesis, the feasibility and optimality tolerances are set at 10^{-6}; however, due to the difficulty of aerodynamic optimization problems, the optimality is rarely ever reduced below this value.

2.4 Optimization Workflow

Once the SNOPT optimization routine is called, it begins performing design iterations where it continually calls a user-defined function, provided by Jetstream. The main steps performed in each user function can be summarized as follows

- Using the updated design variables from SNOPT, compute the new locations of the B-spline control points defining the aerodynamic surface, depending on the type of geometry control approach used;
- perform linear elasticity mesh movement to deform the rest of the B-spline control mesh and generate the new computational mesh;
- solve the flow based on the new computational mesh and angle of attack;
- evaluate the aerodynamic functionals such as lift, drag, and moment;
- compute the objective function gradient by solving the flow and mesh adjoint systems;
- compute the aerodynamic constraint function gradients by solving the flow and mesh adjoint systems;
- evaluate nonlinear geometric constraint functions and their gradients;
- and finally, provide the updated function values and gradients back to SNOPT.

Chapter 3

Improvements in RANS-based Optimization

This chapter describes some of the lessons learned over the course of this thesis that can be applied to reduce errors and improve the convergence of RANS-based aerodynamic optimization with Jetstream. The first section discusses the common errors encountered and the measures that can be taken to address them. The following section outlines strategies for meshing that have shown to be successful in RANS flowsolves.

3.1 Addressing SNOPT Failures

In optimization cases that show stalled convergence histories, SNOPT terminates the routine prematurely due to repeated failure of design iterations. These errors occur within Jetstream's user-defined function, as a result of either a mesh movement failure or a flow solver failure. Once SNOPT encounters a failed iteration, its response is to continue along its search direction but to scale back its step length by a factor of 10. This results in a design change which is only a small perturbation from the most recent successfully evaluated design. While this almost guarantees a valid design iteration, repeated occurrences of this will slow down the optimization significantly. This can become very problematic because when Jetstream fails to converge during a mesh movement or flow solution, it does not provide SNOPT with gradients or any indication that it encountered a problematic area of the design space. Consequently, the optimizer has a tendency to repeatedly fail iterations, which eventually cause SNOPT to terminate early before a significant reduction can be made in the optimality measure or merit function.

3.1.1 Surface Continuity

Mesh movement failures occur either when the flow solver finds areas in the control mesh with negative Jacobians, or when the mesh movement equations fail to converge. The former case occurs most frequently, and can be spotted visually as the B-spline control mesh becomes heavily distorted. In most cases, this can be addressed by increasing the number of mesh movement increments so that geometric changes are subdivided into smaller steps. Doing so improves the ability of Jetstream to make large shape changes in the control mesh without introducing cell distortion. Since the mesh equation solver is relatively fast in comparison to the flow solve and gradient computation, mesh increments can be added without much increase in computational time per design iteration. When the number of blocks in a mesh is over 100, using PETSc to solve the mesh equations in parallel can also reduce the computational time. Optimizations involving only section variables, such as in Section 4.1 typically require a maximum of 5 increments, while optimizations with larger shape changes such as the winglet optimization in Section 4.3 can require up to 20 increments. Adjusting parameters in the PCG mesh equation solver can address the issue of the mesh equations failing to converge. In particular, using an incomplete Cholesky factorization as a preconditioner instead of the default Jacobi preconditioner has shown to be effective, especially when using higher fill levels.

A more problematic case is when the geometry is insufficiently constrained against producing a non-physical geometry, such as in the case with the optimization of the NASA Common Research Model wing in Osusky.[26] The CRM wing is different from most of the optimization cases studied in previous works with Jetstream due to the fact that it includes a blunt trailing edge. While this is a small geometric feature, it means that the airfoil requires a more complex grid topology than a simple top and bottom surface. The blunt trailing edge requires an additional grid block to be created at the trailing edge, which in turn requires another block to be created around the leading edge for a structured mesh to be viable. As a result, the CRM wing requires at least four patches around an airfoil, leading to the O-O topology, shown later on in Figure 3.4b. This wing geometry can be controlled either through a surface-based or the volume-based geometry control method. When using surface-based geometry control with this topology, problems can arise with mesh movement if one is not careful with constraining the design variables. Since the surface control points are being directly manipulated, it is easy to introduce discontinuities or surface intersections.

Figure 3.1 shows one of the failed iterations from the CRM wing optimization. The optimizer is attempting to produce a sharp drooped leading edge; however it goes too far and creates an infeasible geometry with the bottom surface crossing over the top. Figure 3.2 shows another defect at the leading

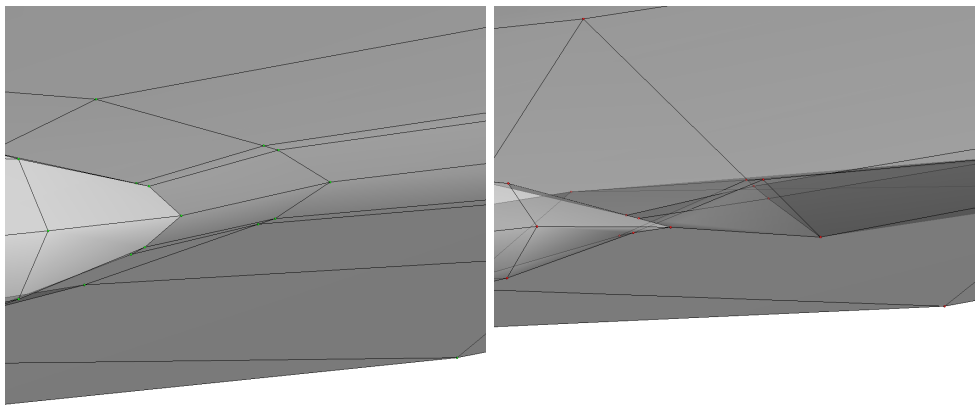


Figure 3.1: Surface crossover in the CRM wing optimization at the leading edge near wingtip

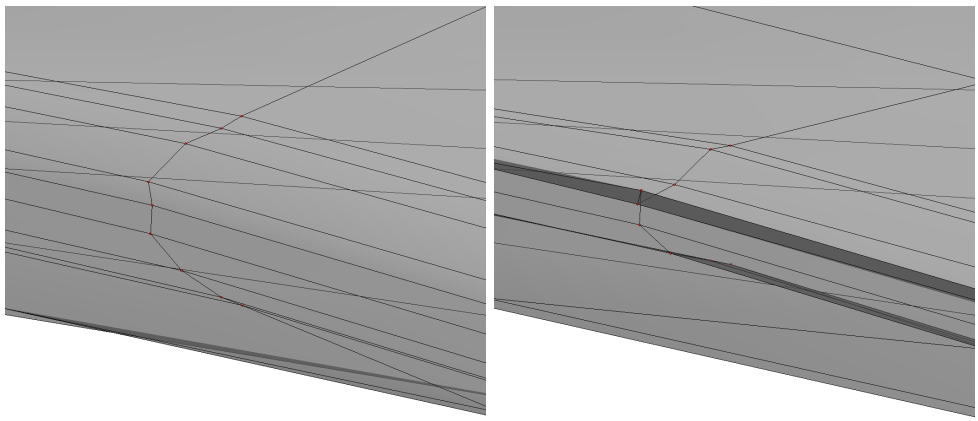


Figure 3.2: Surface continuity error in the CRM wing optimization at the leading edge

edge where control points have crossed over. While it is possible for SNOPT to recover from a design iteration with infeasible geometry, failing to constrain against this case will cause the optimization to stall.

Ensuring Geometric Feasibility Through Linear Constraints

The problems described can be prevented through the specification of constraints to SNOPT. Using linear constraints has the advantage that SNOPT satisifies these constraints exactly and does not require gradients to be supplied at every iteration. Simple geometric relationships such as thickness and curvature can be formulated as a linear equation between surface control point coordinates and supplied this way. For issues such as in Figure 3.1 on the leading edge surface patch, a constraint is added to ensure that the distance between a control point on the upper side and its corresponding point on the lower side must be greater than 25% of the distance in the initial geometry.

The issue shown in Figure 3.2 is addressed by imposing the additional constraint that adjacent points

on the leading-edge surface patch must remain convex throughout the optimization. This is enforced by restricting the z-coordinates of the control points at the leading edge from moving too far above or below the z-coordinates of adjacent control points. In addition to addressing these problems, additional constraints must be added to ensure that the B-spline surfaces remain smooth and continuous across patches. This is done by imposing a linear constraint such that three control points at the interface between two surface patches are always collinear in the x-z plane.

Ensuring Geometric Feasibility Through Volumetric Deformation

Implementing linear constraints requires careful programming by the user depending on the optimization case, since SNOPT requires the indices and coefficients of each constraint equation to be passed as an array into its subroutine call. More importantly, it requires the designer to have detailed knowledge of the surface topology and anticipate any problems that may arise. While surface-based geometry control provides finer and more local control of the airfoil, the need to program different linear constraints for different grid topologies is its main drawback.

Geometry control methods based on volumetric deformation, such as those described in Section 2.1.3, can address some of these difficulties as they avoid directly manipulating the surface. Instead of working with the surface control points as variables, the surface is embedded into an FFD control volume, where the design variables then become the outer FFD lattice. Once embedded, it no longer matters to the user what the topology of the surface contained in the FFD volume is. A wing surface that requires 8 surface patches can be controlled in the same manner as a wing with 32 patches and a blunt trailing edge. While FFD optimizations are not immune from surface crossover and infeasible geometries, a single set of constraints can be generalized to any surface topology embedded within. Furthermore, as Gagnon and Zingg[9] found in their study, FFD based optimizations do not require explicitly constraining the B-spline continuity at surface patch interfaces during optimization.

This method of geometry control is useful when working with more complex configuations such as the winglet and wingtip fence geometry, to be discussed in Section 4.3. In Osusky's[26] study of a wing with vertical tip geometry, she experienced difficulty creating the junction between the horizontal and vertical components of the wing, and as a result had to apply significant constraints. Since her design variables were specified as the y and z coordinates of the surface control points, the optimizer had a tendency to produce misshapen or warped geometries which led to flow solver failures. Osusky's optimized winglet on the left in Figure 3.3 illustrates this point. With axially-driven FFD volumes, the designer can ensure smoothness in the junction between the winglet and the wing by specifying the order of the axial curve that drives the wing shape. In addition, the FFD design sections are always on an orthogonal plane to

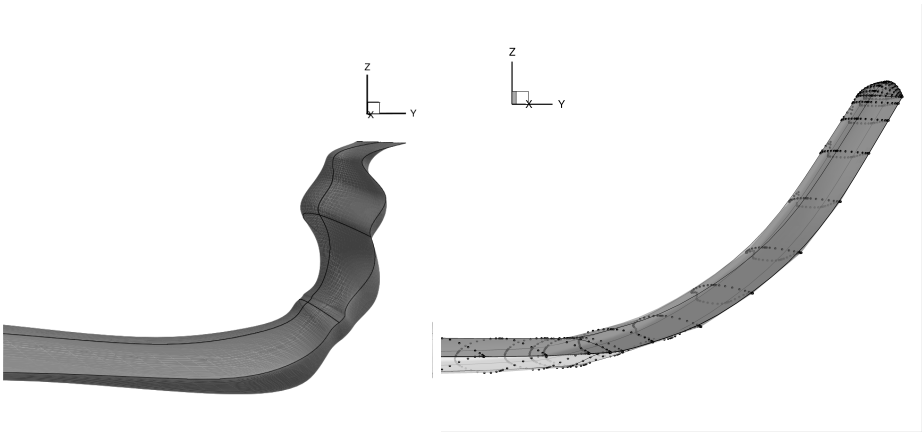


Figure 3.3: Comparison of winglet geometry parameterizations with surface-based control of B-spline points(left) from Osusky and volume-based control(right)

the axial curve, which gives the optimizer better control over the cross-sectional shape at the winglet. The blended winglet on the right in Figure 3.3, created through FFD, is much more realistic and feasible than on the left.

3.1.2 Flow Solver Tuning

The second main cause of failed design iterations in Jetstream is flow solver errors. While some instances of failed flow solutions can be attributed to non-physical geometry due to insufficient constraints, Jetstream can still fail occasionally in solving the flow around valid geometries created by the optimizer. Failed flow solutions during RANS typically occur in either one of two ways - divergence or stall. Flow solver divergence most often occurs towards the end of the approximate-Newton startup phase and is usually a result of strong shocks, which can induce unsteadiness in the flow. For RANS solutions, divergence can also occur for relatively shock-free solutions as it is particularly sensitive to the time-stepping during the continuation phase. The approximate-Newton phase utilizes a spatially varying and increasing time step to accelerate convergence to steady state. The main driving parameter in this phase is the reference time step, which at the nth step is calculated as

$$\Delta t_{\text{ref}}^{(n)} = a(b)^n$$

where a controls the initial value at the beginning of the flow solve and the value of b controls the rate at which it accelerates. In his work on the Euler equations, Hicken and Zingg[11] ran solutions with the values $a = 0.1$ and $1.4 \leq b \leq 1.7$ for fast convergence. Later work by Osusky and Zingg[28] indicated that more conservative values were required for RANS solutions to prevent the flow solver from creating

non-physical solution updates and crashing.

In the RANS optimizations of the CRM geometry, Osusky and Zingg[27] used values of $a = 0.001$ and $b = 1.3$. While this was adequate for the flow solutions during the first set of iterations, occasional flow solver divergence was encountered later on. Over the course of optimization however, Jetstream is likely to encounter flows much different from the initial one due to changes in geometry and angle of attack.

To address this, the RANS optimizations should be run with $1.1 \leq b \leq 1.2$ to improve the likelihood of flow solver convergence at all stages in the optimization. In addition, lowering the initial time step variable to $a = 0.0001$ can also help to mitigate divergence, particularly when optimizing at higher Reynolds numbers. Surprisingly, scaling back on the time step parameters does not incur a significant penalty on the overall solution time, mainly because the approximate-Newton phase iterations are very fast. In addition, starting with a smaller time step has the effect of slightly reducing the stiffness that can occur in the later stages of the approximate-Newton phase due to excessively large time steps.

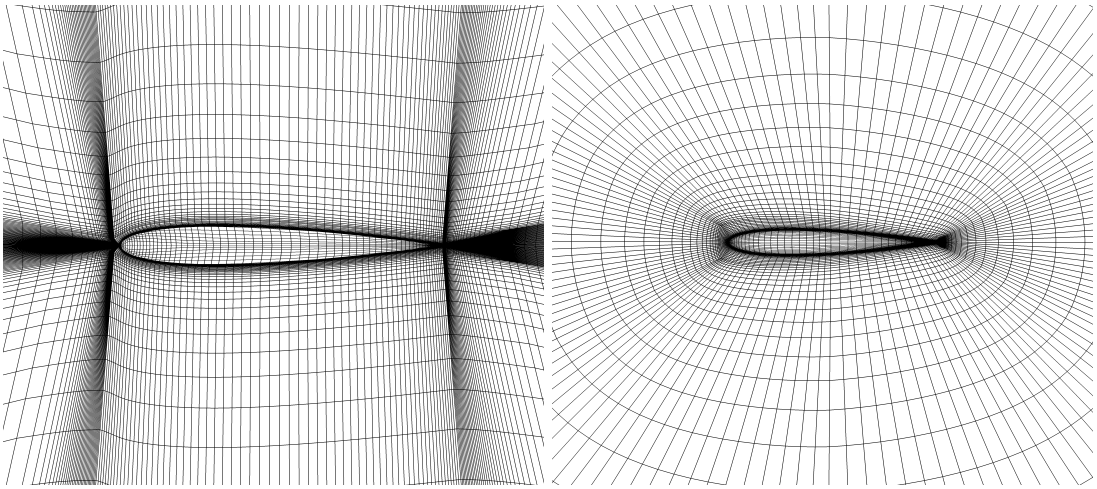
Several other parameters can be used to address residual stall that can sometimes occur in the inexact-Newton phase of the flow solver. While the inexact phase theoretically converges quickly to steady state, this may not occur if the initial iterate is not sufficiently close to the final solution. For RANS cases, Osusky and Zingg[28] suggested the solver should switch into the inexact phase once the flow residual was reduced by 10^{-4}. Since the flow conditions during the optimization can change significantly, it was found that lowering the residual drop tolerance to $5{\times}10^{-5}$ or even 10^{-5} reduced the number of occurrences of residual stalling. As the inexact phase utilizes the time step obtained from the startup phase, lowering the parameters a and b can improve outcomes in the inexact phase by preventing the solver from reaching excessively large time steps. Specifying a maximum time step parameter Δt_{max} at some reasonable value such as 10^3 can help in this regard as well.

3.2 Meshing Strategies

This section will discuss the alternative meshing strategies for producing RANS grids used in this work. While most of the optimization work using Jetstream by past researchers has used meshes with the H-H topology[3, 9, 12] , the optimization cases in this thesis will make use of an O-O topology grid. Figure 3.4 shows the two topologies side-by-side on a NACA0012 airfoil.

