

Spring 2008

Wing Optimization Technique Based on Vortex Lattice Theory

Santiago Pinzon

Embry-Riddle Aeronautical University - Daytona Beach

Follow this and additional works at: <https://commons.erau.edu/db-theses>



Part of the [Aerospace Engineering Commons](#)

Scholarly Commons Citation

Pinzon, Santiago, "Wing Optimization Technique Based on Vortex Lattice Theory" (2008). *Theses - Daytona Beach*. 167.

<https://commons.erau.edu/db-theses/167>

This thesis is brought to you for free and open access by Embry-Riddle Aeronautical University – Daytona Beach at ERAU Scholarly Commons. It has been accepted for inclusion in the Theses - Daytona Beach collection by an authorized administrator of ERAU Scholarly Commons. For more information, please contact commons@erau.edu.

**WING OPTIMIZATION TECHNIQUE BASED ON VORTEX LATTICE
THEORY**

by

Santiago Pinzon

A Thesis Submitted to the

Graduate Studies Office

in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Aerospace Engineering

Embry-Riddle Aeronautical University

Daytona Beach, Florida

Spring 2008

UMI Number: EP32025

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI[®]

UMI Microform EP32025
Copyright 2011 by ProQuest LLC
All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

WING OPTIMIZATION TECHNIQUE BASED ON VORTEX LATTICE

THEORY

By
Santiago Pinzon

This thesis was prepared under the direction of the thesis committee chairman, Dr. Richard Anderson, Department of Aerospace Engineering and has been approved by the members of the thesis committee. It was submitted to the Aerospace Engineering Department and was accepted in partial fulfillment of the requirements for the degree of Master of Science in Aerospace Engineering.

THESIS COMMITTEE:



Dr. Richard Pat Anderson
Chairman



Professor Charles Eastlake
Member



Dr. Reda Mankbadi
Member



Associate Provost



Department Chair, Aerospace Engineering

4/1/08

Date

ACKNOWLEDGEMENTS

The author would like to thank Dr. Richard Pat Anderson and Professor Charles Eastlake for their valuable contribution and guidance during the development and completion of this thesis work. Special thanks go to Dr. Ray Mankbadi for his support of the project.

Special recognition goes to Mr. Snorri Gudmundsson of Cirrus Design Corporation for his unconditional support, knowledge and encouragement during all stages of this thesis development. His contribution will always be greatly appreciated.

The author cannot end without thanking his family, especially his parents Alvaro and Maria Claudia for their unconditional love, and for being a constant inspiration of success and steadily encouraging for the best.

ABSTRACT

Author: Santiago Pinzon
Title: Wing Optimization Technique Based on Vortex Lattice Theory
Institution: Embry-Riddle Aeronautical University
Degree: Master of Science in Aerospace Engineering
Year: 2008

An integrated aerodynamic optimization technique based on vortex lattice theory is being proposed. The technique in this work will aim at optimizing the geometric parameters of a general wing with NACA 65-210 airfoil sections under a low speed aerodynamic regime. This will be completed by performing an aerodynamic analysis based on vortex lattice theory with the objective of satisfying a performance index function. This function will seek to optimize the high lift-to-drag ratio of the wing at cruise velocity, while maximizing theoretical cruise range and minimizing wing weight. The optimization method has been developed using MATLAB and the focus of the study will be used by Cirrus Design Corporation and implemented under the panel method software SURFACES as an optimization tool.

TABLE OF CONTENTS

ABSTRACT.....	iv
LIST OF TABLES	viii
TABLE OF FIGURES	ix
SYMBOLS.....	xi
1. INTRODUCTION	1
1.1 Statement of the Problem.....	1
1.2 Review of the Literature	3
1.3 Statement of the Hypothesis	4
2. METHODOLOY	5
2.1 Finite Wing Aerodynamics	5
2.1.1 Introduction to Vortex Induced Drag.....	6
2.1.2 Vortex Flow, and Helmholtz’s Theorems.....	9
2.1.3 The Vortex Filament, and the Biot-Savart law	10
2.1.3.1 The Infinite Vortex Filament	11
2.1.3.2 The Semi-Infinite Vortex Filament.....	14
2.1.3.3 The Finite Vortex Filament.....	15
2.2 Vortex Lattice Theory.....	16
2.2.1 Analysis and Implementation of the Horseshoe Vortex	17
2.2.2 Location of the Control Point and Bound Vortex	25
2.2.3 Analysis and Application of the Boundary Conditions	27
2.3 Aerodynamic Performance and Analysis.....	28

2.3.1	Introduction to the Drag Components of a Wing.....	29
2.3.2	The Drag Polar	30
2.3.3	Equations of Motion for Steady, Level Flight	31
2.3.4	Range	32
3.	ANALYSIS	36
3.1	Introduction to the Computational Method.....	36
3.2	Input Geometry and Operating Conditions.....	38
3.2.1	Wing Geometry.....	38
3.2.2	Wing Section Geometry.....	41
3.2.3	Operating Conditions	43
3.3	Vortex Lattice Solver	43
3.3.1	Derivation of the Panel Geometry	44
3.3.2	Solution of the Vortex Strengths.....	46
3.3.3	Calculation of the Aerodynamic Coefficients.....	47
3.3.3.1	Computation of the Vortex Induced Drag and the Profile Drag	49
3.4	Fuel Volume and Fuel Weight Estimation.....	53
3.5	Wing Weight Estimation.....	55
3.6	Performance Computation	57
3.6.1	Performance Index Function.....	60
3.7	Initial Wing Properties and Program Initialization	62
4.	RESULTS	64
4.1	Optimal Wing Geometric Configuration	64
4.1.1	Optimal Wing Versus Initial Wing Layout.....	65

4.2	Performance Index Function Analysis	68
4.3	Aerodynamic Analysis	71
4.4	Wing Weight Comparison	78
4.5	Spanwise Distribution of the Pressure Coefficient	79
4.6	Further Considerations: C_L vs. Alpha Curve	81
4.7	Validation.....	82
4.7.1	C_l vs. Alpha Curve Comparison	82
4.7.2	Drag Polar Comparison.....	84
4.8	Error Analysis	89
4.8.1	Factors affecting the accuracy of the Vortex Lattice Method.....	90
4.8.2	Computational Error in the Numerical Analysis.....	90
5	CONCLUSIONS.....	91
6	RECOMMENDATIONS AND FUTURE WORK.....	93
	REFERENCES.....	94
	APPENDIX A	95

LIST OF TABLES

Table 3.1: Geometric Variation: Optimization Analysis.....	62
Table 4.1: Geometric Analysis.....	64
Table 4.2: Aerodynamic Ratio Analysis: Optimal Wing vs. Initial.....	71
Table 4.3: Lift to drag ratio Analysis at Original Cruise Velocity.....	74
Table 4.4: Fuel Weight Comparison.....	76
Table 4.5: Estimated Wing Weight Comparison.....	77
Table 4.6: Experimental vs. Calculated Results: C_L vs. Alpha Curve.....	84
Table 4.7: Comparison between Experimental and Calculated C_{D0}	89

TABLE OF FIGURES

Figure 2.1: Physical interpretation of the spanwise flow for a finite wing.....	6
Figure 2.2: Effect of the downwash on a typical airfoil section of a finite wing.....	7
Figure 2.3: Curved three dimensional vortex filament of strength Γ	11
Figure 2.4: The infinite vortex filament.....	12
Figure 2.5: Semi-infinite vortex filament	14
Figure 2.6: The finite vortex filament.....	15
Figure 2.7: Typical lattice arrangement for a wing planform	17
Figure 2.8: The horseshoe vortex.....	18
Figure 2.9: New nomenclature for the bound vortex filament.....	19
Figure 2.10: The horseshoe vortex implementation.....	21
Figure 2.11: Trailing vortex nomenclature	23
Figure 2.12: Control Point and Bound Vortex placement	26
Figure 2.13: The Drag Polar Diagram	31
Figure 3.1: Computational Analysis Flow Chart.....	37
Figure 3.2: Wing Geometry.....	39
Figure 3.3: NACA 65-210 Airfoil.....	42
Figure 3.4: Panel representation of a wing with unswept quarter-chord.....	45
Figure 3.5: Induced Drag Factor δ as a function of taper ratio.....	50
Figure 3.6: Profile Drag Calculation.....	51
Figure 3.7: Fuel Tank Description.....	54
Figure 3.8: Computational Routine for the Performance Analysis of Wing P1.....	58

Figure 3.9: Computational Routine for the Performance Analysis of Wing P2.....	59
Figure 3.10: Program Initialization Windows.....	63
Figure 4.1: Geometric Layout of the Optimal Wing: Loft Analysis.....	66
Figure 4.2: Geometric Layout of the Initial Wing: Loft Analysis.....	67
Figure 4.3: Performance Index Surface Plot: $\lambda = 0.4$, $\sigma_{LE} = 4.64$ deg, G.T = 2 deg.....	68
Figure 4.4: L/D Cruise Surface Plot: $\lambda = 0.4$, $\sigma_{LE} = 4.64$ deg, G.T = 2 deg.....	69
Figure 4.5: Wing Weight Surface Plot: $\lambda = 0.4$, $\sigma_{LE} = 4.64$ deg, G.T = 2 deg.....	69
Figure 4.6: Theoretical Range Surface Plot: $\lambda = 0.4$, $\sigma_{LE} = 4.64$ deg, G.T = 2 deg.....	70
Figure 4.7: Variation of $C_L^{0.5}/C_D$ with Velocity.....	72
Figure 4.8: (a) L/D Ratio Variation with Airspeed (b) L/D Ratio Variation with angle of Attack.....	73
Figure 4.9: Drag Variation with Velocity: Initial vs. Optimal Wing.....	75
Figure 4.10: Wing Fuel Volume.....	76
Figure 4.11: Theoretical Maximum Range Comparison.....	77
Figure 4.12: Estimated Wing Weight Comparison.....	78
Figure 4.13: C_p Spanwise Distribution: Optimal Wing.....	79
Figure 4.14: C_p Spanwise Distribution: Initial Wing.....	80
Figure 4.15: C_L vs. Alpha Curve: Initial vs. Optimal.....	81
Figure 4.16: C_L vs. Alpha Curve: Validation.....	83
Figure 4.17: Drag Polar Diagram (0 deg washout): Exp vs. Calculated.....	85
Figure 4.18: Drag Polar Diagram (2 deg washout): Exp vs. Calculated.....	86
Figure 4.19: Profile Drag Variation (2 deg washout): Exp vs. Calculated.....	87
Figure 4.20: Profile Drag Variation (2 deg washout): Exp vs. Calculated.....	88

SYMBOLS

AR	-	Aspect Ratio
b	-	Wing span
C_D		Total drag coefficient
C_d	-	Profile drag of the wing
C_{d_i}	-	Vortex induced drag
$C_{D,0}$	-	Zero-lift drag coefficient
C_L	-	Total lift coefficient
C_{Lmax}	-	Maximum lift coefficient
C_{LW}	-	Lift coefficient at gross weight
C_r	-	Root chord
C_t	-	Tip chord
c_l	-	Section lift coefficient
c	-	Airfoil chord
c_{avg}	-	Average panel chord
c_d	-	Profile drag coefficient
c_{dp}	-	Form drag coefficient
c_f	-	Skin friction drag coefficient
D	-	Total drag of the wing
D_i	-	Total induced drag
e	-	Oswald efficiency factor
G.T	-	Geometric twist angle

K	-	Drag-due-to-lift factor
L	-	Lift
l	-	Lift per unit span
N	-	Number of horseshoe vortex panels (halfspan)
P	-	Number of chordwise horseshoe vortex panels
r	-	Distance between bound vortex and control point
T	-	Thrust
T_{req}	-	Thrust required
U	-	Tangential velocity at the surface
\bar{V}	-	Induced velocity by a finite length vortex
V_{∞}	-	Free stream velocity
V_{stall}	-	Stall velocity
W	-	Gross weight
W_w	-	Wing weight
w	-	Downwash velocity component

Greek Symbols

α	-	Geometric angle of attack
α_{eff}	-	Effective angle of attack
α_i		Induced angle of attack
δ	-	Induced drag factor
δ_m		Slope of the mean camber line
Γ	-	Vortex strength

Φ	-	Dihedral angle
ρ_∞		Density
σ_{Le}		Leading edge sweep angle

1. INTRODUCTION

1.1 Statement of the Problem

Optimization is a fundamental aspect in the process of engineering design. It is critical in minimizing the time required to generate a successful product for a given objective. In fact, optimization has become a common groundwork in the different engineering areas that focus on breakthrough technology and performance. Such is the case behind the science of flight, especially in the field of aerodynamics which constantly seeks advancement in design, innovation in performance, and reliability. This particular field focuses optimization methods that employ various numerical techniques that seek optimal solutions and model correctly the engineering problem. It is imperative for aerodynamic optimization techniques to depend on well established methods for the computation of pertinent aerodynamic coefficients. Consequently, the correct modeling of aerodynamic phenomena is intimately related to the success of the optimization method.

Panel methods have been widely used in industry and are well established since the 1970s. The Vortex Lattice Panel Method is the method of choice for this study due to its sophistication, quick solution time, and its ability to allow rapid changes in geometry. The subsequent chapters of this work will further explain in depth the theory behind the vortex lattice method, and the reason behind its selection as the method for aerodynamic analysis. Aerodynamic optimization benefits from various numerical methods that serve

as mathematical tools for the computation of optimal solutions. Extensive literature and research exists for mathematical procedures that seek the successful optimization of engineering problems. For example, gradient base methods involve the computation of sensitivity gradients. The reliability and success of these methods generally require a smooth design space and the existence of only a single global extremum or an initial guess close enough to the global extremum that will ensure proper convergence [1]. The optimization analysis in this study is not a gradient based method. Instead, the process provides an alternate approach that focuses on numerical iterations mandated by the number of design variables. This approach involves an exhaustive numerical search of the wing's geometric parameters with the purpose of satisfying a performance index within the objective function. The process ultimately determines the best possible wing geometric configuration that outperforms the initial wing design in all of the parameters specified by the objective function. Consequently, this approach is bounded by initial constraints that filter out the geometric variables that are not pertinent to the optimization process in order to reduce the computational time. For this study, the optimization method formulated for the respective analysis is capable of converging to an optimal solution in a timely manner.

The aerospace industry is highly competitive in design efficiency, and perhaps one of the most important factors on airplane design and engineering today is multidisciplinary optimization. Any cost reduction method in the design cycle of a product becomes vital in the success of its outcome.

1.2 Review of the Literature

Multiple optimization methods in the area of aerodynamics are utilized in today's aerodynamic design process. Various techniques include the use of genetic algorithms which serve as an alternative approach to the widely used gradient methods. In [1], the author suggests a multi-objective genetic algorithm procedure to optimize the lift to drag ratio of a transonic wing coupled with the minimization of structural mass. During the study, the author explains in detail the theory behind the implementation of the genetic algorithm, and leaves the aerodynamic analysis explanation to a minimum. Due to the complexity of the multi-objective genetic algorithm optimization technique, the study in this work disregards its application. Other studies, such as the one proposed in [7], explain the importance of linking an aerodynamic analysis routine to an optimization technique. In [7], computational fluid dynamics serves as the method of choice for the integrated wing aerodynamic analysis. One of the attractive features in that study, include the integration of finite element analysis in addition to the CFD routine to minimize the weight of the wing under maximum stress and deflection constraints. In contrast, the study in this work suggests the implementation of a Vortex Lattice Method in conjunction with a numerical optimization technique to perform a wing optimization analysis. The author in [5] suggests that modern panel methods such a vortex lattice, can quickly and accurately calculate the inviscid flow properties of finite wings. Since the aerodynamic analysis in this work deals with incompressible inviscid flow, vortex lattice theory serves as the desirable aerodynamic analysis tool for this study.

1.3 Statement of the Hypothesis

The ability to efficiently link an aerodynamic method of analysis with an optimization technique can drastically reduce the duration of a design cycle and improve product efficiency. An aerodynamic optimization analysis based on Vortex Lattice Theory is being proposed. This optimization analysis utilizes a numerical panel method to efficiently model aerodynamic phenomena and maximize a performance index function.

The focus of this study will concentrate on a subsonic aerodynamic regime meant to analyze the performance of a wing under low speed aerodynamic conditions. The aerodynamic optimization technique will aim at optimizing a generic wing configuration through the numerical variation of the wing's geometric variables such as wingspan, wing area, taper ratio, washout and leading edge sweep. The geometric parameter that remains constant during the analysis is the airfoil section which defines the wing section geometry. This particular wing optimization technique disregards wing properties such as aerodynamic twist and dihedral angle. Overall, the performance index function or target function is designed to maximize the high lift to drag ratio at cruise velocity, as well as maximizing cruise range and minimizing wing weight. In order to determine the validity of the results from the aerodynamic optimization analysis, a NACA technical document that contains experimental wind tunnel test data will be used.

2. METHODOLOGY

2.1 Finite Wing Aerodynamics

An introduction to finite wing aerodynamics presents the basis of the material required in the rationale for this study. This chapter presents background theory for finite wing analysis, modern panel numerical methods such as vortex lattice, and aircraft performance.

A wing is a three dimensional body of finite span that differs aerodynamically from the airfoil, mainly due to the three-dimensional component of the flow in the spanwise direction. This spanwise flow is the product of the net imbalance pressure distribution on the wing causing the flow beneath the wing to curl around the wing tips to the low pressure region on top. As a result, the streamlines on the top surface of the wing shift towards the root chord, and the streamlines on the bottom surface shift away from the root chord of the wing. Figure 2.1 illustrates the mechanics of the net pressure imbalance and the approximate shift of a streamline over the top surface of the wing. According to [2], this flow establishes a circulatory motion that trails downstream of the wing creating a trailing vortex that is generated by the presence of wingtips. Consequently, the difference in spanwise velocity components will cause the formation of streamwise vortices distributed along the span.

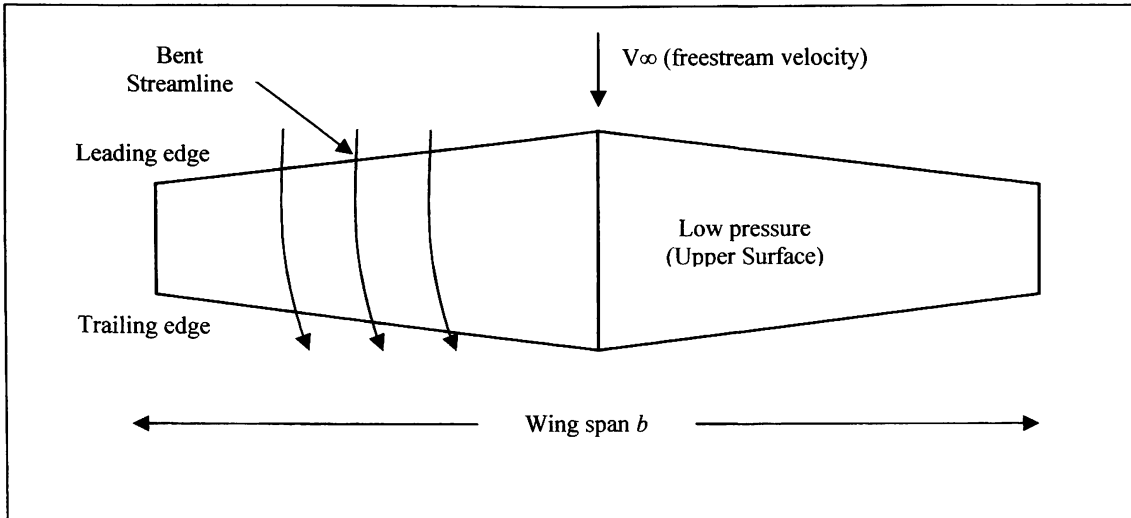


Figure 2.1: Physical interpretation of the spanwise flow for a finite wing

The wingtip vortices induce a downward velocity component on the wing that combines with the freestream velocity V_∞ to produce a local relative wind. This effect has a direct impact on the airfoil section because the local relative wind is inclined below the direction of the undisturbed free-stream flow, affecting the lift force produced by the finite wing.

2.1.1 Introduction to Vortex Induced Drag

It is important to examine the airfoil section of the wing and the physical impact that the local relative wind has on it. The inclination of the local relative wind in the downward direction changes the angle of attack experienced by the airfoil. The effective angle of attack is a consequence of this inclination, and becomes the new angle seen by the local airfoil section. The distinction between the angle of attack seen by the wing and the effective angle of attack experienced by the local wing section arises from the interaction of the downward velocity component created by the wingtips and the wing itself. This interaction is responsible for the induced angle of attack accounted for in the

difference between the geometric angle of attack and the effective angle of attack. This difference is given by the following equation:

$$\alpha_i = \alpha - \alpha_{eff} \quad [2.1]$$

A typical local airfoil section of a finite wing is presented on figure 2.2. This figure shows the effect of downwash and its impact on the geometric angle of attack.