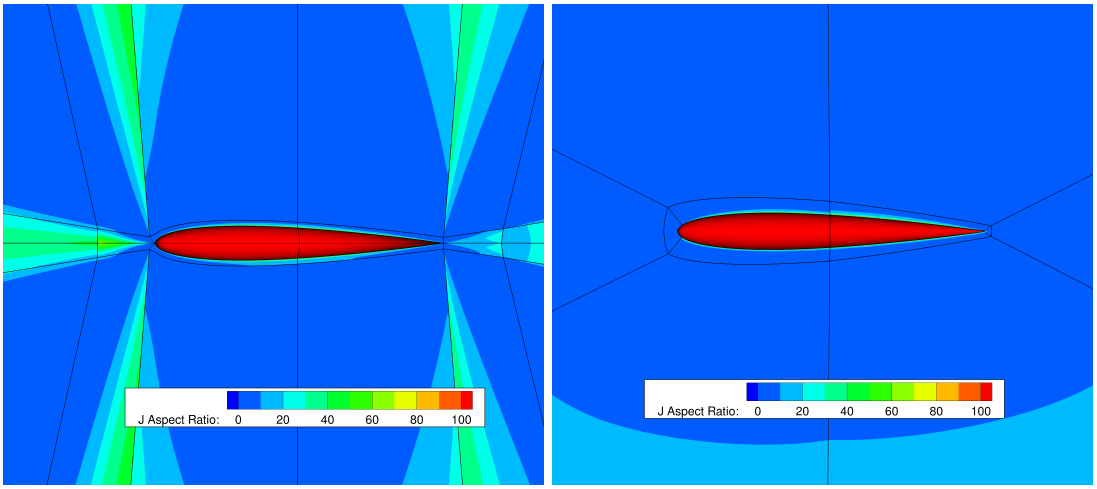
3.2.1 Improved Grid Quality

The main disadvantage of using H-H grids for computing turbulent flows around wings is that the small off-wall spacings required for the Spalart-Allmaras model ($y^+ = 1$) lead to low grid quality and higher



(a) H-grid (b) O-grid

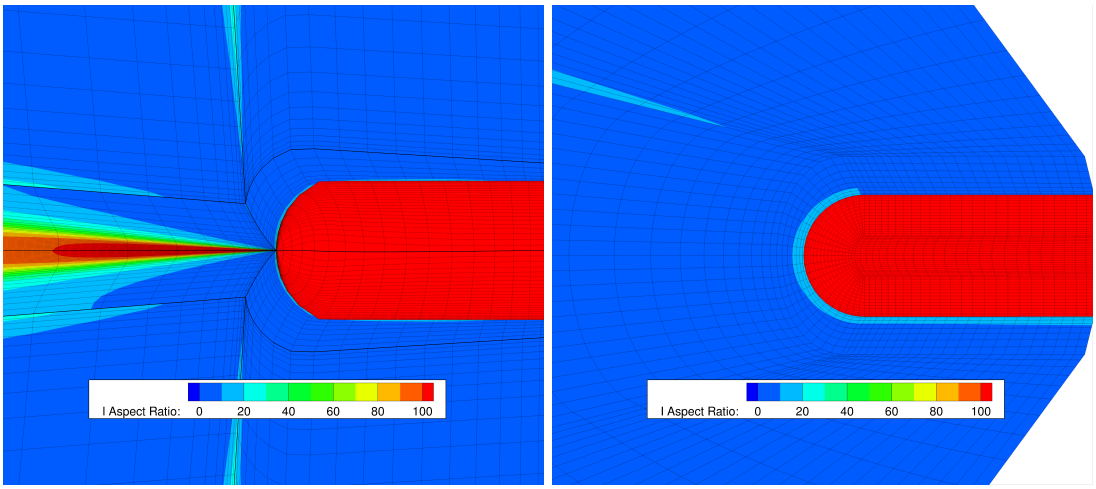
Figure 3.4: Comparison of two different mesh topologies around a NACA0012 airfoil



(a) H-grid (b) O-grid

Figure 3.5: Aspect ratio contours of different mesh topologies around a NACA0012 airfoil

aspect ratios around the trailing edge and wingtips. In Figure 3.4a, the tightly spaced mesh lines around the leading edge and trailing edge have to fan out immediately, creating skewed cells. The O-O grid in Figure 3.4b provides a noticeable advantage in this respect since the grid wraps around the airfoil, meaning that the off-wall grid lines are orthogonal to the wing surface even at the leading and trailing edges. This eliminates the need to have tightly spaced grid lines in two directions, as can be seen in Figure 3.4a. Figure 3.5 shows that the cells surrounding the leading and trailing edge have lower aspect ratios in the O-O grid than the H-H grid. The drawback is that the O-O topology requires the cross section to be a four-sided, meaning it can only be made around airfoils with a blunt trailing edge. The NACA0012 mesh in Figure 3.4b has this modification made.



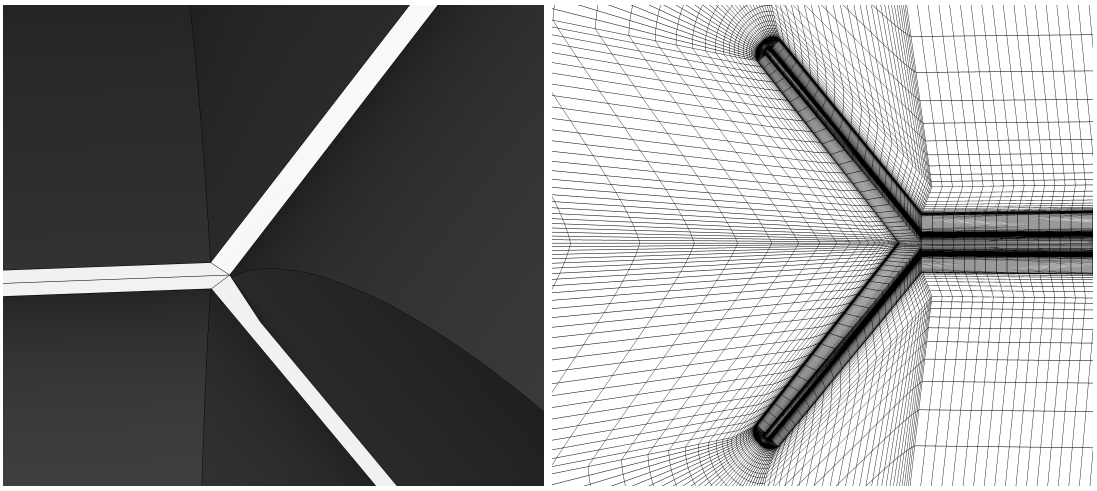
(a) H-grid (b) O-grid

Figure 3.6: Aspect ratio contours at the wingtip of the NACA00012 wing

The other advantage of the O-O grid geometry is at the wingtips of a three dimensional wing. The H-H topology requires that the wing be 'pinched' at the leading edge of the wingtip, meaning that the block faces comprising the upper and lower wing surfaces will converge at a block corner. As with the trailing edge, the tightly spaced grid lines on the surface have to converge at the wingtip and then immediately fan out. The O-O grid however, allows the mesh lines to wrap around the wingtip. Figure 3.6 shows the differences in the meshes at the wingtips with aspect ratio contours. The mesh immediately outboard of the H-H grid wingtip has higher aspect ratios than the corresponding meshes in the O-O grid, which can lead to stiffness in the flow solver.

3.2.2 Complex Geometries

One of the additional benefits of the blunt trailing edge and O-O topology grid is that it allows for the creation of RANS grids around complex geometries. The main example of this is the wingtip fence geometry which is later optimized in Section 4.3. The main wing splits into two extensions at the wingtip - one deflected upwards, the other deflected downwards. The blunt trailing edge feature allows for the geometry to be modified in a way that allows for this to occur. Figure 3.7a shows that the blunt trailing edge is split into two grid blocks, which then split off to form trailing edges for each of the two tips. Figure 3.7b illustrates the advantage of this topology: since the outer O-block wraps around the geometry, it allows for the mesh spacing orthogonal to the surface to be very fine without comprimising mesh quality, even at the junction between the two wingtips.



(a) Trailing edge (b) O-grid

Figure 3.7: Detail of the mesh formed around the wingtip fence

Chapter 4

Drag Minimization Studies

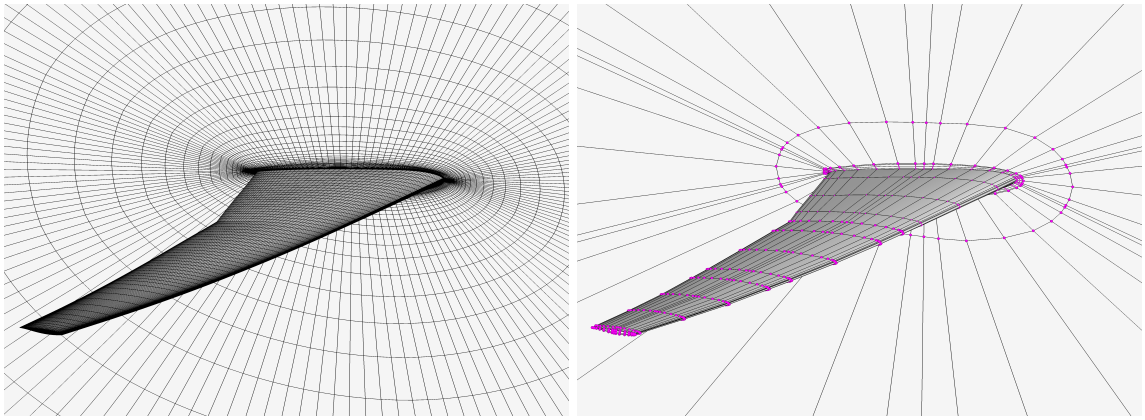
This chapter presents results from four cases in drag minimization. The first case is the optimization of the NASA Common Research Model wing, which employs B-spline surface-based geometry control. This geometry is investigated in further detail, looking at multi-point optimization, the effect of different thickness constraints, and an investigation of multimodality. The subsequent cases are meant to demonstrate Jetstream's robustness in handling unconventional geometries and large shape changes. These include a planform optimization, a wingtip or winglet optimization, and a box-wing optimization, all of which will make use of the FFD volumetric geometry control method.

4.1 NASA Common Research Model Wing

The problem is the drag minimization of the wing geometry extracted from the Common Research Model (CRM) wing-body configuration from the Fifth Drag Prediction Workshop. The single-point and multi-point optimizations of this geometry are benchmark cases for the Aerodynamic Design Optimization Discussion Group (ADODG)[1]. The goal is to optimize the sectional shape and twist to minimize drag at a nominal lift coefficient of $C_L = 0.50$, a Mach number of 0.85, and a Reynolds number of 5 million. The design variables are the z-coordinates of either the B-spline surface control points in addition to the angle of attack. The B-spline points on the trailing edge of the wing are fixed to permit arbitrary twist, except for the root, where the leading edge control point is also fixed. Additional constraints include

- a pitching moment constraint, $C_M \geq -0.17$;
- the wing volume must be greater than or equal to its original volume;
- the wing thickness at every point must be greater than or equal to 25% of the original thickness.

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



(a) CFD Mesh (b) B-spline Surface

Figure 4.1: The computational mesh and B-spline surface for the CRM wing geometry

Initial Geometry & Grid

The wing geometry is scaled by the mean aerodynamic chord of 275.8 inches and translated such that the origin is at the root leading edge. Moments are calculated about the point (1.2077, 0, 0.007669). All aerodynamic force coefficients are calculated using a reference area of $S_{\text{ref}} = 3.407$ squared reference units, which is constant during optimization. The initial volume which must be maintained, V, is 0.2617 cubed reference units. The surface is controlled directly through the coordinates of the B-spline control points. The wing is divided into three spanwise sections, with each section consisting six patches. Including the wingtip patches, the geometry has a total of 20 surface patches. The leading-edge patches consist of 5 points in the streamwise and spanwise directions, while all other patches have 9 points in the streamwise direction and 5 points in the spanwise direction. This gives a total of 15 spanwise design sections, each controlled by 35 points.

In this thesis, a new grid is created in ICEM CFD using the same CRM wing geometry and surface topology as in Osusky.[26] Figure 4.1 shows the surface and symmetry planes for the the computational mesh as well as the optimization B-spline surface. The main difference is that this grid is coarser and requires only 40 blocks while Osusky's used 144. As a result, the grid can be solved with fewer cores, allowing multi-point cases to be run in parallel. Osusky ran a case with five operating points in serial, completing about 22 function evaluations in 120 hours with 162 cores. The nine-point operating case in Section 4.1.2 was run in parallel, completing 110 function evaluations in 90 hours using 360 cores. If Osusky's grid were to be used, completing the same amount of function evaluations for the nine-point case would require around 1080 hours in serial with 162 cores, which would take over a month. In addition, the single-point cases also run faster on the 40 block mesh. The single-point optimization presented here completed over 350 function evaluations in 163 hours, while Osusky's performed 100

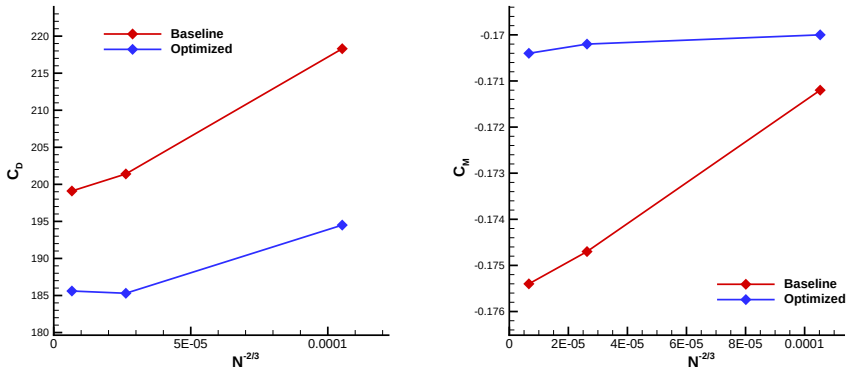


Figure 4.2: Grid convergence for the CRM wing grid study

Table 4.1: Grid parameters for CRM wing grid study

Grid	**Nodes**	**Blocks**	**Off-wall Spacing** (ref. units)	**Average y^+**
Coarse	925,888	40	1.5×10^{-6}	0.33
Fine	7,407,104	320	8.1×10^{-7}	0.17
Superfine	58,456,064	1280	3.9×10^{-7}	0.081

evaluations in 120 hours.

Table 4.1 shows the information on the different grid levels. The grid is refined in all directions by factors of 2 and 4 to give three levels in total. Figure 4.2 shows the results of the grid convergence study on the initial and B-spline surface optimized geometries, with angle of attack adjusted to give $C_L = 0.50$. Note that on the finer meshes, the baseline wing geometry already violates the pitching moment constraint of $C_M \geq -0.17$. The difference between the fine and superfine grid levels is about 2 drag counts for the initial geometry, and less than a count for the optimized geometry.

4.1.1 Single-Point Optimization

To reduce computational time, optimization is performed on the coarsest grid level. Figure 4.3 shows the pressure contours of the baseline and optimized wings; Figure 4.4 shows the corresponding sectional pressure distributions, computed on the fine grid level. When the optimized geometry is analyzed on the finer mesh, there is a small region where oscillations can be seen in the surface pressure distributions, which are not seen on the coarse optimization mesh. The pitching moment constraint is satisfied on the coarse mesh, however Table 4.2 show that this constraint is violated slightly when analyzed on a finer mesh. Both of these minor issues can be addressed by performing the optimization at a finer grid level if resources permit.

The wing sections all become thinner except at the root, which thickens to maintain the initial

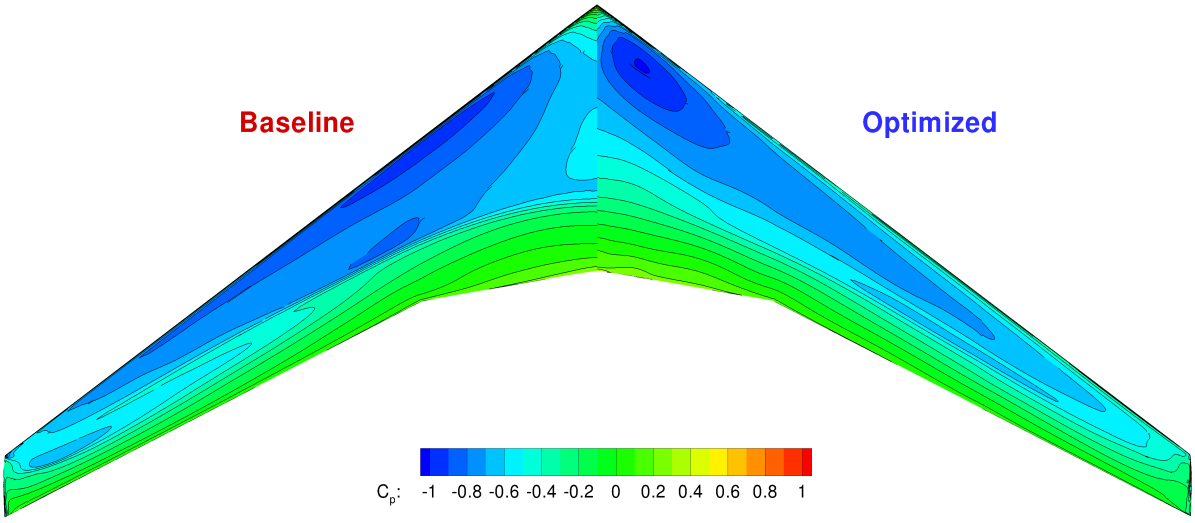


Figure 4.3: Pressure contours for baseline and optimized CRM wing, computed on the fine mesh