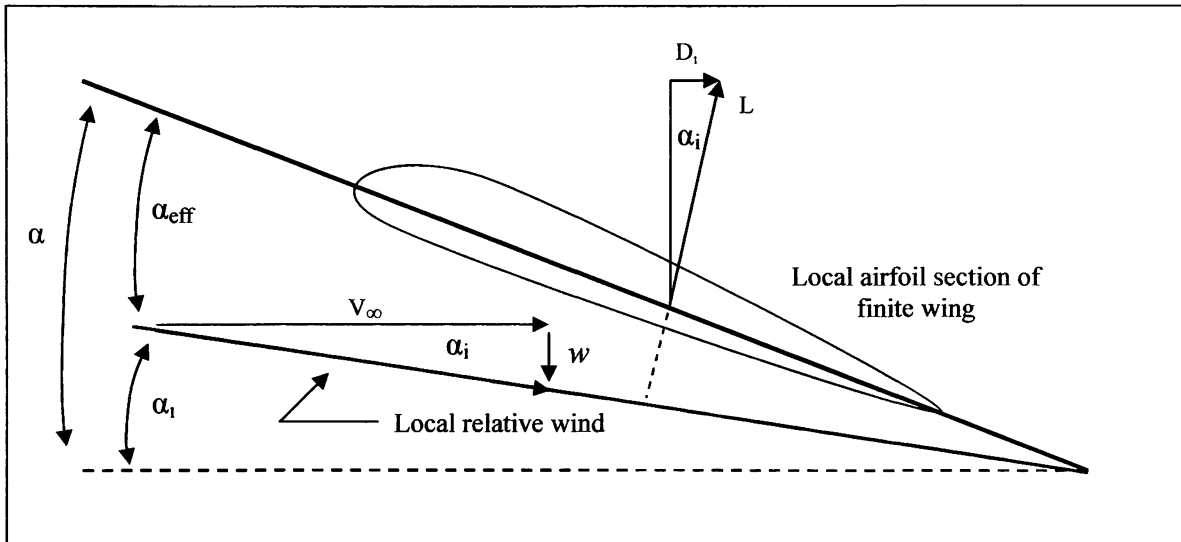


Figure 2.2: Effect of the downwash on a typical airfoil section of a finite wing

The local lift vector which is aligned perpendicular with the local relative wind is inclined from the vertical by the induced angle of attack α_i . As a result, a drag component in the direction of V_∞ is created. This inclination is a product of the downwash effect which is the main characteristic of the three dimensional flow encountered on a finite wing. This new drag component is defined as vortex induced drag denoted by D_i in figure

2.2. Clearly, induced drag is the final result of the net pressure imbalance on the finite wing that exists in the direction of V_∞ .

The aerodynamic phenomenon described in this section is critical to this particular research. The development of aerodynamic theories that evolved in the 20th century focused on mathematical explanations and methods of analysis dealing with incompressible flow over finite wings. Prandtl's classical lifting-line theory, modern numerical lifting-line method, lifting surface theory and vortex lattice theory are all different methods of aerodynamic analysis pertinent to finite wing aerodynamics. The selection of vortex lattice theory as the method of choice for the aerodynamic analysis in this work, comes as no surprise. The sophistication and simplicity of this method and its level of numerical implementation far exceeds the capacity of others. Even though Prandtl's classical lifting-line theory provides a reasonable estimate of the flow field over a wing, it is only suitable for straight thin wings at moderate high aspect ratio.

However, modern panel methods can quickly and accurately calculate the inviscid flow properties of straight and highly swept wings of low aspect ratio [5]. Furthermore, numerical panel methods convey on additional tools for the analysis of finite wing aerodynamics. These tools are the mathematical interpretation of the physical aerodynamic phenomena that govern finite wing theory. The vortex lattice numerical panel method relies on the Biot-Savart law, the curved vortex filament and the Helmholtz's theorems to explain the nature behind incompressible, inviscid, irrotational flow about a finite wing.

2.1.2 Vortex Flow, and Helmholtz's Theorems

Several tools have been developed to mathematically model incompressible inviscid flow. The beauty of relating nature through mathematics is an extraordinary achievement, and it is a set milestone in the science behind aerodynamics.

Laplace's equation is one of the most widely used and extensively studied equations in mathematical physics[2]. It is through this equation that vortex theory explains the generation of finite lift abiding the laws of irrotational and incompressible flow. The elementary vortex flow and its two dimensional vortex singularity satisfy Laplace's equation. A vortex flow is a physically possible incompressible flow, and irrotational at every point except the origin. Vortex flow can be used to model lifting surfaces through its unique flow properties. These key properties are defined by Helmholtz's vortex theorems and Kelvin's circulation theorem. The basic principles of vortex behavior are known as Helmholtz's vortex theorems and are as follows.

1. The strength of a vortex filament is constant along its length.
2. A vortex filament cannot end in a fluid; it must extend to the boundaries of the fluid.

The vortex line must be closed, extend to infinity, or end at a solid boundary.

Kelvin's circulation theorem on the other hand, states that the time rate of change of circulation around a closed curve consisting of the same fluid elements is zero. According to [3], Kelvin's theorem is proof that an initially irrotational, inviscid flow will remain irrotational.

Vortex theory is essential to the correct modeling of lifting surfaces. A sheet of vortices can support a jump in tangential velocity while the normal velocity is continuous, allowing a vortex sheet to accurately represent a lifting surface. The introduction of the two-dimensional vortex flow and its properties, paves the way to the analysis of the three dimensional vortex flow. In this case, the interaction between a three dimensional vortex filament and an arbitrary point in space is described mathematically by the Biot-Savart law.

The purpose of the next section is to explain how the theory behind vortex flow can be implemented on three dimensional lifting surfaces through the interaction of a vortex filament and the surrounding space.

2.1.3 The Vortex Filament, and the Biot-Savart law

The importance of the Biot-Savart law is apparent with the introduction of the vortex filament. The Biot-Savart law is one of the most fundamental relations in the theory of inviscid, incompressible flow [2]. It is through this law, where a mathematical expression can describe how a vortex filament induces a flow field in the surrounding space.

Consider a curved three dimensional vortex filament of strength Γ as shown in figure 2.3. The filament induces a flow in the surrounding space affecting an arbitrary point P . The Biot-Savart law states that the vortex filament segment $d\mathbf{l}$ induces a velocity or an increment in velocity at point P equal to:

$$dV = \frac{\Gamma}{4\pi} \frac{d\mathbf{l} \times \mathbf{r}}{|\mathbf{r}|^3} \quad [2.2]$$

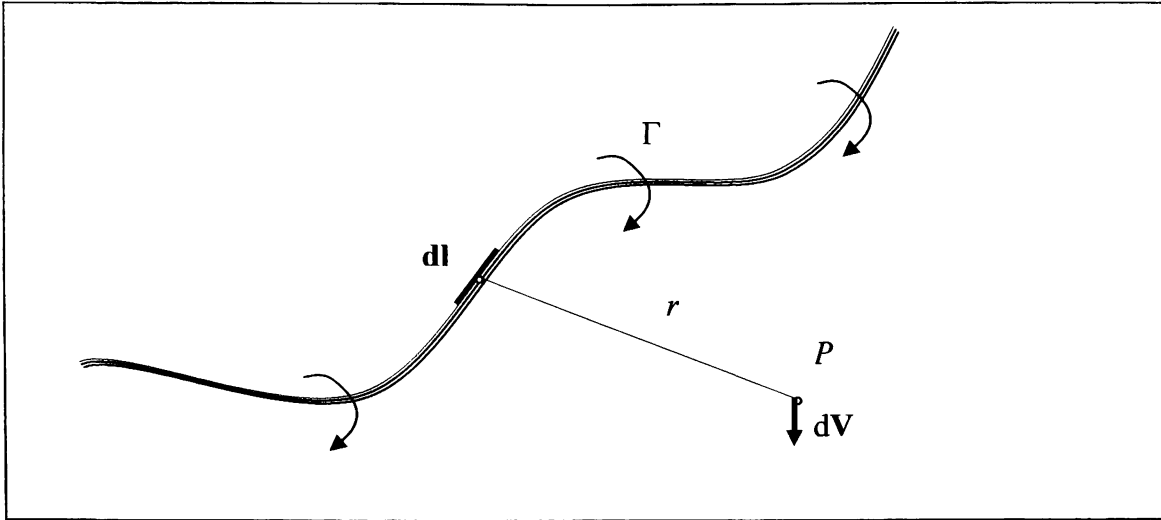


Figure 2.3: Curved three dimensional vortex filament of strength Γ

The vortex filament of strength Γ is analogous to a wire carrying an electric current inducing a magnetic field of specific strength to an arbitrary point in space. The Biot-Savart law is a general result of potential theory capable of describing inviscid, incompressible flows. The law by itself is a mathematical tool that can be used to model the interaction of various vortex filaments in conjunction with a uniform freestream. As a result, the velocity induced by the vortex filament at point P can be obtained by integrating equation 2.2 over the length of the vortex filament. The application of the integral's boundary conditions is crucial in determining its application; like modeling the flow over a finite wing.

2.1.3.1 The Infinite Vortex Filament

The first case of the application of the Biot-Savart law is the infinite vortex. Figure 2.4 shows a vortex filament of infinite length having strength Γ . The velocity induced at point P by the entire vortex filament is:

$$\mathbf{V} = \int_{-\infty}^{\infty} \frac{\Gamma}{4\pi} \frac{d\mathbf{l} \times \mathbf{r}}{|\mathbf{r}|^3} \quad [2.3]$$

Equation 2.3 calls for the definition of the vector cross product. The numerator on the equation can be expressed as:

$$d\mathbf{l} \times \mathbf{r} = |d\mathbf{l}||\mathbf{r}|\sin\theta \frac{\mathbf{V}}{|\mathbf{V}|} \quad [2.4]$$

The direction of \mathbf{V} in figure 2.4 is in the downward direction. Hence, the magnitude of the velocity at point P is given by the following equation:

$$V = \frac{\Gamma}{4\pi} \int_{-\infty}^{\infty} \frac{\sin\theta}{r^2} dl \quad [2.5]$$

The geometric relations depicted in figure 2.4 are of importance for the solution of equation 2.5.

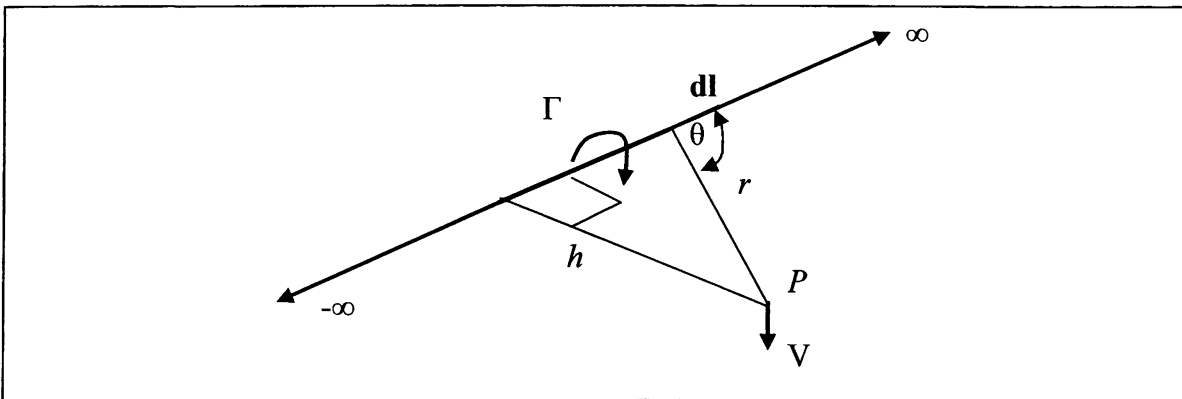


Figure 2.4: The infinite vortex filament

These geometric relations can be expressed through the following equations as:

$$r = \frac{h}{\sin \theta} \quad [2.6]$$

$$l = \frac{h}{\tan \theta} \quad [2.7]$$

$$dl = \frac{-h}{\sin^2 \theta} d\theta \quad [2.8]$$

Equations 2.6 through 2.8 are substituted in Equation 2.5 as follows:

$$V = \frac{\Gamma}{4\pi} \int_{-\infty}^{\infty} \frac{\sin \theta}{r^2} dl = \frac{\Gamma}{4\pi} \int_{-\infty}^{\infty} \frac{\sin^2 \theta}{h^2} \sin \theta \frac{-h}{\sin^2 \theta} d\theta$$

$$V = -\frac{\Gamma}{4\pi h} \int_{-\infty}^{\infty} \sin \theta d\theta = -\frac{\Gamma}{4\pi h} \int_{\pi}^0 \sin \theta d\theta$$

$$V = \frac{\Gamma}{2\pi h} \quad [2.9]$$

Thus, the velocity induced at a given point P by an infinite, straight vortex filament at a perpendicular distance h from P is simply $\Gamma/2\pi h$ [2]. This result represents the foundation of vortex lattice theory. To complete the analysis, two other areas of consideration need to be included to complete the geometric and aerodynamic analysis behind the horseshoe vortex

2.1.3.2 The Semi-Infinite Vortex Filament

The next case focuses on the semi-infinite vortex. The basis of the theory mirrors the infinite vortex case, acknowledging that the limits of integration change respectively.

Figure 2.5 represents the typical semi-infinite vortex filament.

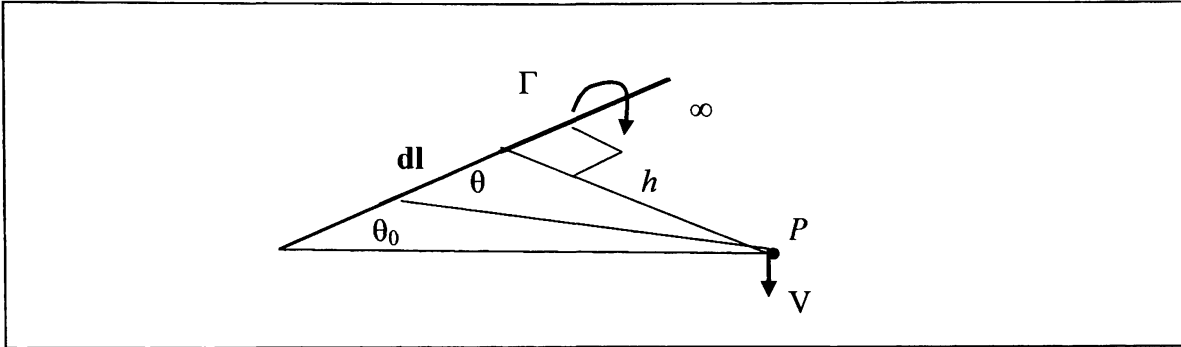


Figure 2.5: Semi-infinite vortex filament

The semi-infinite vortex filament is used in vortex lattice theory to model the vortex extending from the wing to downstream infinity. The only mathematical and conceptual variations of this vortex filament when compared to the infinite case are the limits of integration of the bounded integral.

$$V = \frac{\Gamma}{4\pi h} \int_{\theta_0}^{2\pi} \sin \theta d\theta$$

$$V = \frac{\Gamma}{4\pi h} (1 + \cos \theta_0) \quad [2.10]$$

2.1.3.3 The Finite Vortex Filament

The last case to be studied is the finite vortex. The only substantial change from the previous stated cases, relates to the bounded integral of equation 2.5. According to figure 2.6, the vortex filament of strength Γ is bounded by two angles. This bound vortex is essential to the geometry of the horseshoe vortex and its application in the traditional vortex lattice theory.

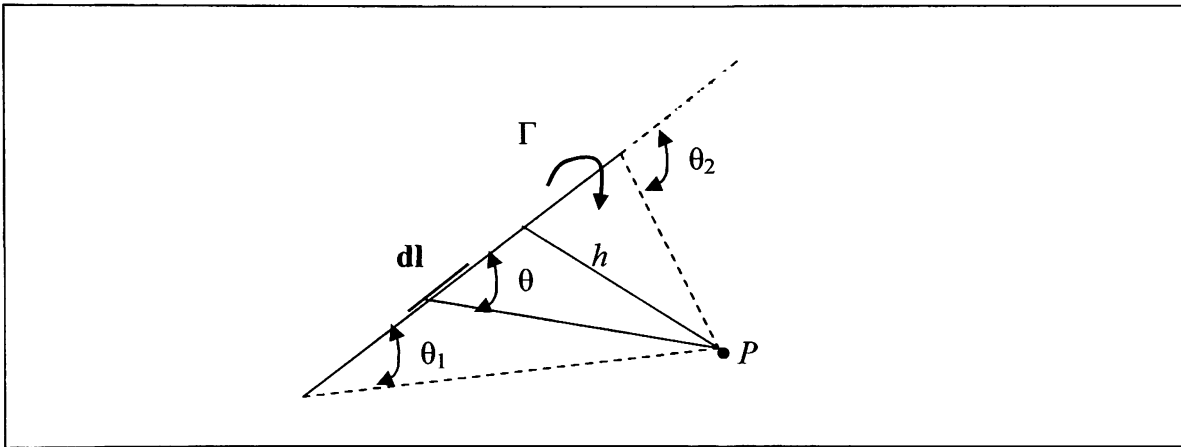


Figure 2.6: The finite vortex filament

The two angles bounding the finite vortex alter the limits of integration of equation 2.5. Therefore, the velocity induced by a vortex filament of strength Γ and a length $d\mathbf{l}$ at point P is:

$$V = \frac{\Gamma}{4\pi h} \int_{\theta_1}^{\theta_2} \sin \theta d\theta$$

$$V = \frac{\Gamma}{4\pi h} (\cos \theta_1 - \cos \theta_2) \quad [2.11]$$

The ability to model a complete lifting surface such as a wing through the use of vortex theory is only possible by the application of the three vortex filament cases. Vortex lattice theory uses the previous systems of vortices and the vortex theorems to model aerodynamic phenomena.

2.2 Vortex Lattice Theory

The vortex lattice method uses the combined analysis of the vortex filament along with the vortex theorems and the Biot-Savart law to model a complete lifting surface. The lifting surface or wing is represented by a grid of superimposed horseshoe vortices. These horseshoe vortices are each a vortex system that combines the three main vortex expressions described on the previous section. The velocity induced by each horseshoe vortex at a specific control point is calculated using the Biot-Savart law. According to [4], a summation is performed for all control points on the wing to produce a set of linear algebraic equations for the horseshoe vortex strengths that satisfy the boundary condition of no flow through the wing. In addition, the vortex strengths are related to the wing circulation and the pressure differential between the upper and lower surfaces of the wing. The vortex lattice method gets its name from the geometric distribution of the horseshoe vortices over the wing surface, which simulates trapezoidal panels or finite elements commonly known as lattices. Figure 2.7 depicts a typical configuration of horseshoe vortices for a standard wing planform. This figure shows an unswept quarter-chord wing where the bound vortex coincides with the quarter-chord line of the panel. In a rigorous theoretical analysis, the vortex lattice panels are located on the mean camber surface of the wing [4]. The trailing vortices are aligned parallel to the vehicle axis and

extend downstream to infinity. The position of both the control point and the bound vortex will be determined in the forthcoming sections.

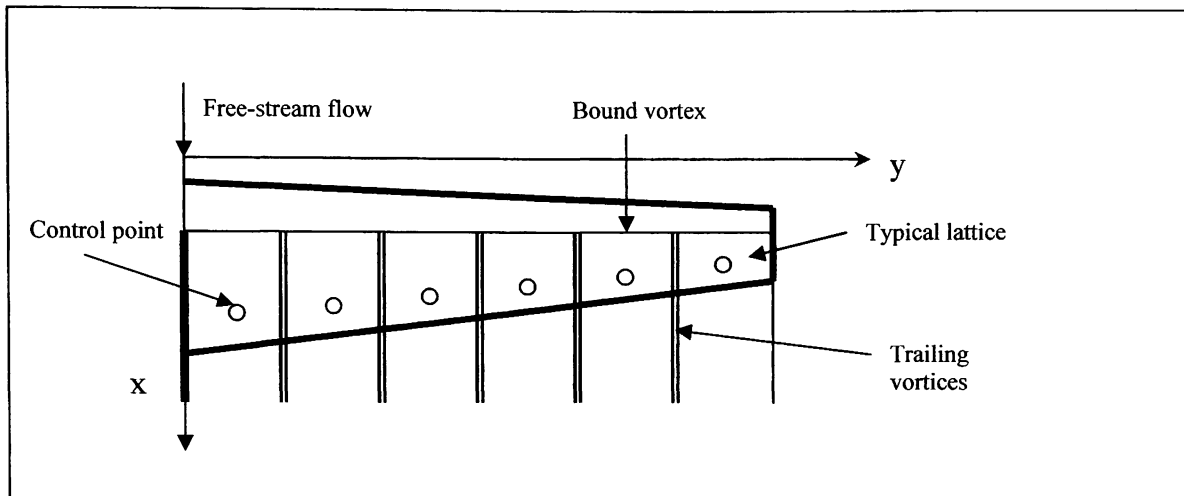


Figure 2.7: Typical lattice arrangement for a wing planform

2.2.1 Analysis and Implementation of the Horseshoe Vortex

The flow field induced by a horseshoe vortex is of great importance to the vortex lattice method. In order to describe mathematically this flow field, the use of the three vortex filament expressions are necessary to develop the main governing equations of the vortex lattice method. A horseshoe vortex consists of one finite length vortex and two semi-infinite vortices. This vortex system is illustrated in figure 2.8.