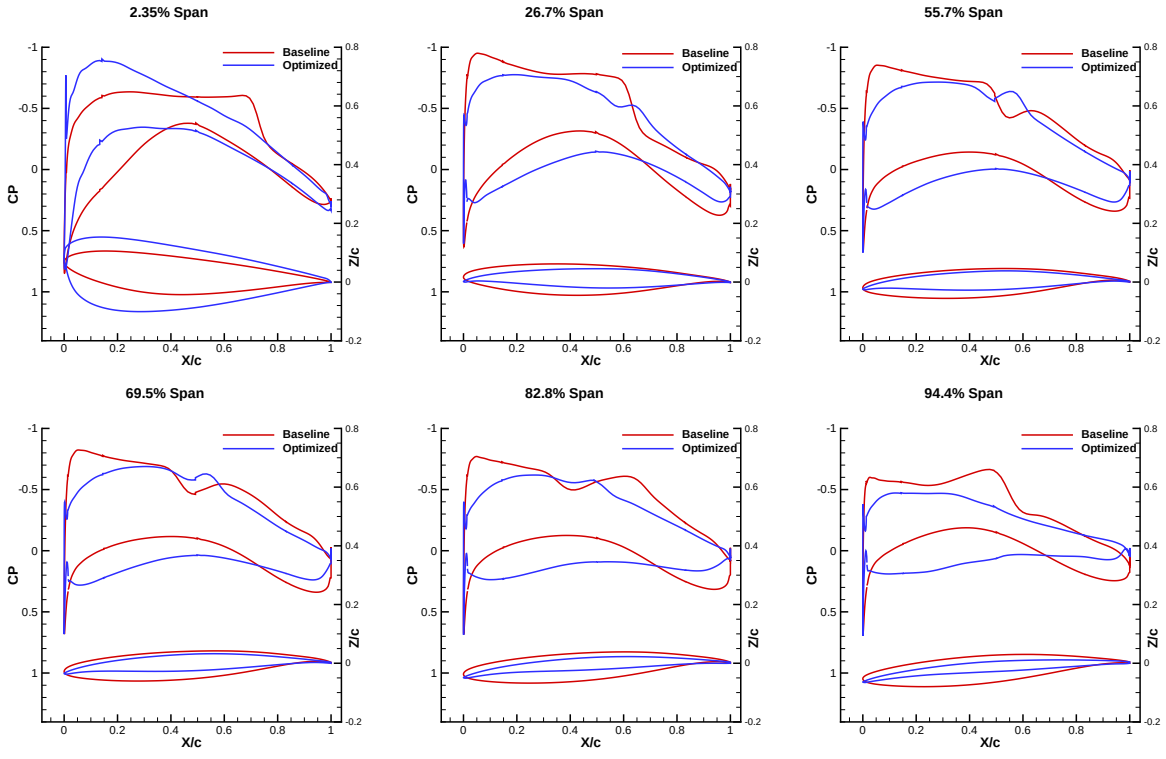


Figure 4.4: Sectional pressure plots and shapes for baseline and optimized CRM wings computed on fine mesh

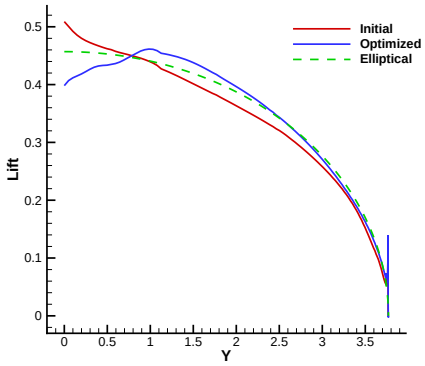


Figure 4.5: Initial, B-spline surface optimized, and elliptical lift distributions

Table 4.2: Results for CRM wing single-point optimization

	Optimization Mesh		Fine Mesh		Superfine Mesh	
	C_D (counts)	C_M	C_D (counts)	C_M	C_D (counts)	C_M
Baseline	218.3	-0.1712	201.5	-0.1747	199.1	-0.1754
Optimized	194.5	-0.1700	185.2	-0.1702	185.6	-0.1704

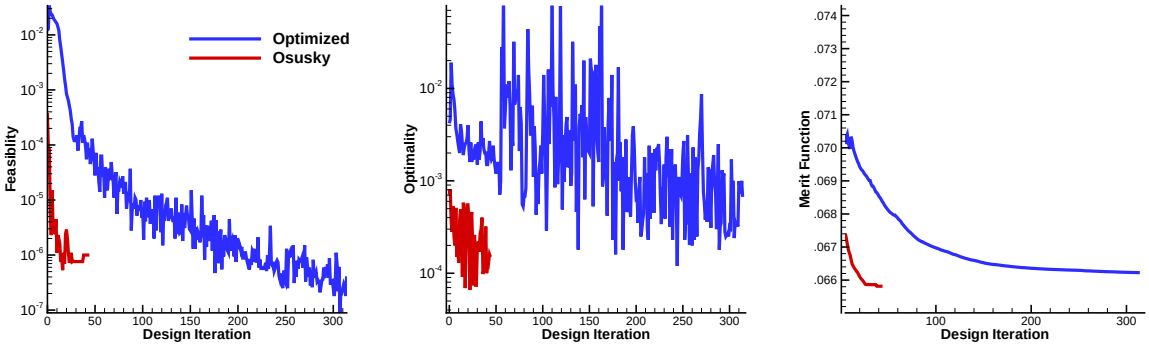


Figure 4.6: Optimization convergence for single-point optimizations of the CRM wing

volume. While the wing has a sharp leading edge, it is not as sharp as in the optimized geometry of Osusky.[26] This is due to the linear constraints that were implemented to maintain a convex leading edge and prevent surface crossover. As a result, this optimization does not encounter mesh movement errors at the leading edge such as in Figure 3.1. The sharp leading edge likely occurs due to the absence of a low-speed lift constraint for the wing. The spanwise lift distributions of the initial and optimized geometries evaluated on the fine mesh are compared to the elliptical distribution in Figure 4.5. With the exception of the root sections, which must remain thick due to the volume constraint, the optimized lift distribution is closer to elliptical than the baseline.

Figure 4.6 shows the SNOPT convergence history for the optimization in this work and from Os-

usky.[26] The feasibility and optimality tolerances are both set at 10^{-6}. While Osusky's optimization stalls after around fifty design iterations, this optimization is able to run to around three hundred iterations and reduce optimality further. The optimality measure is reduced by roughly two orders of magnitude relative to its highest value and the merit function plot shows that most of the drag reduction has already been achieved. There is also a greater reduction in feasibility. The value of the optimality measure in Osusky's work has a lower initial value due to differences in the gradient scaling. The merit function from Osusky's work is plotted, but it only appears to be less than the merit function in this work since the friction drag in her case was calculated using the farfield viscosity value, ν. In this work, the local value of ν is used, which is more accurate but increases friction drag by around three counts.

4.1.2 Multi-Point Optimization

The degrees of freedom and geometry for the CRM wing multi-point optimization are the same as the single-point problem, with the exception that the angle of attack at each design point is a design variable. As with the single-point optimization, surface-based geometry control with the B-spline control points is used. In each design iteration of the multi-point optimization, a flow solution is computed at each of the operating points in parallel. The objective function and gradient are computed as a weighted sum of the results from each of the converged flows. The pitching moment constraint is only satisfied at the nominal design point, which is given the greatest weight $\mathcal{T}_i$. There are four three-point cases: one with variable C_L and constant Mach number, two with variable Mach number and constant C_L, and one with variable Mach number and constant lift. In addition, there is a nine-point case over a range of Mach numbers and lift coefficients. The operating points for each case are summarized in Table 4.3. The optimization problem is posed as

$$\begin{aligned}
\text{minimize } & \sum_{i=1}^{n} \mathcal{T}_i C_{Di} \\
\text{wrt } & z, \alpha_i \\
\text{subject to } & C_L = C_{Li} \\
& C_M \geq -0.17 \text{ (at nominal design point)} \\
& V \geq V_{\text{baseline}} \\
& |z| \geq 0.25 \times |z_{\text{baseline}}|.
\end{aligned}$$

All of the multi-point optimizations are carried out on the coarse mesh, while all the drag polars and analyses are performed with the optimized geometry on the fine level mesh. The optimization histories of cases MP2 and MP6 are compared in Figure 4.7. The three-point optimizations converged more

Table 4.3: CRM wing optimization - Multi-point problem operating points

Case	Point i	Weight $\mathcal{T}_i$	M	C_L	Re
MP2	1	1	0.85	0.450	5.00×10^6
	2	2	0.85	0.500	5.00×10^6
	3	1	0.85	0.550	5.00×10^6
MP3	1	1	0.84	0.500	5.00×10^6
	2	2	0.85	0.500	5.00×10^6
	3	1	0.86	0.500	5.00×10^6
MP4	1	1	0.82	0.500	5.18×10^6
	2	2	0.85	0.500	5.00×10^6
	3	1	0.88	0.500	4.83×10^6
MP5	1	1	0.82	0.537	4.82×10^6
	2	2	0.85	0.500	5.00×10^6
	3	1	0.88	0.466	5.18×10^6
MP6	1	1	0.82	0.483	4.82×10^6
	2	2	0.82	0.537	4.82×10^6
	3	1	0.82	0.591	4.82×10^6
	4	2	0.85	0.450	5.00×10^6
	5	4	0.85	0.550	5.00×10^6
	6	2	0.85	0.550	5.00×10^6
	7	1	0.88	0.442	5.18×10^6
	8	2	0.88	0.466	5.18×10^6
	9	1	0.88	0.513	5.18×10^6

successfully than the nine-point case, as the nine-point case suffered from more flow solver convergence difficulties at the more demanding flow conditions. The drag and moment coefficients of the optimized geometries computed on the fine mesh at the nominal condition are displayed in Table 4.4. The sections and sectional pressure plots computed on the fine mesh at the nominal condition are displayed in Appendix A. Figure 4.8 shows the lift and moment curves for the initial, single-point, and multi-point optimized geometries. The single-point optimization result is also shown for comparison and labeled as case SP. Figure 4.9 shows the drag coefficient vs. angle of attack and the drag polar at Mach 0.85. The drag polar shows that all the optimized geometries produce a noticeable improvement in L/D compared the baseline geometry. Compared to the multi-point optimizations, the single-point result shows poorer performance at lower lift coefficients and a slight improvement at the nominal flight condition $C_L = 0.50$. The degradation in performance at off-nominal points is not as pronounced as in results from other groups, such as Ledoux et al.[20] Cases MP2 and MP3 perform the best over the range of lift coefficients at $M = 0.85$, which is not surprising as these cases have a narrow range of operating points around the nominal condition.

Figure 4.10a better illustrates the advantage of multi-point optimization over single-point. The single-point geometry shows a higher drag over most Mach numbers, with a slight benefit at the nominal

Table 4.4: CRM wing optimization - Drag counts at nominal operating point $C_L = 0.5$ and Mach 0.85 computed on fine mesh

	C_D (counts)	C_M
Baseline	201.5	-0.1746
Single-Point	185.2	-0.1702
Case MP2	185.8	-0.1704
Case MP3	185.8	-0.1705
Case MP4	187.8	-0.1711
Case MP5	187.0	-0.1709
Case MP6	189.7	-0.1717

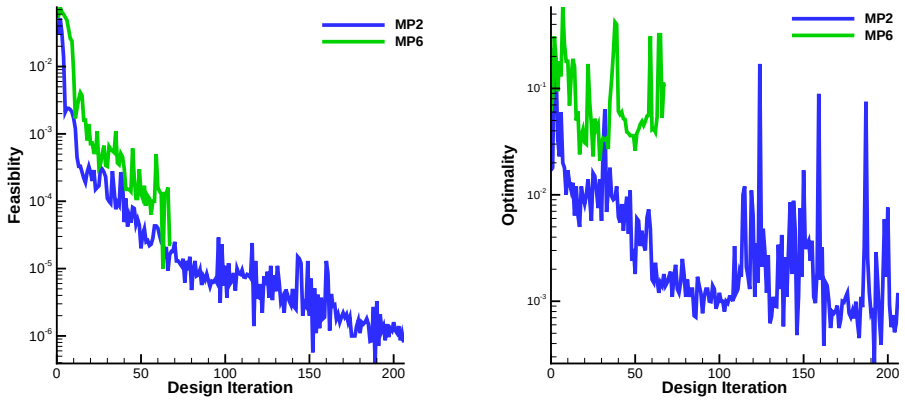
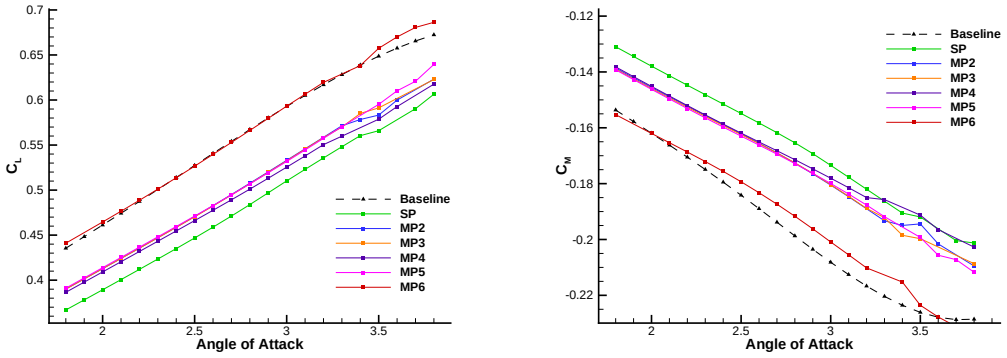


Figure 4.7: Optimization convergence for multi-point optimizations of the CRM wing

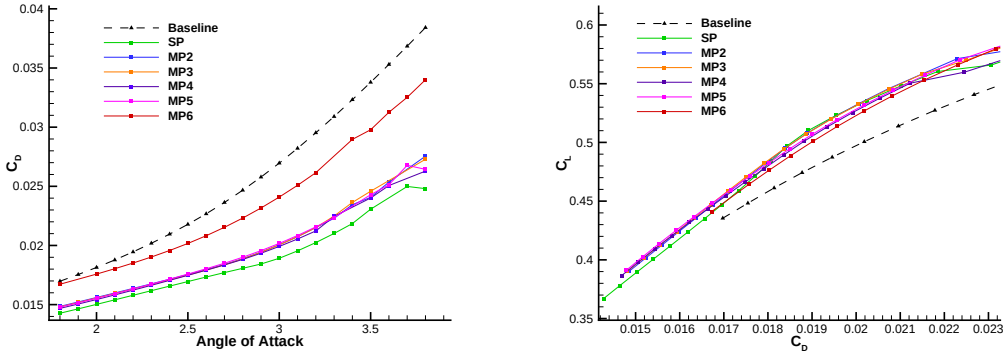
Mach number 0.85. Once again, cases MP2 and MP3 perform the best around the operating condition, with the case MP3 geometry showing slightly better drag. Case MP4, which was optimized at Mach 0.82 and 0.88 in addition to the nominal Mach number, shows significantly better drag at higher Mach numbers. Figure 4.10b shows case MP5 outperforming MP6, which is expected since case MP6 was optimized with consideration of additional operating conditions to those of case MP5.

Figure 4.11 shows the lift and moment curves for the initial and nine-point optimized geometries, at varying Mach numbers. Figure 4.12 shows the drag coefficient vs. angle of attack and the drag polar at varying Mach numbers. The drag polar shows that for a given Mach number, the drag reduction relative to the baseline curve improves at increased C_L. The drag reduction is marginal for Mach 0.82, but much more significant at Mach 0.85 and 0.88. Figure 4.13 plots drag coefficient against Mach number for three fixed lifts given by $C_L = 0.45$, 0.50, and 0.55 at Mach 0.85. Again, the drag reduction relative to the baseline geometry increases at higher Mach numbers. The drag and moment coefficients computed on the fine mesh for optimized geometries at their design conditions are summarized in Appendix A.



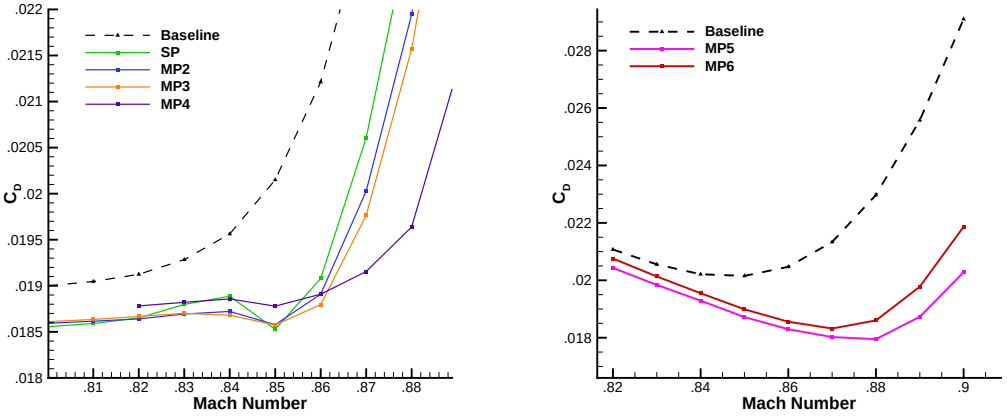
(a) C_L vs. α (b) C_M vs. α

Figure 4.8: CRM wing optimization - C_L and C_M vs. α for single-point, three-point, and nine-point optimizations at $M = 0.85$



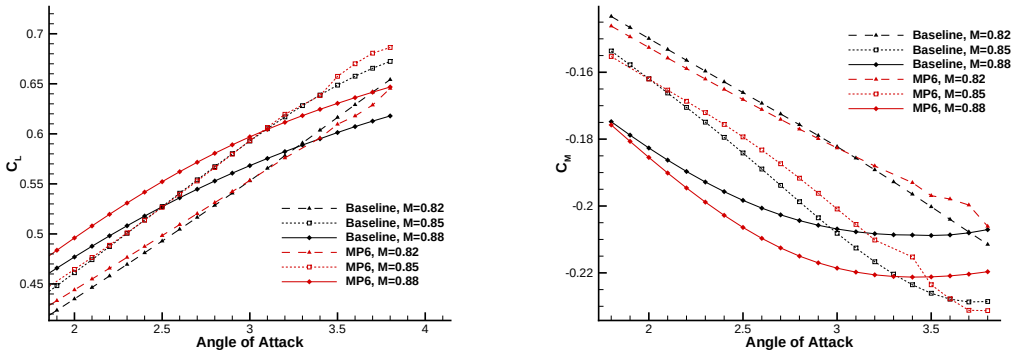
(a) C_D vs. α (b) Drag polar

Figure 4.9: CRM wing optimization - C_D vs. alpha and vs. C_L for single-point, three-point, and nine-point optimizations at $M = 0.85$



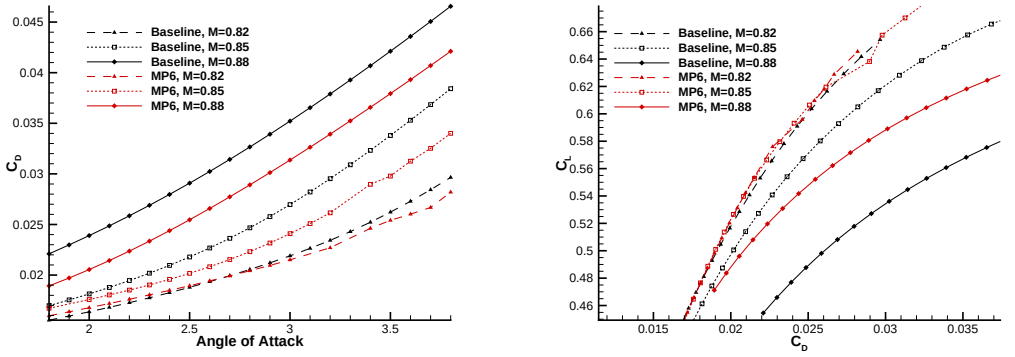
(a) Constant $C_L = 0.5$ (b) Constant lift

Figure 4.10: CRM wing optimization - C_D vs. Mach number for (a) C_L = 0.5 (b) constant lift based on C_L = 0.5 and $M = 0.85$



(a) C_L vs. α (b) C_M vs. α

Figure 4.11: Case MP6 - C_L and C_M vs. α for nine-point optimization at various Mach numbers



(a) C_D vs. α (b) Drag polar

Figure 4.12: Case MP6 - C_D vs. α and vs. C_L for nine-point optimization at various Mach numbers

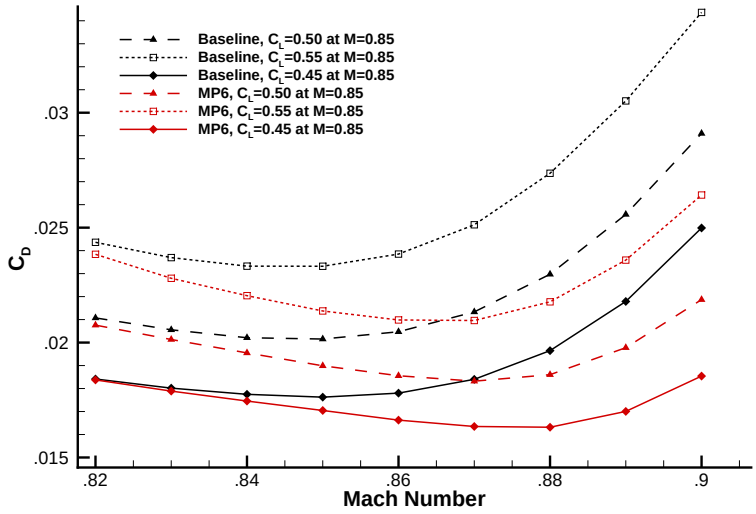


Figure 4.13: Case MP6 - C_D vs. Mach number for nine-point optimization at various lift values

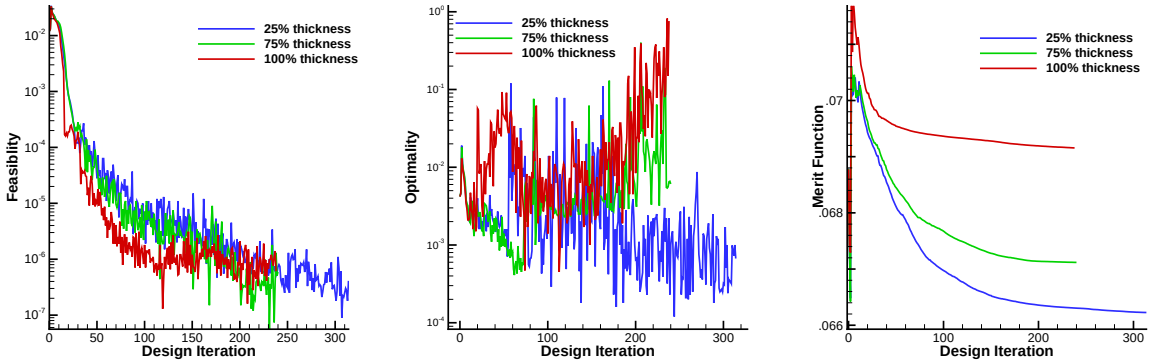


Figure 4.14: Optimization convergence for varying thickness constraints

4.1.3 Optimization with Different Thickness Constraints

In both the single and multi-point optimizations, the optimizer is able to exploit the minimum 25% thickness constraint imposed to produce very thin outboard sections as well as sharp leading edges. To study the impact of thickness constraints on the resulting geometry and drag, the CRM single-point optimization is re-run with both a 75% minimum thickness constraint and a 100% minimum thickness constraint. Figure 4.14 shows the optimization convergence histories for the three optimizations. The optimizations with 75% and 100% minimum thickness were run for about 250 design iterations until they no longer produced any significant reduction in drag. Despite this, the optimality measures for these cases do not decrease over time, which may suggest that optimizing with a greater thickness is a more difficult design problem.

Table 4.5 shows the drag counts of the various optimized geometries on the coarse optimization mesh and fine mesh. Figure 4.15 shows the various optimized geometries and pressure distributions. All of the optimizations are able to reduce the drag and remove the strong shock near the root in the initial geometry, with the main difference in pressure distribution being at the outboard sections. In addition, the thickening of the root section is less pronounced as the thickness constraint increases due to the optimizer's reduced freedom in redistributing the wing volume. Not surprisingly, the optimized geometry with the lowest thickness constraint has the thinnest airfoil sections and the best drag performance. On the fine mesh, there is a difference of only 1.5 drag counts between the 75% optimized and the 25% optimized geometry, meaning that the unusally thin sections in the original optimization only have a small impact on the overall drag. The optimized geometry at 100% thickness has some trouble removing the shocks outboard from 55.7% of the span.

Table 4.5: Results for CRM wing single-point optimization with different thicknesses

	Optimization Mesh		Fine Mesh	
	C_D (counts)	C_M	C_D (counts)	C_M
Baseline	218.3	-0.1712	201.5	-0.1747
25% minimum thickness	194.5	-0.1700	185.2	-0.1702
75% minimum thickness	197.0	-0.1700	186.7	-0.1708
100% minimum thickness	203.0	-0.1700	191.0	-0.1725

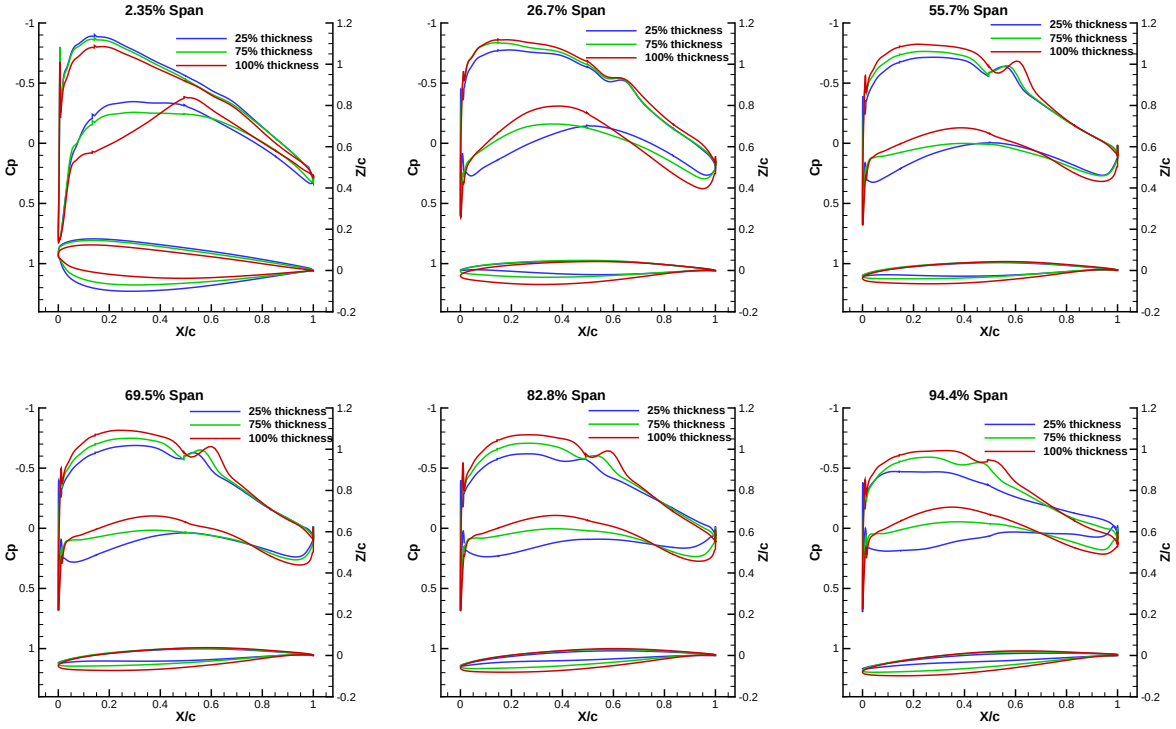


Figure 4.15: Sectional pressure plots and sections for optimized CRM wings with different thickness constraints, computed on fine mesh

4.1.4 Multimodality Investigation

Osusky[26] observed in her work that optimizing the CRM wing geometry with varying initial geometries led to several different 'optimized' geometries that had the same drag performance. Osusky postulated that the optimization problem belonged to a multimodal or flat design space. Due to issues both in the flow solution and mesh movement, the optimizations did not make a noticeable reduction in optimality, suggesting that these solutions were not yet optimal. In light of the improvements made to the convergence of the CRM optimization case, another study in multimodality is performed with varying section shapes. The single-point optimization with a 25% minimum thickness constraint is attempted, and the solutions are only compared on the coarse optimization mesh. The different initial geometries were generated by first creating a control mesh of the original CRM surface, then replacing

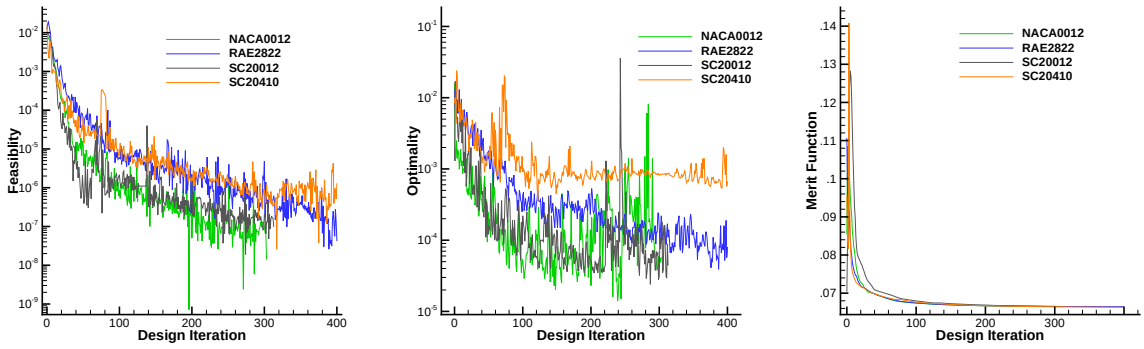


Figure 4.16: Optimization convergence for CRM optimization with different initial geometries

Table 4.6: Results for CRM wing single-point optimization with different initial geometries, with drag coefficient computed on coarse optimization mesh

	C_D (counts)	C_M
Baseline	218.3	-0.1712
Optimized original geometry	194.5	-0.1700
Optimized with NACA0012 sections	195.2	-0.1700
Optimized with RAE2822 sections	195.0	-0.1700
Optimized with SC20012 sections	195.0	-0.1700
Optimized with SC20410 sections	194.5	-0.1700

the B-spline fitting at each spanwise section with the B-spline fitting corresponding to a specified airfoil section, scaled up with the correct chord and twist. The geometric constraints such as volume and thickness are still based on those of the original CRM wing geometry. While a Sobol sampling method is a more rigorous way to thoroughly explore the design space,[3] this often led to geometries that were geometrically unfeasible or could not be solved. Instead, a number of different initial airfoils were used, including the NACA0012, RAE2822, SC20012 and SC20410 sections.

Table 4.6 summarizes the optimized drag coefficients evaluated at $C_L = 0.50$, and Figure 4.17 shows the optimized sections and pressure distributions. All of the optimized geometries are within one drag count of the original single-point optimization and the section shapes are nearly the same. While the initial geometries all have significantly different pressure contours, as shown in Figure 4.18, the optimized surface contours look very similar. Figure 4.16 shows the convergence histories for the different optimizations. All of the optimizations were able to run for over 350 design iterations until the there was no longer any significant change in drag. With the exception of the case beginning with SC20410 sections, all the cases reduced the optimality by around two orders of magnitude, indicating that they are fairly close to the optimum. In Figure 4.17, the SC20410 optimization shows a significant deviation in sectional shape from the rest of the geometries at 55.7% span. Since this is the case that also converged

the least in opimality, it is likely that running this optimization for longer will lead to the same geometry and a greater reduction in optimality. These results suggest that, contrary to what was suggested by Osusky,[26] the CRM optimization does not have multiple local optimum solutions. While the additional linear constraints that were implemented make this a slightly different optimization than that of Osusky, these constraints were necessary to prevent the surface crossover issues at the leading edge that caused issues in her multimodality study. The section shapes in Figure 4.17 show that these constraints were not overly restrictive, since a sharp leading edge is still formed in each case.

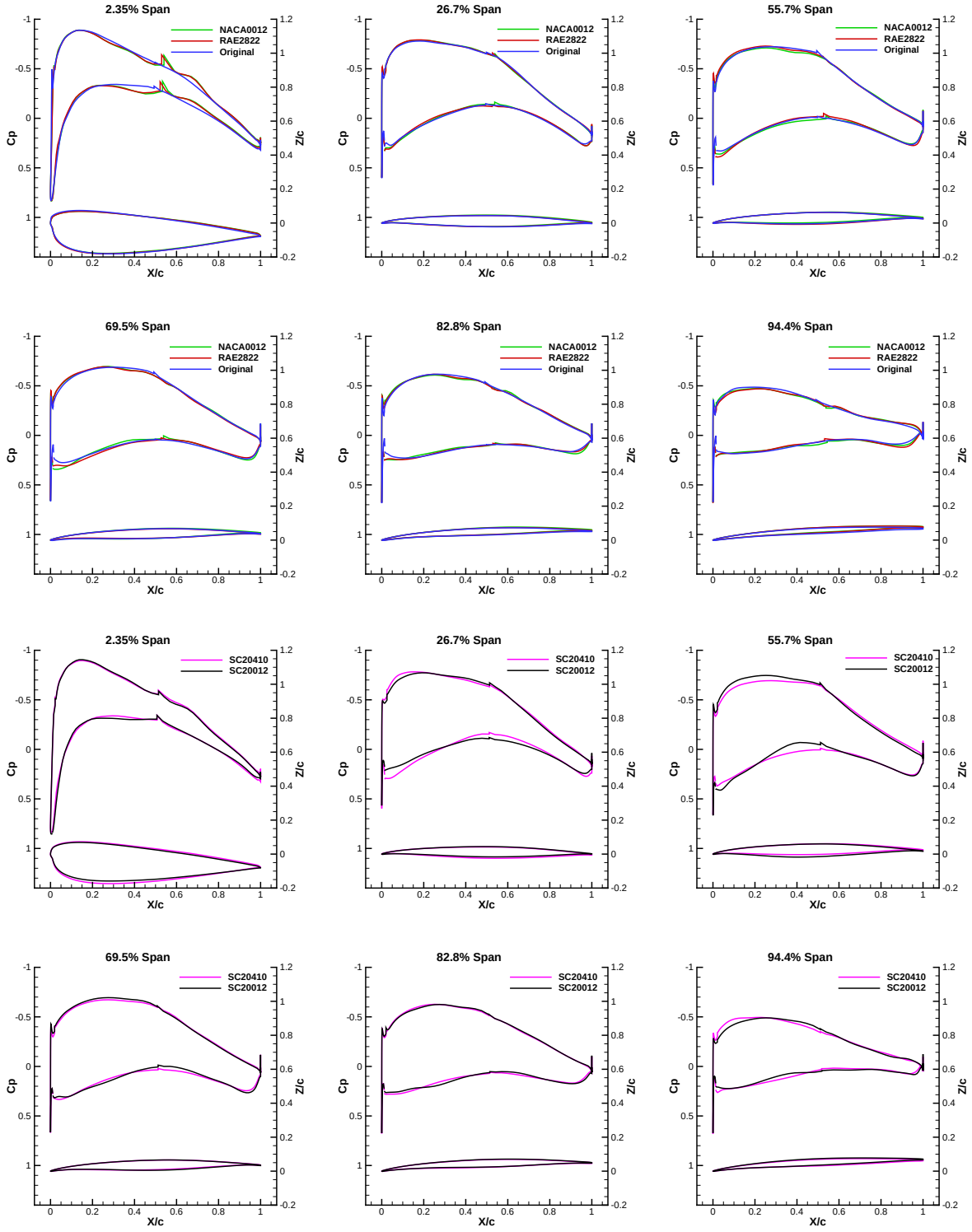


Figure 4.17: Sectional pressure plots and sections for optimized CRM wings with different initial geometries, computed on the optimization mesh