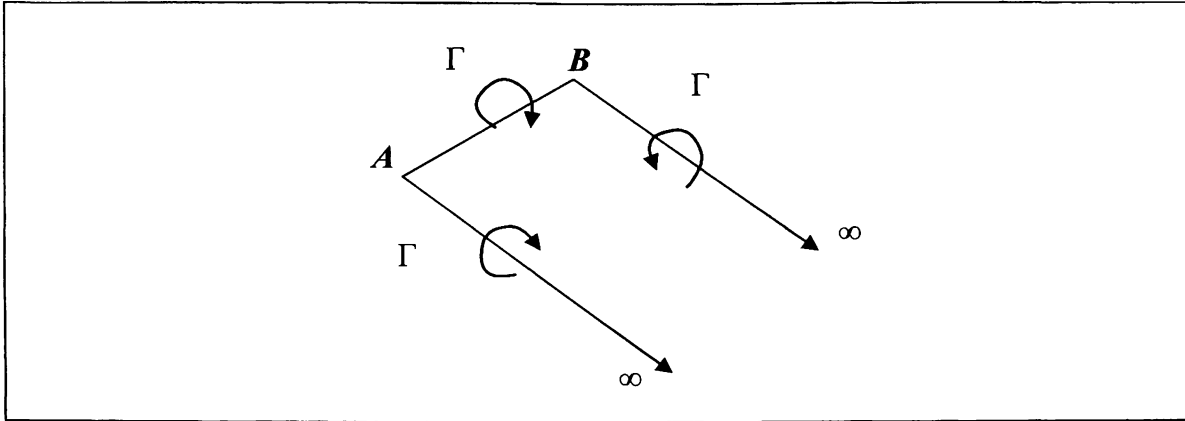


Figure 2.8: The horseshoe vortex

The horseshoe vortex and its flow field can be analyzed by looking into the effect of each individual vortex segment. The bound vortex described by segment AB represents the third vortex filament case analyzed earlier in section 2.1.3. Figure 2.9 will be used to describe the effect of the bound vortex AB on a point C in space whose normal distance from the bound vortex AB is r_p . According to section 2.1.3 and the finite vortex filament case, the magnitude of the velocity induced by the bound vortex AB of strength Γ_n on point C is:

$$V = \frac{\Gamma_n}{4\pi r_p} \int_{\theta_1}^{\theta_2} \sin \theta d\theta = \frac{\Gamma_n}{4\pi r_p} (\cos \theta_1 - \cos \theta_2) \quad [2.12]$$

The solution of equation 2.12 calls for the relation between the angles of the bound vortex filament and the vector definitions. The vectors in figure 2.8 are defined as:

$$\begin{aligned}\vec{r}_0 &= \overline{AB} \\ \vec{r}_1 &= \overline{AC} \\ \vec{r}_2 &= \overline{BC}\end{aligned}$$

The definitions of the dot product as well as the expression for the area of a parallelogram serve as tools to derive an expression relating the three main vectors and the angles that bound the vortex filament.

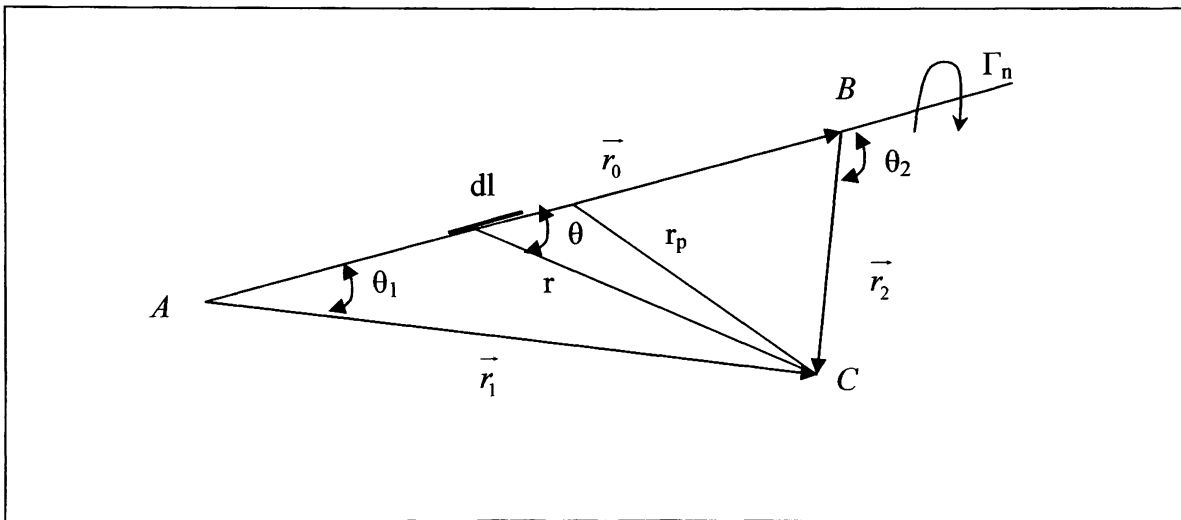


Figure 2.9: New nomenclature for the bound vortex filament

Hence, the designated vectors and angles of the bound vortex filament are expressed in the following way:

$$r_p = \frac{|\vec{r}_1 \times \vec{r}_2|}{r_0} \quad [2.13]$$

$$\cos \theta_1 = \frac{\vec{r}_0 \cdot \vec{r}_1}{r_0 r_1} \quad [2.14]$$

$$\cos \theta_2 = \frac{\vec{r}_0 \cdot \vec{r}_2}{r_0 r_2} \quad [2.15]$$

Expressions 2.13 through 2.15 are substituted in equation 2.12 with the appropriate vector identities to determine the magnitude of the induced velocity by the bound vortex at point C . The substitution yields

$$\vec{V} = \frac{\Gamma_n}{4\pi} \frac{\vec{r}_1 \times \vec{r}_2}{|\vec{r}_1 \times \vec{r}_2|^2} \left[\vec{r}_0 \cdot \left(\frac{\vec{r}_1}{r_1} - \frac{\vec{r}_2}{r_2} \right) \right] \quad [2.16]$$

Equation 2.16 is the general expression for the calculation of the induced velocity for a finite length vortex segment. In addition, the horseshoe vortex is made up by the summation of the finite length vortex segment and two trailing vortices that extend to infinity. A general expression is required for the velocity induced at a point in space (x,y,z) by a horseshoe vortex. The derivation of this general expression is divided in three main parts that go in accordance with each vortex segment. Figure 2.10 illustrates the case for the horseshoe vortex with a general point in space with three spatial coordinates.

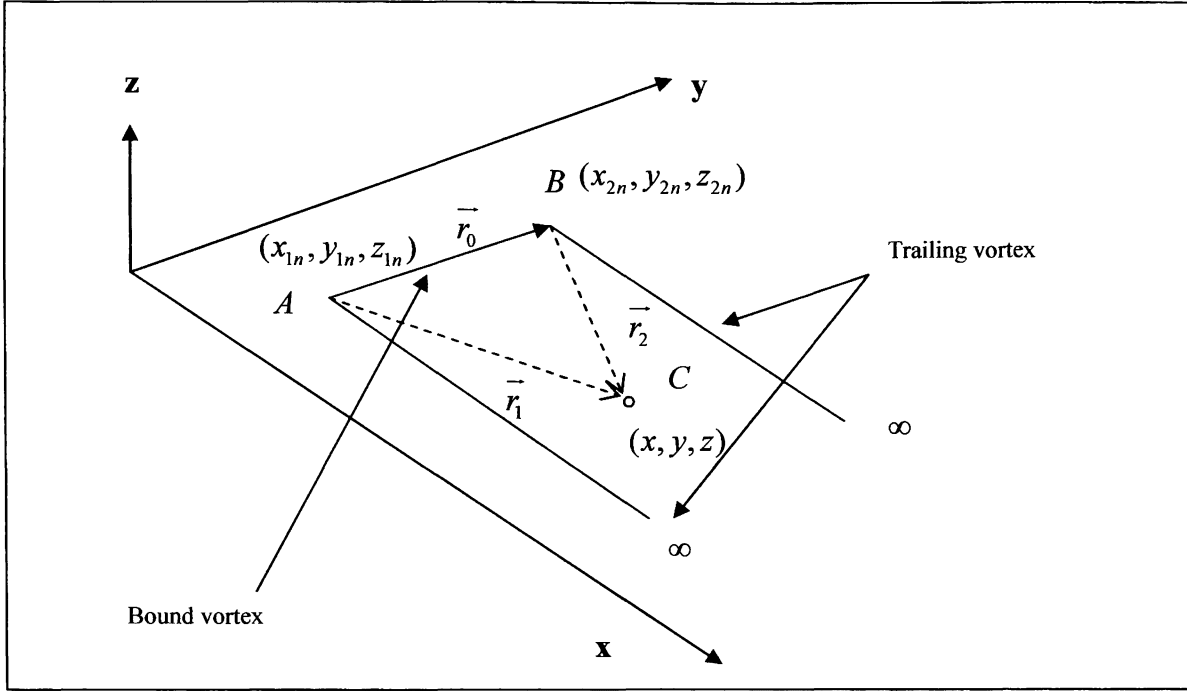


Figure 2.10: The horseshoe vortex implementation

The vector definitions needed for the calculation of the velocity induced by the bound vortex segment \overline{AB} at point C are given by the following expressions:

$$\begin{aligned}
 \vec{r}_0 &= \overline{AB} = (x_{2n} - x_{1n})\hat{i} + (y_{2n} - y_{1n})\hat{j} + (z_{2n} - z_{1n})\hat{k} \\
 \vec{r}_1 &= (x - x_{1n})\hat{i} + (y - y_{1n})\hat{j} + (z - z_{1n})\hat{k} \\
 \vec{r}_2 &= (x - x_{2n})\hat{i} + (y - y_{2n})\hat{j} + (z - z_{2n})\hat{k}
 \end{aligned}
 \tag{2.17}$$

The vector definitions above are substituted in equation 2.16 to find the general expression for the induced velocity calculation at point C by the bounded vortex filament \overline{AB} . The final expression yields

$$V_{AB} = \frac{\Gamma_n}{4\pi} (\omega_{AB})(\psi_{AB})
 \tag{2.18}$$

Where,

$$\omega_{AB} = \frac{\vec{r}_1 \times \vec{r}_2}{|\vec{r}_1 \times \vec{r}_2|^2}$$

$$= \frac{\left\{ \begin{aligned} &[(y-y_{1n})(z-z_{2n})-(y-y_{2n})(z-z_{1n})]\hat{i} - [(x-x_{1n})(z-z_{2n})-(x-x_{2n})(z-z_{1n})]\hat{j} \\ &+ [(x-x_{1n})(y-y_{2n})-(x-x_{2n})(y-y_{1n})]\hat{k} \end{aligned} \right\}}{\left\{ \begin{aligned} &[(y-y_{1n})(z-z_{2n})-(y-y_{2n})(z-z_{1n})]^2 + [(x-x_{1n})(z-z_{2n})-(x-x_{2n})(z-z_{1n})]^2 \\ &+ [(x-x_{1n})(y-y_{2n})-(x-x_{2n})(y-y_{1n})]^2 \end{aligned} \right\}}$$

And,

$$\psi_{AB} = \vec{r}_0 \cdot \frac{\vec{r}_1}{r_1} - \vec{r}_0 \cdot \frac{\vec{r}_2}{r_2}$$

$$= \left\{ \begin{aligned} &\frac{[(x_{2n}-x_{1n})(x-x_{1n})+(y_{2n}-y_{1n})(y-y_{1n})+(z_{2n}-z_{1n})(z-z_{1n})]}{\sqrt{(x-x_{1n})^2+(y-y_{1n})^2+(z-z_{1n})^2}} \\ &\frac{[(x_{2n}-x_{1n})(x-x_{2n})+(y_{2n}-y_{1n})(y-y_{2n})+(z_{2n}-z_{1n})(z-z_{2n})]}{\sqrt{(x-x_{2n})^2+(y-y_{2n})^2+(z-z_{2n})^2}} \end{aligned} \right\}$$

The velocity induced by the trailing vortices can be calculated using the semi infinite vortex case analysis. This vortex case calls for a new vector definition that recognizes a third point on the vortex filament that extends to infinity represented by point D . The

new vector definitions will keep the same numbering notation used earlier for the explanation of the bound vortex. Figure 2.11 illustrates the new vector expression.