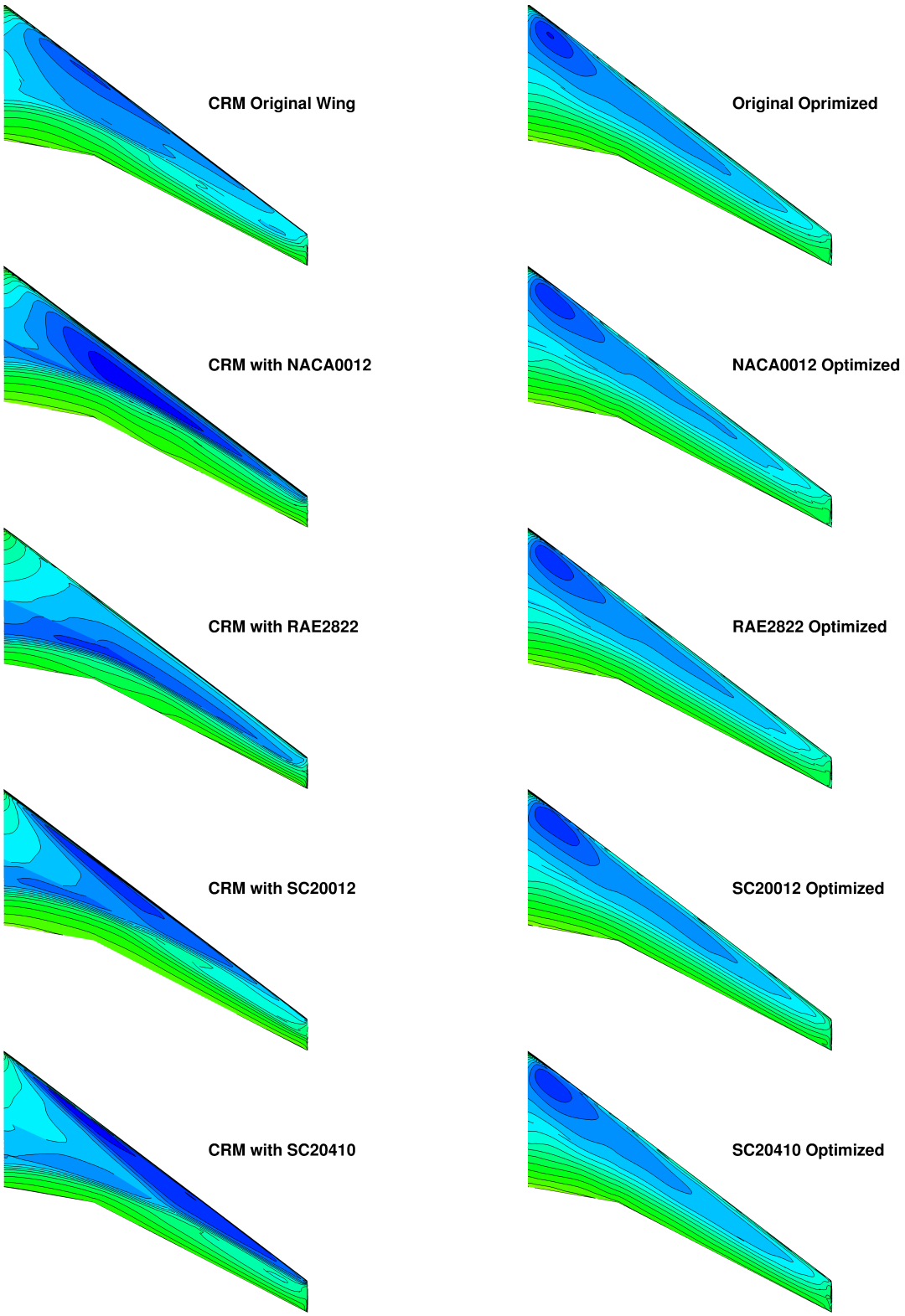


Figure 4.18: Surface pressure contours of starting geometries (left) and optimized (right)

Table 4.7: Planform variable limits

Variable	Units	Upper	Lower
Half-Span	chords	3.95	1.00
Chord	chords	2.0	0.1
Taper	ratio	20.0	0.10
Tip Twist	degrees	4.0	-4.0
Quarter Chord Sweep	degrees	32.0	-32.0
Angle of Attack	degrees	-4.0	4.0

4.2 Exploratory Planform Optimization

4.2.1 Problem Definition

The planform optimization case initially attempted by Osusky[26] is revisited. The goal of this drag minimization study is to demonstrate Jetstream's ability to produce large changes in both wing planform and section shape during optimization. The results demonstrate that Jetstream can be used not only as an aerodynamic shape optimization software, but as a powerful tool for exploratory design.

The initial wing is rectangular with NACA0012 sections. Jetstream is tasked with turning this poorly designed wing into one suitable for transonic flight. The initial mean aerodynamic chord is used as the reference length. The initial span is 3.0 reference units. The design variables enabled are wing sweep, wing span, taper, section shape, twist and angle of attack. The projected area of the wing S is free to change, as long as the optimizer achieves the required lift, which is specified as a non-dimensionalized lift area $C_L S$. The limits on these planform variables, shown in Table 4.7, are chosen so that they allow for significant planform change but are somewhat realistic. In addition, the following constraints are imposed:

- The wing maintains at least 85% of its original thickness at any location;
- the wing volume must be greater than or equal to its original volume;
- the root bending moment must be less than or equal to that of an elliptical wing of with a half-span of 4.0 reference units
- $C_L S = 2.00$;

Table 4.8 lists some of the statistics for the O-O grid used in this optimization. The wing consists of 32 surface patches fitted with 9×9 B spline control points. The wing surface is then embedded into an FFD volume with 10 spanwise control sections and 10 chordwise control points. Figure 4.19 shows the computational mesh and the FFD setup for this optimization. The FFD volume is in turn driven by a simple linear axial curve which can control the wing's span and sweep. The wing geometry has an initial volume of $V = 0.24$ cubed reference units and a reference area of $S = 3.0$ squared reference units.

Table 4.8: Grid parameters for planform optimization

Grid Level	Nodes	Blocks	Off-wall Spacing (ref. units)	Average y^+
Coarse	1,320,000	64	1.35×10^{-6}	0.75
Fine	10,560,000	512	7.30×10^{-7}	0.32

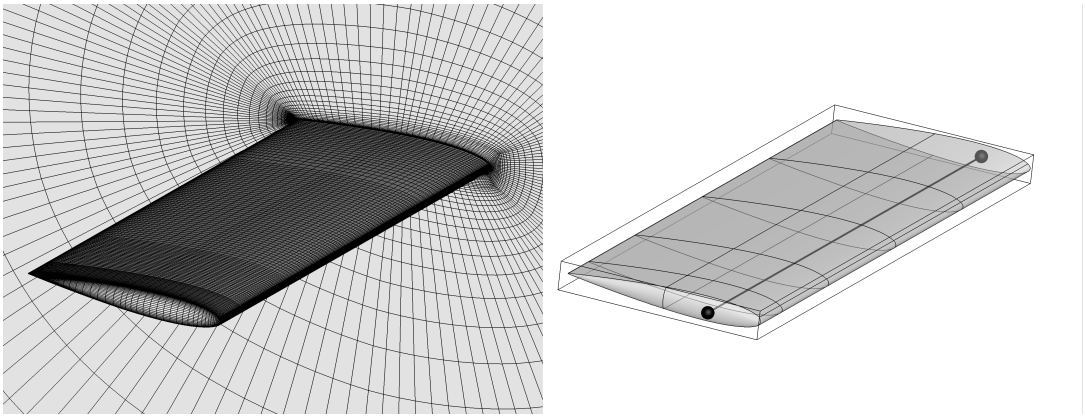


Figure 4.19: Initial mesh and FFD parameterization for planform optimization

The initial angle of attack is 1.0°. The flow analysis is performed at a Reynolds number of 20×10^6 and at both Mach 0.78 and Mach 0.85. Despite the fact that the wing chords are free to change, The final geometries are tapered such as that the overall mean aerodynamic chord is close to one, meaning that the initial Reynolds number is still valid at the end of optimization.

4.2.2 Results

Figure 4.20 shows the initial solution on the NACA0012 at Mach 0.85 at $\alpha = 2.0°$, showing a strong shock on the upper surface of the wing. The optimizer is able to produce large changes in both the planform as well as the airfoil, evidenced in Figure 4.21. Table 4.9 summarizes the results of the optimizations performed at both Mach numbers. The thickness, volume, and root bending moment constraints are active in both optimizations. The drag values are computed on the 'Fine' mesh level. As expected, the optimizer chooses to extend the span to the limit to minimize induced drag, as well as sweeping it as far back as possible to reduce wave drag. In each case, the taper of the wing is used to adjust to the optimal planform area. At the higher Mach number of 0.85, the optimizer creates a larger wing area to reduce the lift coefficient, reducing wave drag at the cost of increasing friction drag. The section shapes show significant change from the initial NACA0012 airfoil and are all free of strong shocks. The sections optimized for Mach 0.85 also have a flatter upper surface and aft camber than those optimized at Mach 0.78. Figure 4.22 shows the optimization convergence history - both optimization cases are able to run without error for over 200 iterations and achieve about three orders of magnitude reduction in

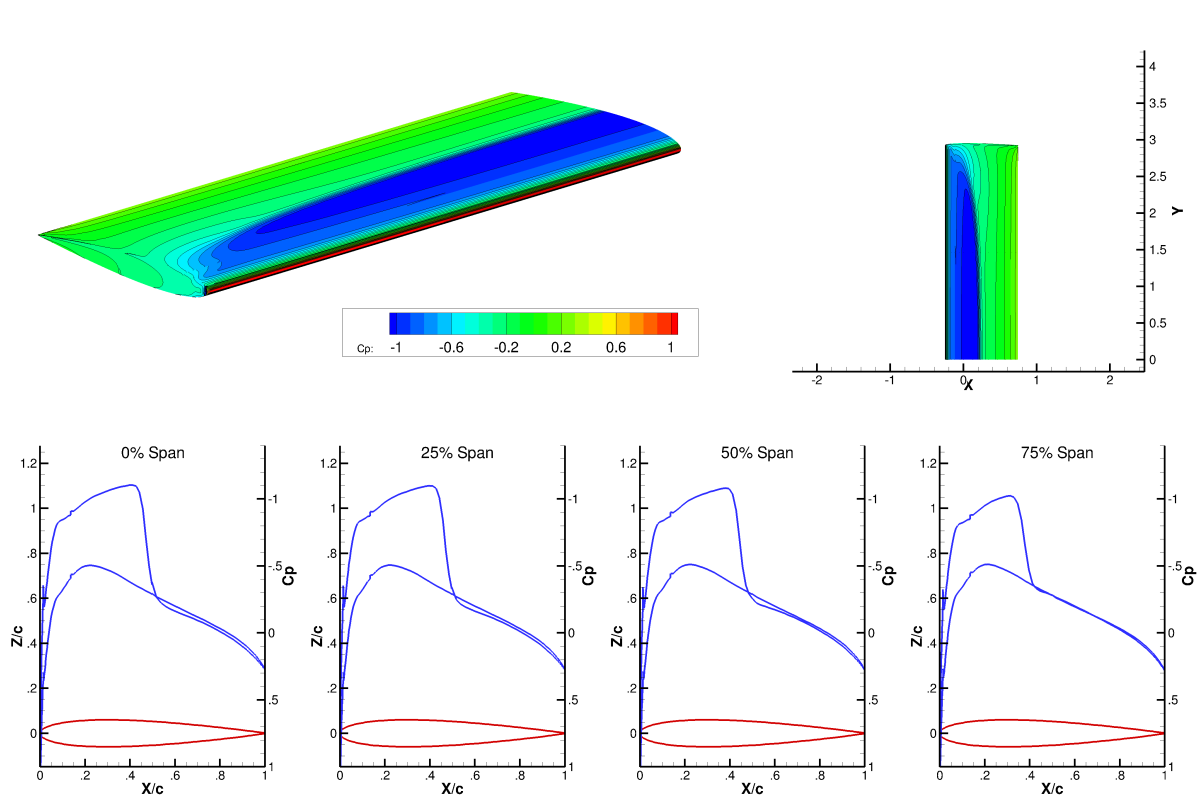


Figure 4.20: Initial flow solution for NACA0012 mesh at Mach 0.85

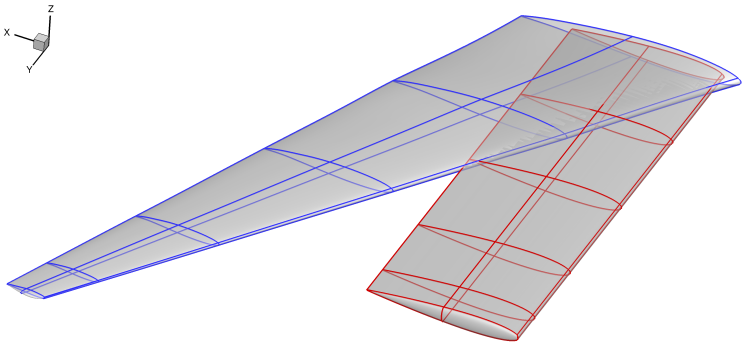


Figure 4.21: The intitial geometry compared to the optimized geometry at Mach 0.78

optimality. Figures 4.23 and 4.24 show the optimized planform geometries, pressure contours, as well as section data at various spanwise locations for the optimized wings, all computed on the Fine mesh level. An oscillation in the pressure distribution appears near the tip in Figure 4.24 in the fine mesh analysis, which was not apparent in the coarse mesh optimized surface. This is similar to the fine mesh results from the CRM optimization, suggesting that optimization at a higher grid level is necessary for fine tuning the geometry.

Table 4.9: Results summary

	Mach 0.78	**Mach 0.85**
C_L	0.608	0.563
S	3.29	3.55
$C_D S$	0.0682	0.0727
C_D (counts)	207	205
L/D	29.3	27.5
α	3.19°	2.83°
Span	3.95	3.95
Root Chord	1.42	1.36
Volume	0.24	0.25
Taper Ratio	0.17	0.32
Sweep	32.0 °	32.0°
Pressure Drag %	74.4	75.1
Friction Drag %	25.6	24.9

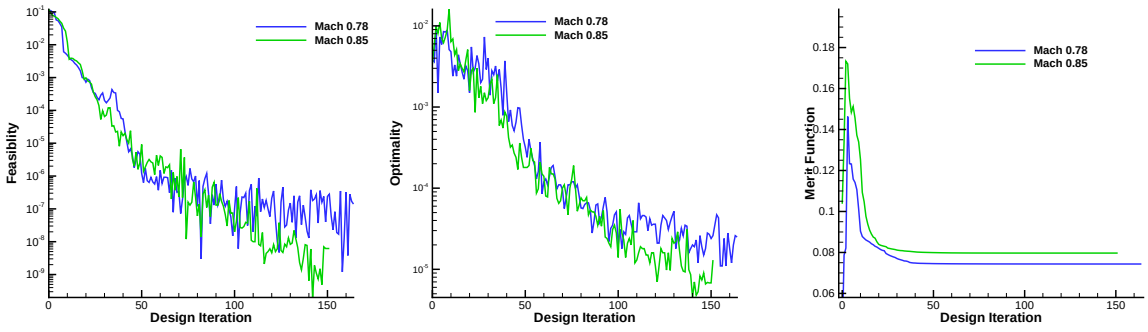


Figure 4.22: Optimization convergence for planform optimization

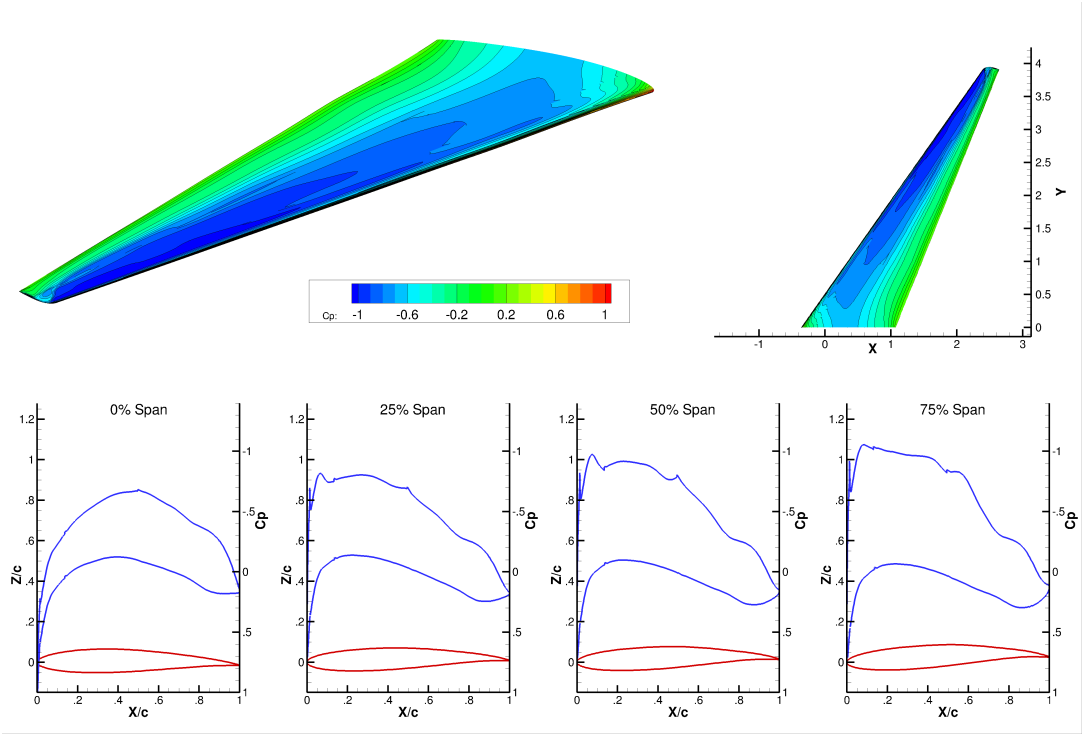


Figure 4.23: Optimized planform and section geometry and pressure contours at Mach 0.78

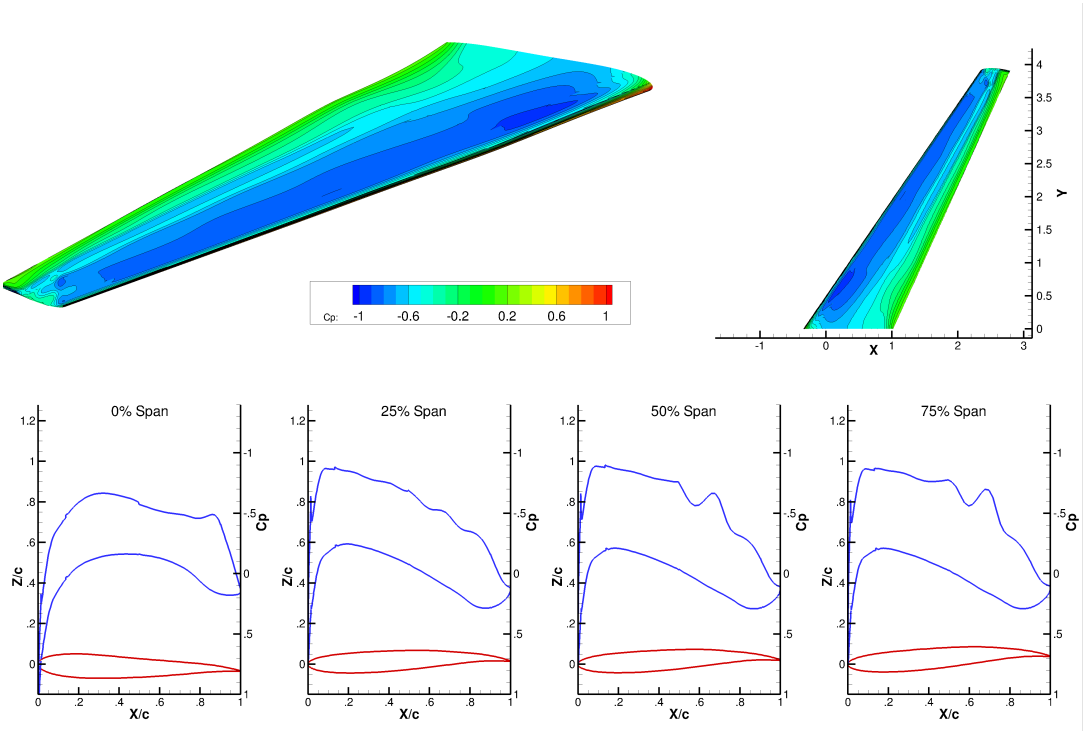


Figure 4.24: Optimized planform and section geometry and pressure contours at Mach 0.85

Table 4.10: Grid parameters for the wing based on the Boeing 737-900

Grid Level	Blocks	Nodes	Off-wall Spacing (ref. units)	Average y^+
Coarse	1,631,000	64	7.8×10^{-7}	0.45
Fine	13,048,000	512	3.2×10^{-7}	0.23

4.3 Nonplanar Wing Tip Optimization

4.3.1 Problem Definition

Hicken and Zingg[13] worked on winglet design starting from a flat plate for a rectangular NACA0012 in inviscid, subsonic flow. This was later studied by Osusky and Zingg[27] in turbulent flow with limited success. This optimization case presents a different take on wingtip optimization. The initial geometry and flow conditions are similar to inviscid, aerostructural optimization done by Khosravi and Zingg.[19] The initial wing geometry has a planform based on the Boeing 737-900 with RAE2822 airfoil sections and no initial twist. The optimization is done at a Mach number of 0.78 and a Reynolds number of 20 million. The design variables are the airfoil sections, geometric twist, and the angle of attack. At the wingtip, the taper, sweep, and dihedral are free. In addition to design variable bounds, the following constraints are imposed:

- The wing must maintain at least 75% of its original thickness at any location;
- the initial volume of the wing from root to 90% of the span must be maintained;
- the wing must produce lift equal to the original wing at a C_L of 0.50;
- the wingtip cannot extend past the span of the initial wing;

The Boeing 737 wing is normalized by using the MAC of 3.96 metres as the reference unit. The initial wing geometry has a projected area of $S = 2.899$ squared reference units. The initial volume from the wing root to 90% of the span is $V = 0.1856$ cubed reference units, and is maintained throughout the optimization to constrain the wing sections from becoming too thin. There is no volume constraint on the wingtip design. Each optimization may alter the projected area as long as it produces a non-dimensionalized lift area of $C_L S = 1.449$. The overall reduction in drag is computed based on $C_D S$.

A series of configurations are optimized, including optimization with only section shape and twist variables, a raked wingtip, and winglet up/down. The cases are optimized on the same O-grid mesh outlined in Table 4.10. The computational mesh is mapped to a control mesh in Figure 4.25, which is embedded in two axial curve driven FFD volumes. The first FFD volume governs the majority of the wing from the root to 90% of the span; this portion of the wing can optimize section shape and twist

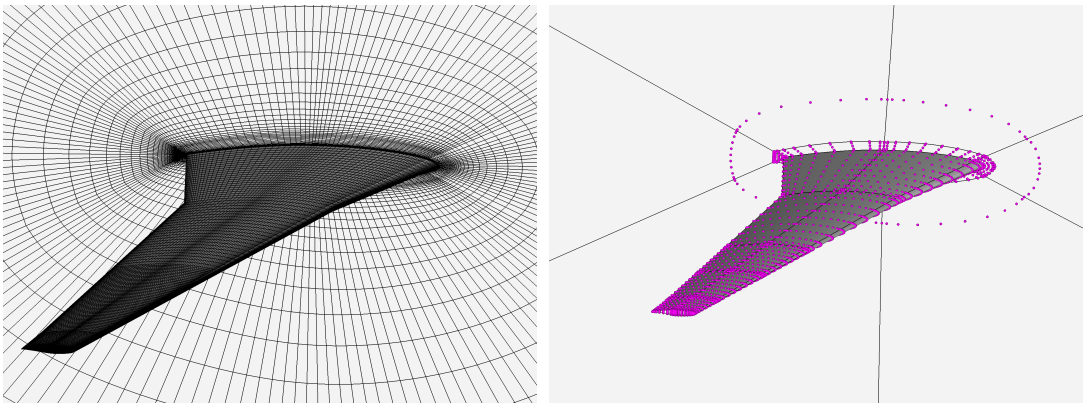


Figure 4.25: Initial wing - the computational grid (left) and the B-spline control mesh (right)

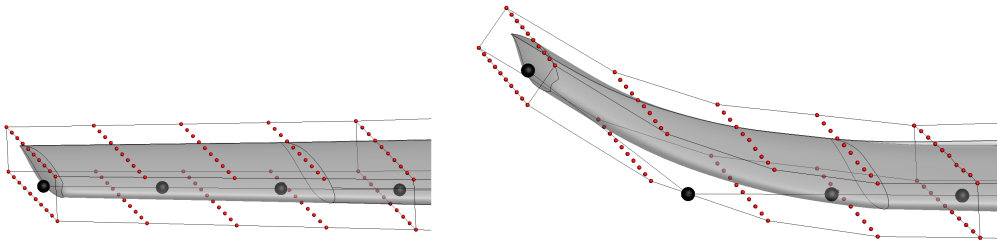


Figure 4.26: The FFD parameterization of the wingtip - initial geometry (left) and an example of how the control point deflection can produce a winglet (right)

but not the planform. The second FFD volume, shown in Figure 4.26, governs the wingtip; in addition to section and twist, the taper is free and the axial curve control points are allowed to move to adjust sweep and wingtip height. The wingtip axial curve is parameterized with a fourth order B-spline curve to ensure a smooth blend from the planar wing to the winglet.

In the first case, only the wing section and twist are design variables. In the raked wingtip, the optimizer also designs the wing section and twist, but the wingtip is given the additional freedom to design taper and sweep. In the winglet optimization, the wing has the same degrees of freedom as the raked tip, with the additional freedom for the outermost axial control point to move vertically up or down. The second outermost axial control point has some freedom to move in the spanwise direction to control the radius of curvature of the winglet transition. Finally, all axial control points cannot extend the wing past its original span. The maximum height to span ratio (h/b) allowed for each winglet is 10%.

In addition, a wingtip fence and a split-tip configuration are optimized. Due to the complexity of the geometry, these cases are performed on separate grids, each with similar off-wall spacing to that in Table

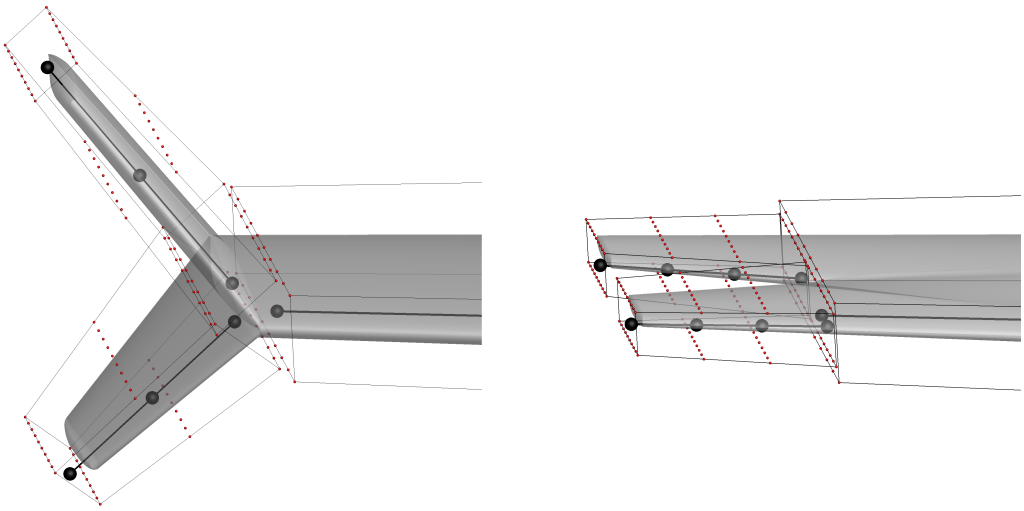


Figure 4.27: Initial geometry and FFD volume of the wingtip fence (left) and the split-tip (right)

4.10. In the wingtip fence, the wingtip is split into two extensions - one up and one down. Each extension is governed by an axial FFD volume and is allowed to change taper, sweep, and height independently. The split-tip is parameterized similarly, with the wingtips split in the chordwise direction. Figure 4.27 shows the FFD parameterization of both initial geometries. In both cases, the entire wing section and twist are designed while the total span, volume and lift constraints are kept at the same values as in the first four cases. In addition, the maximum distance between the upper and lower wingtip for both cases is restricted to the same maximum h/b ratio for the winglets.

4.3.2 Results

Table 4.11 summarizes the optimization results. While the optimizations are performed on the 'Coarse' grid level, all of the drag values are computed on the 'Fine' grid level, shown in Table 4.10. Although the optimization meshes for the wingtip fence and split-tip are different than the previous four due to their unique topology, the final drag values are computed on a mesh with around 13 million nodes to provide an accurate comparison to the rest of the geometries. Figure 4.28 shows the initial wing and the wing with optimized section variables. Figures 4.29 and 4.30 show the geometries and pressure contours for the wings with optimized wingtips. All of the optimizations manage to eliminate the shock on the initial geometry and create smoother pressure contours over the wing surface. In the cases with nonplanar geometries, the optimizer is trading off between induced drag and surface friction drag.

In the section optimization, the optimizer adjusts the airfoil section and quarter-chord twist to remove shock and optimize the spanwise lift distribution. In the raked tip, the wingtip is tapered and swept back as far as possible, providing a small improvement. When the optimizer was given the freedom to choose

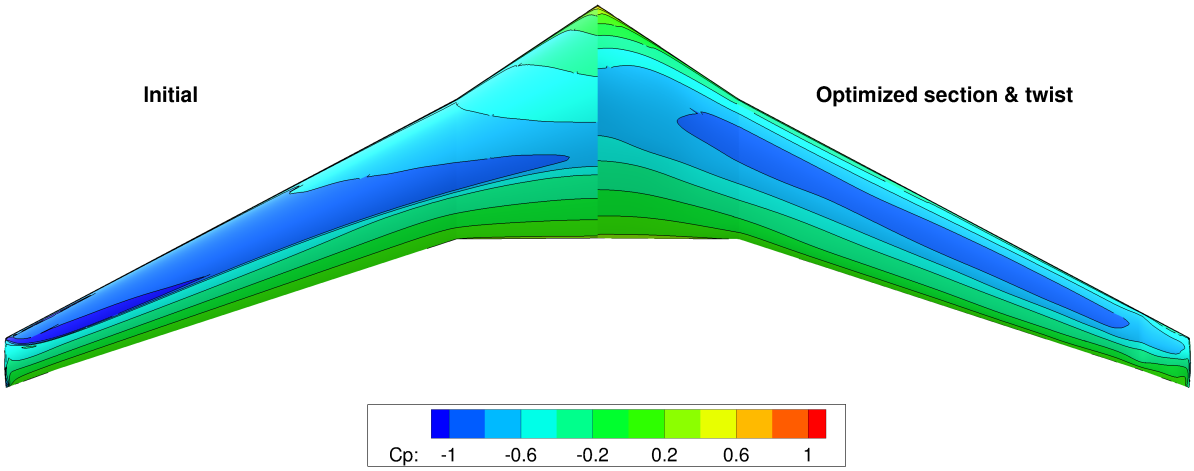


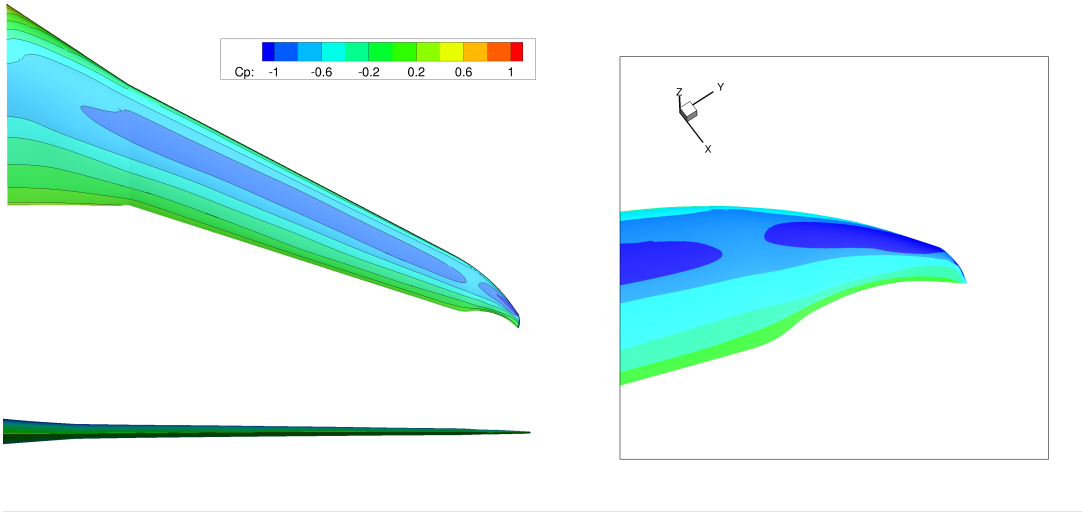
Figure 4.28: Initial and optimized pressure contours with no wingtip optimization

Table 4.11: Results summary for RANS wingtip optimization

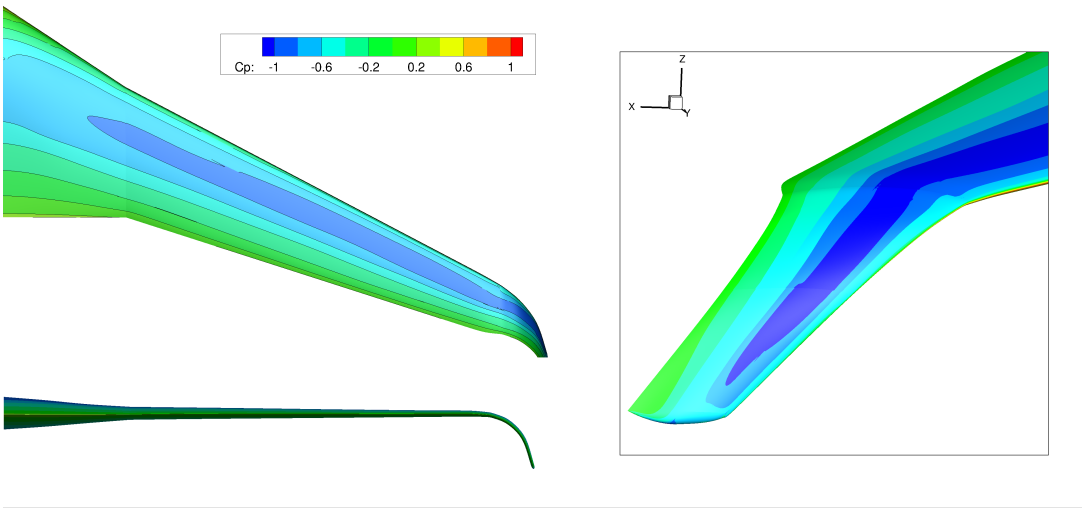
	Initial	Section only	Raked Tip	Winglet Down	Winglet Up	Wingtip Fence	Split-tip
S	2.899	2.899	2.848	2.874	2.880	2.889	2.885
C_L	0.500	0.500	0.509	0.504	0.503	0.502	0.502
C_D (counts)	171.2	154.9	156.4	151.4	152.8	151.7	156.4
L/D	29.15	32.25	32.53	33.29	32.89	33.05	32.09
α	2.41°	1.44°	1.57°	1.96°	1.83°	1.98°	1.05°
S_{wet}	2.962	2.961	2.906	2.984	2.987	3.026	2.969
Pressure Drag	69.75%	65.78%	66.16%	64.38%	64.59%	65.46%	65.36%
Friction Drag	30.25%	34.22%	33.84%	35.62%	35.41%	34.55%	34.64%
$\Delta C_D S$	0.0%	-9.7%	-10.4%	-12.5%	-11.5%	-11.8%	-9.3%

a winglet up or down, it always chose to create a winglet down. As a result, an additional optimization was run with the winglet constrained to only move upwards. The optimized winglet down leads to the greatest improvement in drag, suggesting that even when the wingtip is initially planar, the optimizer can determine immediately that a downward winglet is the optimal solution. In both the winglet up and down configurations, the optimizer designs the winglet to reach the maximum h/b, suggesting that large winglets are beneficial despite the increase in wetted area. Both winglets are swept back as far as possible; the winglet down has washout while the winglet up has wash-in. Despite the formation of a winglet, the optimizer tapered the wingtip sections to decrease the projected area and increase the C_L.