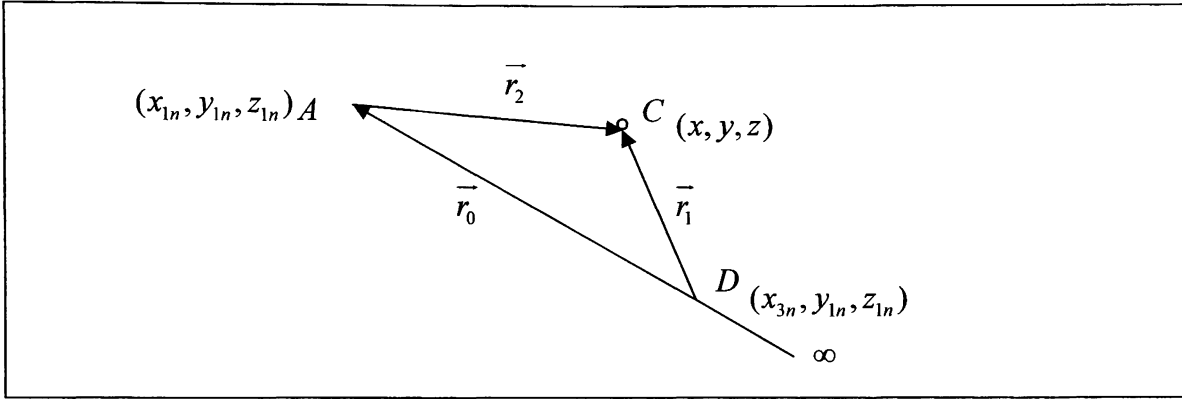


Figure 2.11: Trailing vortex nomenclature

$$\begin{aligned}
 \vec{r}_0 &= \overline{DA} = (x_{1n} - x_{3n})\hat{i} \\
 \vec{r}_1 &= (x - x_{3n})\hat{i} + (y - y_{1n})\hat{j} + (z - z_{1n})\hat{k} \\
 \vec{r}_2 &= (x - x_{1n})\hat{i} + (y - y_{1n})\hat{j} + (z - z_{1n})\hat{k}
 \end{aligned}
 \tag{2.19}$$

The derivation for the velocity induced at point C by the trailing vortex is the same as for the bound vortex. The only difference in this case is perceived when x_3 goes to infinity. Taking this into consideration, the contribution of the trailing vortex is expressed as:

$$\vec{V}_{A\infty} = \frac{\Gamma_n}{4\pi} \left\{ \frac{(z - z_{1n})\hat{j} + (y_{1n} - y)\hat{k}}{[(z - z_{1n})^2 + (y_{1n} - y)^2]} \right\} \left[1.0 + \frac{x - x_{1n}}{\sqrt{(x - x_{1n})^2 + (y - y_{1n})^2 + (z - z_{1n})^2}} \right]
 \tag{2.20}$$

And,

$$\overline{V}_{B\infty} = -\frac{\Gamma_n}{4\pi} \left\{ \frac{(z-z_{2n})\hat{j} + (y_{2n}-y)\hat{k}}{\left[(z-z_{2n})^2 + (y_{2n}-y)^2 \right]} \right\} \left[1.0 + \frac{x-x_{2n}}{\sqrt{(x-x_{2n})^2 + (y-y_{2n})^2 + (z-z_{2n})^2}} \right] \quad [2.21]$$

In general, the total velocity induced by a horseshoe vortex at a point in space C is given by the summation of the contribution of the bound vortex and the two trailing vortices. In fact, the general expression for this velocity is given by:

$$\overline{V} = \overline{V}_{AB} + \overline{V}_{A\infty} + \overline{V}_{B\infty} \quad [2.22]$$

Since Γ_n is contained linearly in each expression, the equations representing each vortex filament (2.18, 2.20, and 2.21) can be expressed in a much simpler form. Allowing point C to be the control point of the m th panel designated by the coordinates (x_m, y_m, z_m) , the new expression becomes:

$$\overline{V}_{m,n} = \overline{C}_{m,n} \Gamma_n \quad [2.23]$$

Equation 2.23 is the velocity induced at the m th control point by the horseshoe vortex representing the n th panel. The influence coefficient $\overline{C}_{m,n}$ depends on the geometry of the n th horseshoe vortex and its distance from the control point of the m th panel [4]. In

order to find the total induced velocity at the m th control point induced by the $2N$ vortices, equation 2.23 is expressed as

$$\vec{V}_m = \sum_{n=1}^{2N} \vec{C}_{m,n} \Gamma_n \quad [2.24]$$

Each control point lies within a horseshoe vortex representing a surface element. Hence, a lifting surface such as a wing is represented by a combination of these surface elements. The location of the horseshoe vortex and its control point is determined by a mathematical analysis described in the following section. According to [3], and [4], tradition has been to determine their locations by comparisons with known results. Their placement has become a rule of thumb in numerical panel methods.

2.2.2 Location of the Control Point and Bound Vortex

The location of both the control point and the bound vortex is determined not by a theoretical law, but instead by a placement that works well in accordance to theory. According to [4], the control point of each panel is centered spanwise on the three-quarter chord line midway between the trailing vortex legs. Figure 2.12 illustrates the placement of the control point and the bound vortex.

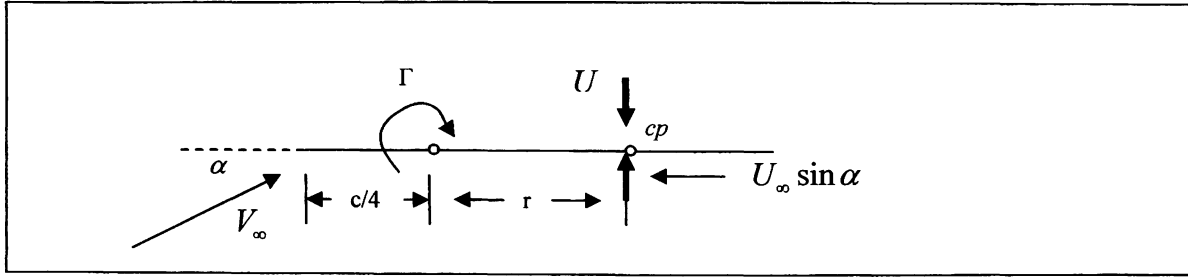


Figure 2.12: Control Point and Bound Vortex placement

The vortex filament of strength Γ positioned at the quarter chord location, induces a velocity at the control point cp given by

$$V = \frac{\Gamma}{2\pi r}$$

This result agrees with the infinite vortex filament case described earlier in sub-section 2.1.3.1. According to [4], if the flow is to be parallel to the surface at the control point, the incidence of the surface relative to the free stream can be expressed as:

$$\alpha \approx \sin \alpha = \frac{U}{V_\infty} = \frac{\Gamma}{2\pi r V_\infty} \quad [2.25]$$

In order to solve for the unknown distance r , equation 2.25 calls for combined use of the Kutta-Joukowski theorem and the results from thin airfoil theory. The combination of both relations gives the following result

$$l = \frac{1}{2} \rho_\infty V_\infty^2 c 2\pi \alpha = \rho_\infty V_\infty \Gamma \quad [2.26]$$

Equation 2.25 is then substituted in equation 2.26; as a result, the unknown distance r can be solved as a function of the chord length.

$$\frac{1}{2}\rho_{\infty}V_{\infty}^2c2\pi\left(\frac{\Gamma}{2\pi rV_{\infty}}\right)=\rho_{\infty}V_{\infty}\Gamma$$

$$r=\frac{c}{2} \quad [2.27]$$

Thus, the control point is located at the three quarter-chord location, and the bound vortex is located at the quarter-chord location. Both, the position of the bound vortex and the control point are functions of the chord length and the panel geometry. Chapter 3 will explain in detail the derivation of the geometric functions that will determine the position of the bound vortex and the location of the control point. These functions will depend on the geometric parameterization of the wing under the optimization process.

2.2.3 Analysis and Application of the Boundary Conditions

The solution of the induced velocity at a control point in space by a horseshoe vortex is possible through the application of the boundary conditions. The vortex strength Γ_n in equation 2.24 represents the lifting flow field of the wing. In order to solve for this flow field, the surface is considered a streamline. The resultant flow is tangent to the wing at each and every control point. As a result, the component of the induced velocity normal to the wing at the control point balances the normal component of the free-stream velocity. The tangency condition yields the following relation

$$-u_m \sin \delta_m \cos \phi - v_m \cos \delta_m \sin \phi + w_m \cos \phi \cos \delta_m + V_\infty \sin(\alpha - \delta_m) \cos \phi = 0 \quad [2.28]$$

where δ_m is the slope of the mean camber line at the control point and ϕ is the dihedral angle of the wing. Equation 2.28 can be simplified according to the shape of the airfoil section and the slope of the mean camber line. The tangency condition will give the solution for the system of simultaneous equations represented by equation 2.24. The unknown vortex strengths of each surface elements or panels are found through this solution. The calculation of all the pertinent aerodynamic coefficients and the solution to equation 2.24 will be exposed later in chapter 3.

2.3 Aerodynamic Performance and Analysis

So far, the previous sections explained in detail the theory behind the aerodynamic analysis used in this work. In summary, the optimization analysis, uses modern panel methods such as vortex lattice to provide the means of calculating numerically the inviscid aerodynamic properties of a wing. The subsequent sections will introduce the theory behind airplane performance, followed by a complete analysis of the main aerodynamic coefficients with emphasis on their impact on the performance index function. In fact, one of the main goals of the target function is the optimization of the high lift-to-drag ratio (L/D) at cruise velocity. This parameter which depends on the drag polar is a fundamental measure for airplane performance and aerodynamic efficiency.

2.3.1 Introduction to the Drag Components of a Wing

It is important to start this discussion with the analysis of drag since it mandates how efficient a wing can generate lift. There are only two sources of aerodynamic force on a body moving through a fluid. The first one is the pressure distribution, and the second the shear stress distribution, both acting over the body surface. As a result, all the different types of drag can be catalogued as either pressure drag or friction drag. The analytical prediction of drag is very complex, and its prediction is beyond the capability of current numerical aerodynamic models. Furthermore drag is affected by Mach number, and it is different around Mach number regimes. This study will focus only in the subsonic prediction of drag for a finite wing, and to further illustrate its calculation, the two-dimensional airfoil case serves as an introduction.