The wingtip fence shows a significant improvement in drag compared to the initial geometry, despite having the greatest wetted area. An interesting feature in this case is that the downward facing tip is tapered less and has greater area than the upward tip. While both wingtips are initially level in



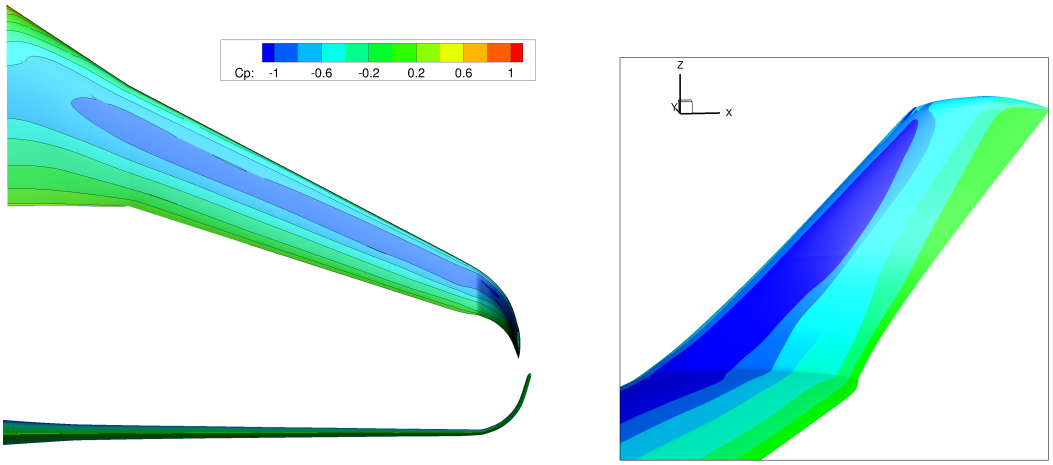
(a) Optimized raked wingtip



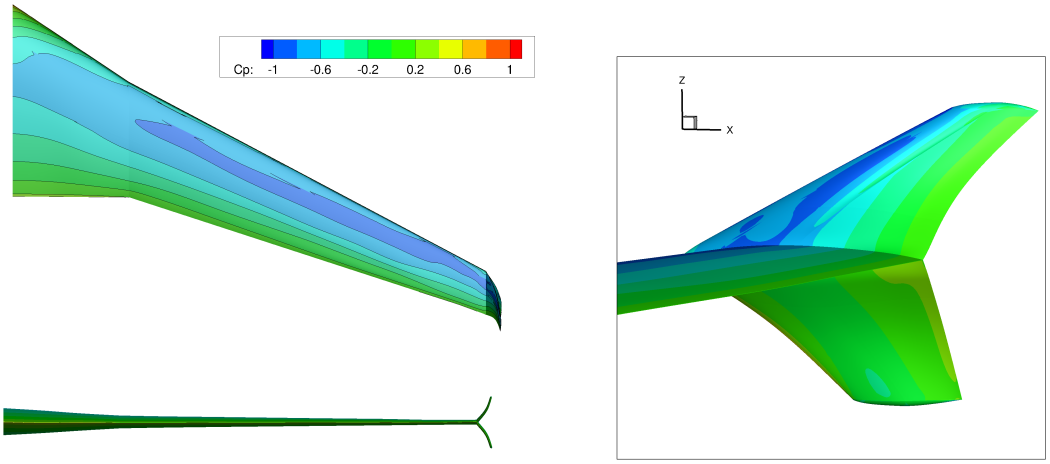
(b) Optimized winglet down

Figure 4.29: Optimized geometries and wingtips for raked and winglet down configurations

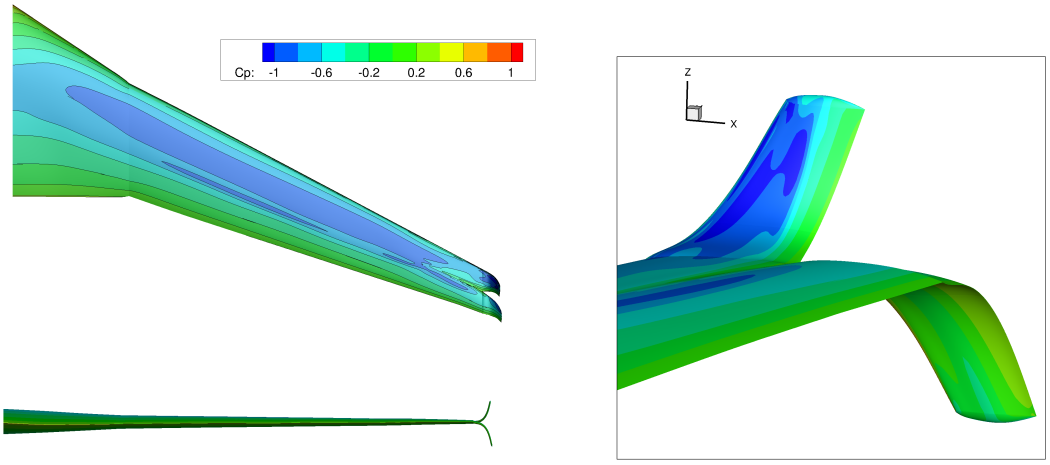
the split-tip, the optimizer moves the forward tip upward and the aft tip downward, which agrees with results from Hicken and Zingg.[13] While the optimized split-tip geometry bears some resemblance to the wingtip fence, it offers less reduction in drag due to high turbulence at the junction between the main wing and the two tips, which may be an artifact of the turbulence model. In both of these two cases, the downward wingtip has wash-out and the upward wingtip has wash-in.



(a) Optimized winglet up



(b) Optimized wingtip fence



(c) Optimized split-tip

Figure 4.30: Optimized geometries and wingtips for winglet up, wingtip fence and split-tip configurations

4.4 Box-Wing Optimization

4.4.1 Problem Definition

Finally, the box-wing optimization case studied by Osusky[26] and Hicken and Zingg[13] is revisited. This time, the box-wing geometry is parameterized using an FFD volume and meshed using an O-grid strategy. The initial geometry is the same as the one from Osusky,[26] a box-wing with a rectangular planform and NACA0012 sections. The chord is 1 reference unit, the span is 3.0 reference units, and the horizontal wings are spaced 0.4 reference units apart in the z direction.

The geometry is generated intitially using the Genair CAD Package,[9] which allows for the creation of smooth junctions between the vertical and horizontal wing components. This time, in addition to the wing twist, the optimizer is free to manipulate the section shape. The wing is initially untwisted, but it can vary each section twist by up to $\pm 4°$. The thickness of each section must be greater than or equal to 85% of the original thickness and the overall volume cannot be lower than the initial volume.

The optimization uses a 2.64 million node mesh with an O-O topology. The grid has an average off-wall spacing of 8×10^{-7} reference units, giving an average y^+ of 0.16. The wing consists of 60 surface patches fitted with 9×9 B-spline control points. The wing surface is then embedded into an FFD volume with 10 spanwise control sections from root to the tip of each horizontal wing, and 10 control sections along the junction and vertical wing. Each control section has 10 FFD control points. The wing geometry has an initial volume of $V = 0.512$ cubed reference units and a reference area of $S_{\text{ref}} = 2.9$ squared reference units, which is constant during the optimization. The optimizer is free to choose the angle of attack and change the section to meet lift constraint $C_L = 0.25$. The initial angle of attack is 2.0 degrees. The flow analysis is performed at a Reynolds number of 5×10^6 and at Mach 0.50.

4.4.2 Results

Table 4.12 summarizes the initial and optimized drag values, computed on the coarse optimization mesh. Figure 4.31 shows the optimization convergence for this case; the optimality is reduced by roughly three orders of magnitude. Figure 4.32a shows the initial box-wing geometry with NACA0012 sections. The isometric view shows that the interior of the box-wing shows a high amount of suction in the flow as it is squeezed through the thickest portion of the airfoil and at the junction. Figure 4.32b shows the optimized geometry, with the significantly reduced suction between the top and bottom wings. The top of the lower wing and the bottom of the upper wing are made nearly flat to maximize the cross sectional area between the wings. Interestingly, the lower wing also appears to produce negative lift - this is

Table 4.12: Results summary

	Initial box-wing	Optimized
C_L	0.250	0.250
C_D (counts)	306	226
L/D	8.28	11.18
α	2.38°	2.69°
Pressure Drag	45.5 %	37.8 %
Friction Drag	54.5 %	62.2 %

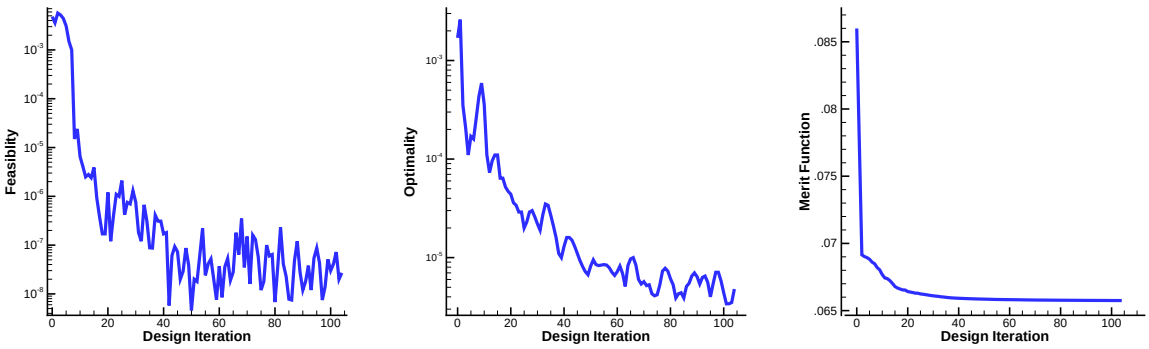
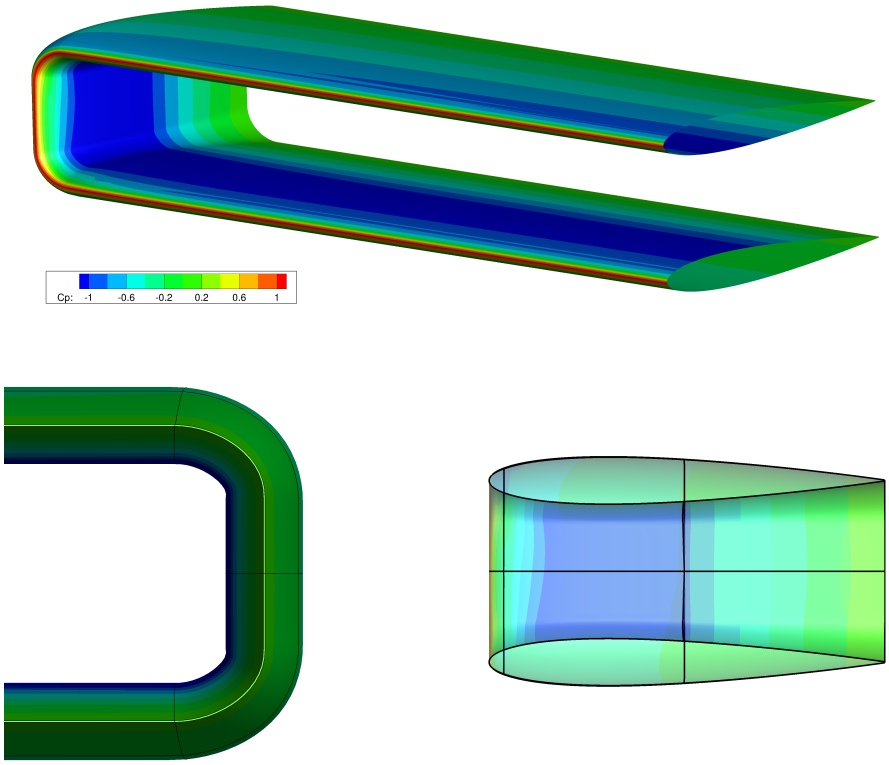


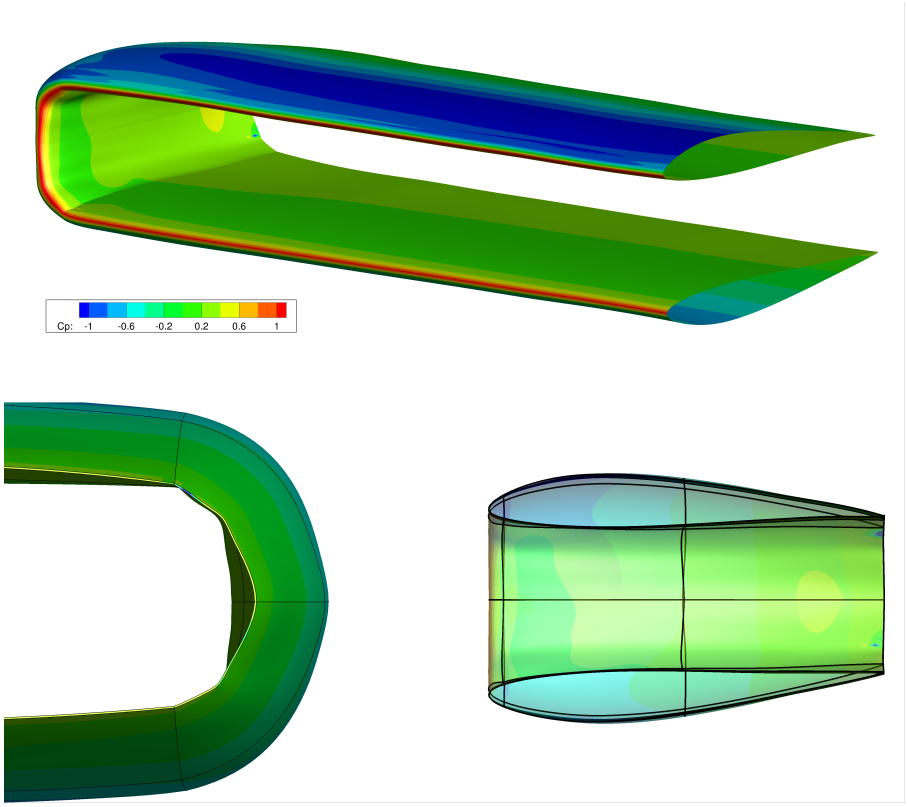
Figure 4.31: Optimization convergence for box-wing optimization

likely a consequence of the fairly low C_L constraint imposed on the optimization. The back view of the junction shows that it has become more rounded as opposed to the 90° bends in in the initial geometry.

Although a different mesh is used from that of Osusky,[26] her case had an initial C_D of 0.0310, which is fairly close to the initial drag in this case. The mesh she used for optimization and analysis had 4.3 million nodes, similar to the one used here. As a result, some comparisons can be made about the convergence of the optimization and the geometry. By enabling section shape as a design variable, the optimizer is able to make more significant geometry changes and make a greater reduction in drag compared to Osusky. Since the FFD section and twist variables are defined orthogonal to the axial curve, the shape changes result in a smoother geometry at the vertical wing junction, preventing the numerous flow solver errors that were encountered in Osusky's case. The optimality measure in Figure 4.31 is reduced gradually over 100 iterations and could be run for a greater period of time, while the optimality in Osusky's work drops suddenly after 16 function evaluations and causes SNOPT to exit abnormally.