The total drag found on an airfoil is called the profile drag. Profile drag is a combination of skin friction drag and pressure drag. The first type of drag is due to frictional shear stresses and the latter due to an imbalance on the pressure distribution over the airfoil surface. In coefficient form, profile drag can be expressed as

$$c_d = c_f + c_{d,p} \quad [2.29]$$

Where c_d is the profile drag coefficient, c_f is the skin friction drag coefficient, and $c_{d,p}$ is the form drag coefficient. There are no exact analytical solutions that calculate or predict the form drag coefficient. Similarly, the calculation of the skin friction drag coefficient lacks the ability to present an exact solution. In this analysis, an empirical method is used to calculate the profile drag coefficient. The vortex lattice analysis is used to calculate

the section lift coefficient of the wing assuming a constant airfoil section. This allows an approximation of the profile drag coefficient for each panel using a two-dimensional lookup table of the drag polar diagram of the selected airfoil. The drag polar diagram is taken from reference [6] (appendix A3). This procedure will be explained in depth on chapter 3, and it will show how the optimization technique uses the result from the vortex lattice analysis to calculate the profile drag coefficient of the entire wing.

The discussion of drag in this work is limited to a finite wing. As a result, the zero-lift drag component consists only of the profile drag for the complete wing. In contrast, the drag due to lift consists of the change in profile drag when the wing is at an angle of attack different from the zero-lift angle, plus the vortex induced drag due to wingtip vortices.

2.3.2 The Drag Polar

According to the author in [5], for every aerodynamic body, there is a relation between the drag coefficient C_D and the lift coefficient C_L that can be expressed as an equation or plotted on a graph. Both the equation and the graph are called the drag polar. All the pertinent aerodynamic information concerning the wing and its performance can be collected through the use of the drag polar. The variation of the lift coefficient C_L with angle of attack vs. the variation of the total drag coefficient C_D with angle of attack constitutes the drag polar diagram. This diagram is illustrated in figure 2.13., and the equation particular to this parabolic curve is given as follows:

$$C_D = C_{D,0} + KC_L^2 \quad [2.30]$$

Where C_D is the total drag coefficient, $C_{D,0}$ is the zero-lift drag coefficient and KC_L^2 is the drag due to lift.

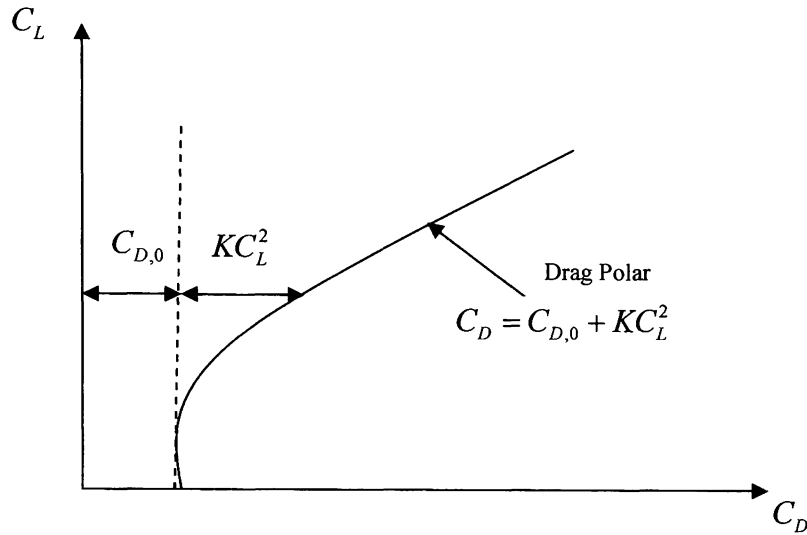


Figure 2.13: The Drag Polar Diagram

The optimization technique in this work uses the drag polar diagram to determine the wing's aerodynamic performance through an iterative procedure. It is important to note that the drag polar curve is sensitive to velocity variations, mainly due to the fact that C_L and C_D are functions of the Mach number. The exact details of this process will be given in Chapter 3. Nonetheless, the introduction of the drag polar is necessary for the development of the methodology within the program.

2.3.3 Equations of Motion for Steady, Level Flight

The performance analysis in this work assumes a straight and level flight condition. Since acceleration is of no concern, the four forces of flight balance in the following way.

$$T = D \quad [2.31]$$

$$L = W \quad [2.32]$$

Equations 2.31 and 2.32 show how the thrust is equal to drag, and the lift is equal to weight. It is important to note that in level flight, the climb angle θ and the roll angle ϕ are zero. These equations are the special cases of the general equations of motion, and in fact, represent the dynamic behavior of an airplane in steady, level flight.

2.3.4 Range

The derivation of the Breguet range equation serves as an introduction to the aerodynamic parameters that are optimized during the numerical optimization process. By definition, range is the total distance traversed by an airplane on one load of fuel. At any instant during the flight the weight of the airplane is equal to the empty weight plus the instantaneous weight of the fuel. The time rate of change of weight is given by the following equation:

$$\frac{dW}{dt} = \frac{dW_f}{dt} = \dot{W}_f \quad [2.33]$$

According to [5], range is intimately connected with engine performance through the specific fuel consumption. For example, the specific fuel consumption for a jet-propelled airplane defined in terms of available thrust T is given by:

$$c_t = -\frac{\dot{W}_f}{T} \quad [2.34]$$

For an airplane in steady, level flight, s denotes the horizontal distance covered over the ground. Assuming a stationary atmosphere, the airplane's velocity is given by [5]

$$V_{\infty} = \frac{ds}{dt}$$

or

$$ds = V_{\infty} dt = -\frac{V_{\infty}}{c_i T} dW_f \quad [2.35]$$

Using the relationships given in equation 2.33, equation 2.35 can be written as

$$ds = -\frac{V_{\infty}}{c_i} \frac{W}{T} \frac{dW}{W} \quad [2.36]$$

For a steady level flight condition, where $L=W$ and $T=D$, equation 2.36 becomes:

$$l = \rho_{\infty} V_{\infty} \Gamma_{\infty} \quad [2.37]$$

Integrating the instantaneous range with respect to the change in aircraft weight yields the Breguet range equation. This integration assumes that the velocity, specific fuel consumption, and L/D are approximately constant.[8]

$$R = \int_{W_1}^{W_0} \frac{V_{\infty}}{c_i} \frac{L}{D} \frac{dW}{W} = \frac{V_{\infty}}{c_i} \frac{L}{D} \ln \frac{W_0}{W_1} \quad [2.38]$$

The Breguet equation can be applied to jet aircraft as well as propeller driven airplanes. Since the focus of this study will aim at optimizing the range of a jet aircraft, equation 2.38 can be expressed in a different manner. The maximum range of a jet aircraft is not dictated by the maximum lift to drag ratio, instead it is driven by the maximum product of $V_\infty(L/D)$. For steady level flight, velocity is expressed as:

$$V_\infty = \sqrt{\frac{2W}{\rho_\infty S C_L}}$$

Hence,

$$V_\infty \frac{L}{D} = \sqrt{\frac{2W}{\rho_\infty S C_L}} \frac{C_L}{C_D} = \sqrt{\frac{2W}{\rho_\infty S C_L}} \frac{C_L^{1/2}}{C_D} \quad [2.39]$$

Substituting equation 2.39 into 2.38 gives the final expression for the range of a jet propelled airplane as follows:

$$R = \int_{W_1}^{W_0} \frac{1}{c_t} \sqrt{\frac{2W}{\rho_\infty S}} \frac{C_L^{1/2}}{C_D} \frac{dW}{W} \quad [2.40]$$

Assuming that specific fuel consumption, density, wing area and $C_L^{1/2}/C_D$ are constant, equation 2.40 can be simplified and expressed as:

$$R = \frac{2}{c_t} \sqrt{\frac{2}{\rho_\infty S}} \frac{C_L^{1/2}}{C_D} (W_0^{1/2} - W_1^{1/2}) \quad [2.41]$$

Theoretical range is one of the driving variables within the performance index function of the aerodynamic optimization tool. In order to maximize range, the analysis will focus on optimizing the product of $(V_\infty)(L/D)$ noting that this term does not vary with weight, and that the specific fuel consumption is held constant at the given altitude.

3. ANALYSIS

3.1 Introduction to the Computational Method

The optimization technique and the vortex lattice analysis in this work have been setup in a MATLAB code developed for the implementation of the numerical analysis. The code layout can be synthesized in the input geometry, the vortex lattice solver, and the performance index computation. The optimization technique is based on the initial wing geometry and operating flight conditions specified by the user at the beginning of the simulation or pre-processing condition. The pre-processing condition refers to the period before the numerical analysis and the vortex lattice analysis have taken place. The vortex lattice solver is the processor, and the performance index computation is post-processor. Each division of the optimization analysis follows a hierarchy mandated by the complexity of the simulation process. The processor occupies the highest level of importance within all three simulation levels. The outline of the computational process is given in figure 3.1. The following sections will explain in detail the three steps of the simulation which include the pre-processor, the processor and the post-processor analysis.