(a) Initial box-wing geometry with NACA0012 Sections



(b) Optimized box-wing geometry

Figure 4.32: Initial and optimized box-wing geometries and pressure contours

Chapter 5

Conclusions and Recommendations

5.1 Conclusions

Jetstream has been demonstrated to be an effective, robust optimization tool in aerodynamic optimization using the RANS equations. Through trial and error, a set of guidelines have been identified which have improved the performance of the optimizer when incorporating viscous and turbulent effects. Enforcing aerodynamic surface continuity through linear constraints or volumetric deformation methods has shown to reduce the incidence of mesh movement problems that can severely hinder the optimization process. Adjusting the solver time stepping parameters to account for more difficult flows has also improved the optimization process by reducing the number of failed design evaluations. Finally, a mesh topology has been presented which can improve convergence as well as handle more complex geometries.

By implementing tighter constraints on the continuity of the wing surface, the CRM wing optimization was able to run for a greater number of design iterations without encountering failure. As a result, there is a improvment in both drag and optimality reduction compared to previous work. An efficient, parallel multi-point optimization was run with the CRM wing geometry to demonstrate Jetstream's ability to handle a wide variety of cases. It was shown that, despite beginning from different initial geometries, different optimization runs of the CRM wing tend to converge to very similar geometries, suggesting that section and twist optimization in RANS is likely not a multimodal design problem.

In the planform variable optimization case, the optimizer was able to make significant changes to the airfoil section, sweep, and span. In doing so, Jetstream has shown that it is possible to make large geometric changes while resolving the physics at a high fidelity. The wingtip optimization of the B737-900 wing demonstrated Jetstream's ability to design a wide variety of non-planar geometries such as the

winglet, wingtip fence, and split-tips. Improvements in both meshing and geometry control have allowed these geometries to be analyzed in a more realistic flow than in previous work. Finally, the box-wing optimization provided another example of Jetstream's ability to improve complex, unconventional wing designs.

5.2 Recommendations for Future Work

One of the main difficulties encountered in RANS optimization is in obtaining converged flow solutions during each design iteration. This is especially important in multi-point optimization, where the solver is prone to fail at high-lift design points. This causes the optimizer to slow down significantly and in the worst case, stall completely. As outlined in Chapter 3, a majority of these issues occur in the continuation phase and are very sensitive to parameters such as the time step acceleration rate b or the residual drop tolerance for switching to the inexact-Newton phase. To this end, research into robust and efficient continuation methods for the RANS equations could yield large improvements in the optimizer's performance.

Another issue encountered in this work was the error between analyzing the optimized geometry on the coarse mesh and the fine mesh. This underscores the need for higher-order flow solutions, as well as higher order gradient calculations for optimization. Some more interesting avenues of research include multimodality - while several initial geometries were explored in the CRM optimization, this was not a thorough study. A multimodality study using a Sobol sampling or random sampling method would yield more definitive results, especially in a design case with large geometric freedom such as the planform design case.

A natural extension for Jetstream is to incorporate a structural model.[32] As a multi-disciplinary optimizer, constraints on cases such as the planform design case can be derived from more realistic structural and weight requirements, instead of limits determined by the user. This can eventually be applied to full fuselage and wing configurations or blended-wing bodies. Inplementing a model to predict laminar-turbulent transition in the optimizer can allow for the design of laminar flow airfoils and wings. Finally, extending Jetstream to optimize under unsteady flow conditions will open the door to designing wings for UAVs, which operate at flight regimes that cannot be adequately modelled by RANS.

References

[1] W. K. Anderson and D. L. Bonhaus. Airfoil design on unstructured grids for turbulent flows. *AIAA Journal*, 37(2):185–191, 1999.

[2] W. K. Anderson, S. L. Karman, and C. Burdyshaw. Geometry parameterization method for multidisciplinary applications. *AIAA Journal*, 47(6):1568–1578, June 2009.

[3] O. Chernukhin and D. W. Zingg. Multimodality and global optimization in aerodynamic design. *AIAA Journal*, 51(6):1342–1354, 2013.

[4] J. Driver and D. W. Zingg. Numerical aerodynamic optimization incorporating laminar-turbulent transition-prediction. *AIAA Journal*, 45(8):1810–1818, 2007.

[5] J. Elliot and J. Peraire. Aerodynamic optimization of unstructured meshes with viscous effects. In *13th Computational Fluid Dynamics Conference*, number AIAA 1997-1849, Snowmass, CO, June 1997.

[6] J. Elliot and J. Peraire. Practical 3D aerodynamic design and optimization using unstructured meshes. *AIAA Journal*, 35(9), September 1997.

[7] B. Epstein, A. Jameson, S. Peigin, D. Roman, N. Harrison, and J. Vassberg. Comparative study of 3D wing drag minimization by different optimization techniques. In *46th AIAA Aerospace Sciences Meeting*, number AIAA-2008-326, Reno, Nevada, U.S.A., January 2008.

[8] H. Gagnon and D.W. Zingg. High-fidelity aerodynamic shape optimization of unconventional aircraft through axial deformation. In *AIAA Science and Technology Forum and Exposition: 52nd Aerospace Sciences Meeting*, number AIAA-2014-0908, National Harbor, Maryland, U.S.A., January 2014.

[9] H. Gagnon and D.W. Zingg. Two-level free-form and axial deformation for exploratory aerodynamic shape optimization. *AIAA Journal*, 53(7):2015–2026, July 2015.

[10] P.E. Gill, W. Murray, and M. A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review*, 47:99–131, 2005.

[11] J. E. Hicken and D. W. Zingg. Parallel Newton-Krylov solver for the Euler equations discretized using simultaneous approximation terms. *AIAA Journal*, 46(11):2773–2786, November 2008.

[12] J. E. Hicken and D. W. Zingg. Aerodynamic optimization with integrated geometry parameterization and mesh movement. *AIAA Journal*, 48(2):401–413, February 2010.

[13] J. E. Hicken and D. W. Zingg. Induced-drag minimization of nonplanar geometries based on the Euler equations. *AIAA Journal*, 48(11):2564–2575, November 2010.

[14] J. E. Hicken and D. W. Zingg. A simplified and flexible variant of GCROT for solving nonsymmetric linear systems. *SIAM Journal on Scientific Computing*, 32(3):1672–1694, June 2010.

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

[16] A. Jameson, N.A. Pierce, and L. Martinelli. Optimum aerodynamic design using the Navier-Stokes equations. *Theoretical and Computational Fluid Dynamics*, 10:213–237, 1998.

[17] A. Jameson and J. Reuther. Control theory based airfoil design using Euler equations. In *5th Symposium on Multidisciplinary Analysis and Optimization*, number AIAA 1994-4272, Septemner 1994.

[18] P. W. Jansen, R. E. Perez, and J. R. R. A. Martins. Aerostructural optimization of nonplanar lifting surfaces. *Journal of Aircraft*, 47(5):1490–1503, September-October 2010.

[19] S. Khosravi and D. W. Zingg. A numerical optimization study on winglets. In *15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, number AIAA 2014-2173, Atlanta, Georgia, U.S.A., June 2014.

[20] S. T. Ledoux, D. P. Young, S. Fugal, J. Elliot, W. P. Huffman, and R. G. Melvin. An updated study for the AIAA aerodynamic design optimization design group test case-4. In *AIAA Science and Technology Forum and Exposition 2014: 53rd Aerospace Sciences Meeting*, number AIAA 2015-1717, Kissimmee, Florida, January 2015.

[21] C. Lee, D. Koo, K. Telidetzki, H. Buckley, and D. W. Zingg. Aerodynamic shape optimization of benchmark problems using jetstream. In *AIAA Science and Technology Forum and Exposition 2014: 53rd Aerospace Sciences Meeting*, number AIAA 2015-0262, Kissimeee, Florida, January 2015.

[22] Z. Lyu and J.R.R.A. Martins. Aerodynamic shape optimization studies of a blended-wing-body aircraft. *Journal of Aircraft*, 51(5):1604–1617, 2014.

[23] M. Nemec, D. W. Zingg, and T. H. Pulliam. Multipoint and multi-objective aerodynamic shape optimization. *AIAA Journal*, 42(6):1057–1065, June 2004.

[24] E. J. Nielson and W. K. Anderson. Aerodynamic optimization on unstructured meshes using the Navier-Stokes equations. *AIAA Journal*, 37(11):1411–1419, November 1999.

[25] S. A. Ning and I. Kroo. Multidisciplinary considerations in the design of wings and wing tip devices. *Journal of Aircraft*, 47(2):534–543, March-April 2010.

[26] L. Osusky. *A Numerical Methodology for Aerodynamic Shape Optimization in Turbulent Flow Enabling Large Geometric Variation*. PhD thesis, University of Toronto, December 2013.

[27] L. Osusky, H. P. Buckley, T. A. Reist, and D. W. Zingg. Drag minimization based on the Navier-Stokes equations using a Newton-Krylov approach. *AIAA Journal*, 53(6):1555–1577, June 2015.

[28] M. Osusky and D. W. Zingg. A parallel Newton-Krylov-Schur flow solver for the Reynolds-averaged Navier-Stokes equations discretized using summation-by-parts operators. *AIAA Journal*, 51(12):2833–2851, December 2013.

[29] S. Peigin and B. Epstein. Robust drag minimization of aerodynamic wings in engineering environment. *Journal of Aircraft*, 43:1195–1203, 2006.

[30] O. Pironneau. On optimum design in fluid mechanics. *Journal of Fluid Mechanics*, 64(1):97–110, 1974.

[31] K. Telidetzki, L. Osusky, and D.W. Zingg. Application of Jetstream to a suite of aerodynamic shape optimization problems. In *52nd Aerospace Sciences Meeting*, number AIAA-2014-0571, National Harbor, Maryland, 2014.

[32] Z. J. Zhang, S. Khosravi, and D. W. Zingg. High-fidelity aerostructural optimization with integrated geometry parameterization and mesh movement. In *56th AIAA?ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, number AIAA 2015-1132, Kissimmee, Florida, January 2015.

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

Appendices

Appendix A

Additional CRM Multi-point Data

Table A.1: Summary of multi-point force coefficients for baseline and optimized geometries computed on the fine mesh

				Baseline		Optimized	
Case	Point	M	C_L	C_D	C_M	C_D	C_M
MP2	1	0.85	0.450	176.2	-0.1582	168.5	-0.1564
	2	0.85	0.500	201.5	-0.1747	185.8	-0.1704
	3	0.85	0.550	233.4	-0.1923	209.7	-0.1861
MP3	1	0.84	0.500	195.6	-0.1713	186.8	-0.1697
	2	0.85	0.500	201.5	-0.1747	185.7	-0.1705
	3	0.86	0.500	212.1	-0.1801	188.0	-0.1737
MP4	1	0.82	0.500	191.2	-0.1680	187.8	-0.1681
	2	0.85	0.500	201.5	-0.1747	187.8	-0.1711
	3	0.88	0.500	260.3	-0.1902	196.4	-0.1831
MP5	1	0.82	0.537	210.7	-0.1777	204.3	-0.1779
	2	0.85	0.500	201.5	-0.1747	187.0	-0.1708
	3	0.88	0.466	229.7	-0.1794	179.5	-0.1709
MP6	1	0.82	0.483	184.1	-0.1632	183.7	-0.1637
	2	0.82	0.537	210.7	-0.1777	207.7	-0.1783
	3	0.82	0.591	243.6	-0.1925	239.4	-0.1917
	4	0.85	0.450	176.3	-0.1584	170.4	-0.1579
	5	0.85	0.500	201.5	-0.1747	189.8	-0.1718
	6	0.85	0.550	233.2	-0.1923	213.7	-0.1865
	7	0.88	0.442	196.5	-0.1623	163.2	-0.1564
	8	0.88	0.466	229.7	-0.1794	186.0	-0.1742
	9	0.88	0.513	273.6	-0.1947	217.7	-0.1924

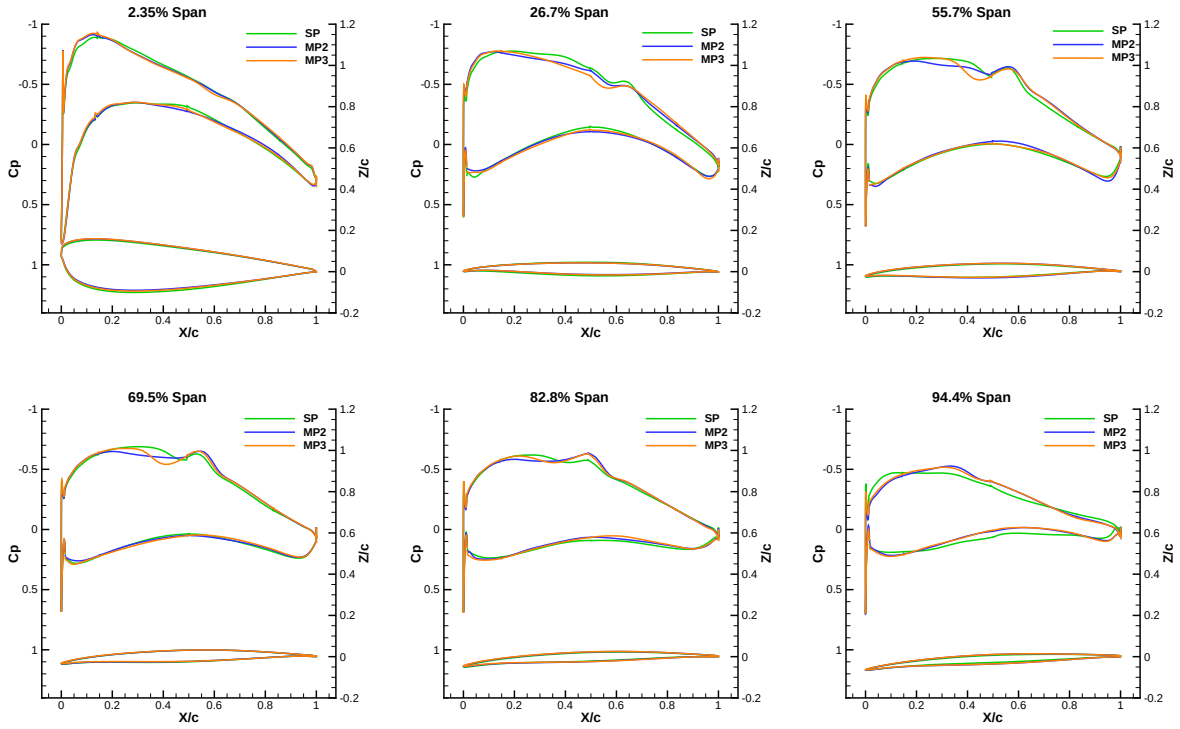


Figure A.1: Sectional pressure plots and sections for optimized CRM wings, computed on the fine mesh at the nominal condition

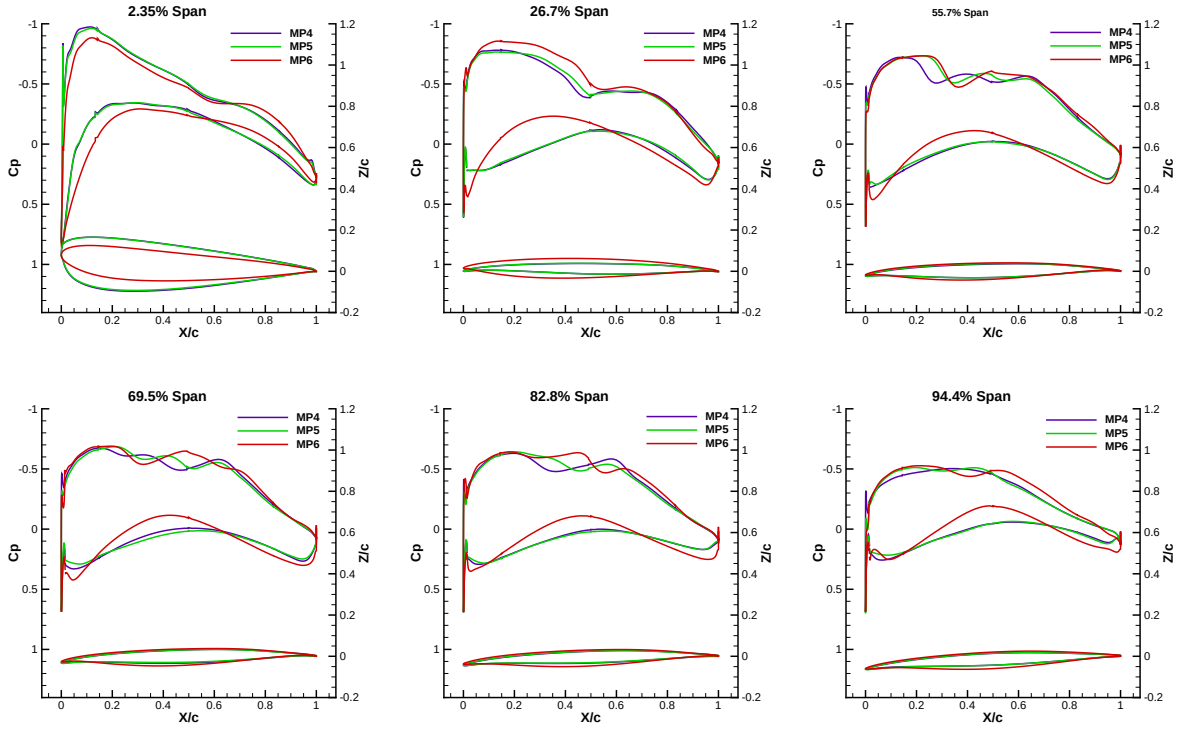


Figure A.2: Sectional pressure plots and sections for optimized CRM wings, computed on the fine mesh at the nominal condition