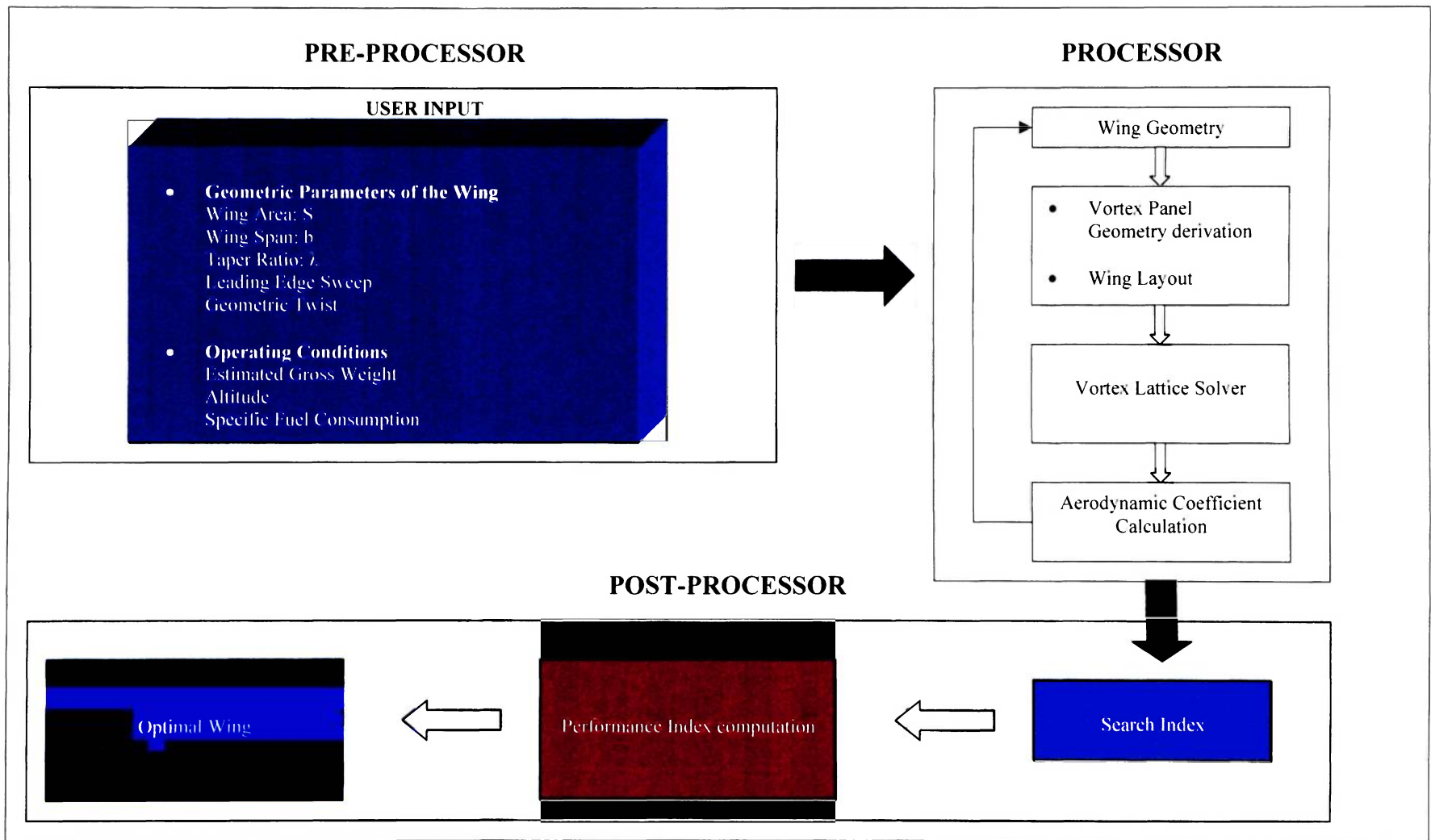


Figure 3. 1: Computational Analysis Flow Chart

3.2 Input Geometry and Operating Conditions

The initial geometric parameters of the wing as well as the operating conditions are user specific. The initial wing geometry is defined by the wing area, wingspan, taper ratio, leading edge sweep and geometric twist. These geometric parameters are specified by the user at the beginning of the simulation. The wing's section geometry remains constant throughout the numerical analysis, and it is selected at the beginning of the simulation with the initial geometric inputs. Consequently, the operating flight conditions such as altitude, gross weight, and specific fuel consumption are selected by the user during the pre-processor state. The subsequent sections will deal with the most important arguments of the numerical analysis at the pre-processing condition.

3.2.1 Wing Geometry

The optimization technique will focus on optimizing the wing area (S), taper ratio (λ), wingspan (b), leading edge sweep and geometric twist of a general wing with a constant airfoil section geometry. The optimal configuration of these parameters will satisfy a performance index function designed to maximize the high lift to drag ratio (L/D) at cruise velocity, while maximizing cruise range at this condition. In order to determine the wing's aerodynamic performance using the vortex lattice method, the configuration of each wing must be derived in terms of the geometric parameters mentioned above. The following geometric relations are used in the analysis of the output of the geometric configuration for each wing. These relations also serve as the foundation for the derivation of the horseshoe panel geometry as a function of the wing's geometric parameters.

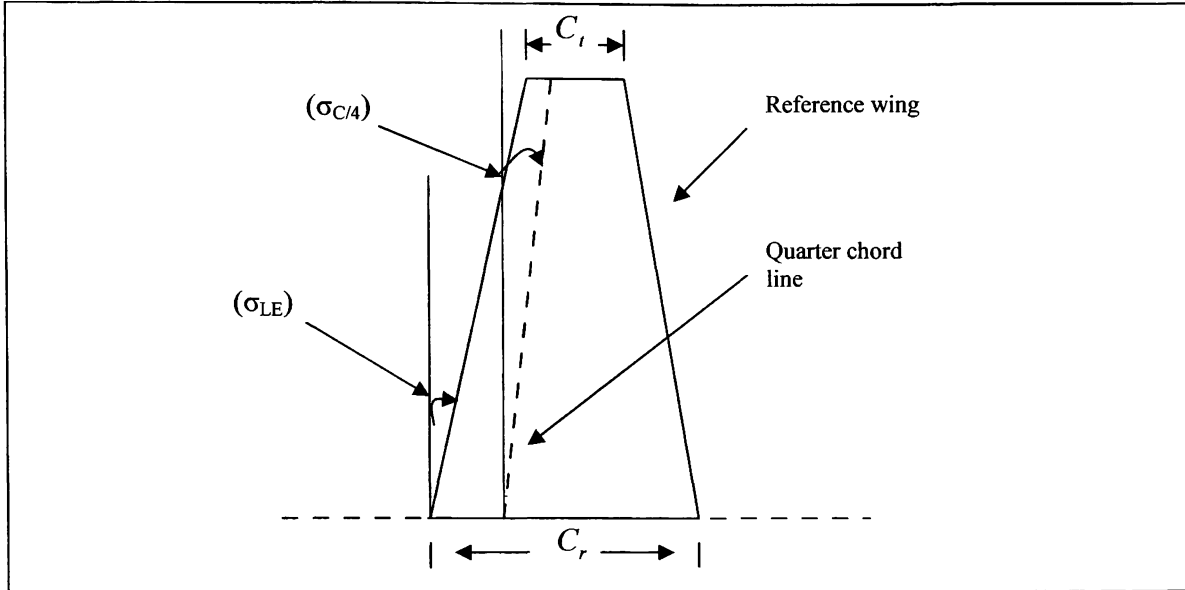


Figure 3.2: Wing Geometry

The general wing layout analyzed on this report pertains to the trapezoidal wing geometry. This wing configuration has several geometric relations that are expressed as functions of the reference parameters. One of the most important geometric relations is the aspect ratio, and it is given by the following expression:

$$AR = \frac{b^2}{S} \quad [3.1]$$

Where b is the span and S is the reference wing area. The next equation deals with the ratio between the root chord C_r and the tip chord C_t known as the taper ratio λ . This geometric relation is expressed as

$$\lambda = \frac{C_t}{C_r} \quad [3.2]$$

Furthermore, the reference wing area S , the wingspan b and the taper ratio λ are all related to the root chord C_r through the following expression.

$$C_r = \frac{2S}{[b(1+\lambda)]} \quad [3.3]$$

Additionally, there are two key angles that describe the wing geometry. The first angle is the leading edge sweep angle which is a function of the quarter chord sweep, taper ratio and aspect ratio given by the following equation:

$$\tan \sigma_{LE} = \tan \sigma_{c/4} + \left[\frac{(1-\lambda)}{AR(1+\lambda)} \right] \quad [3.4]$$

The last geometric parameter is the geometric twist angle of the wing. Geometric twist is the actual change in the airfoil's angle of incidence, usually measured with respect to the root airfoil. During the analysis, the incidence angle of the tip airfoil will rotate in a nose-down direction about the quarter chord of the wing.

These geometric relations will allow the program to loft the wings properly and derive the geometric layout of the horseshoe vortex with the initial input of wing area, taper ratio, wingspan, leading edge sweep, and geometric twist. The next section will introduce the wing section geometry and the selection process regarding the airfoil.

3.2.2 Wing Section Geometry

The airfoil is one of the most fundamental aspects of a wing. It defines the overall aerodynamic efficiency during all phases of flight, and has an effect in several performance parameters including takeoff, cruise and stalling speed.

It is of interest to note, that prior to 1930, the design of an airfoil had a customized and personalized approach with no consistency in its method. According to [2], after 1930, the National Advisory Committee for Aeronautics (NACA) embarked on a series of definitive airfoil experiments using airfoil shapes that were constructed rationally and systematically. As a result, the NACA nomenclature method uses the following definitions to describe the general airfoil geometric properties. The major design feature of an airfoil is the mean camber line, which is the locus of points halfway between the upper and lower surfaces, as measured perpendicular to the mean camber line itself. The chord of the airfoil is the distance of the straight line connecting the leading edge to the trailing edge. The camber is the maximum distance between the mean camber line and the chord line. The camber, the shape of the mean camber line, and the thickness distribution of the airfoil essentially control the lift and the moment characteristics of the airfoil [5]. Figure 3.3 illustrates the airfoil shape used in this study.

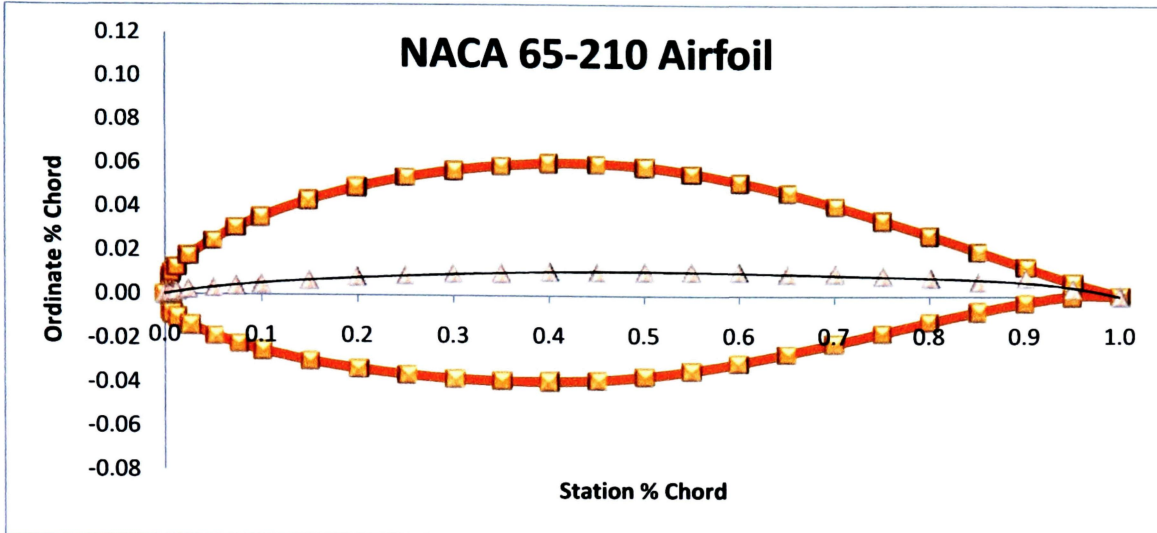


Figure 3.3: NACA 65-210 Airfoil

The NACA investigations were further systematized by separation of the effects of camber and thickness distribution, and the experimental work was performed at higher Reynolds numbers [6]. As a result, the NACA 6 series airfoil and its nomenclature are a product of the systematic approach to define the geometry of the wing section. The airfoil used in this study belongs to the NACA 6 series airfoil family. The 6 series airfoils were designed to encourage laminar flow and profile drag reduction. The exact airfoil used in this study is the NACA 65-210. According to the NACA nomenclature, the first digit represents the series designation. The second digit is the location of minimum pressure in tenths of the chord behind the leading edge for the basic symmetrical section at zero lift. The third digit following the dash, gives the design lift coefficient in tenths. The last two digits indicate the thickness of the wing section in percent of the chord.

The plot of the geometric layout of the NACA 65-210 airfoil used in this study was lofted using the stations and ordinates given in per cent of airfoil chord by reference [6].

The non-dimensional mean camber line equation is approximated by a sixth degree polynomial as follows:

$$z(x) = -0.7825x^6 + 2.2021x^5 - 2.3763x^4 + 1.2458x^3 - 0.3676x^2 + 0.0782x + 0.0001 \quad [3.5]$$

Equation 3.5 will be used by the simulation to determine the camber distribution along the chordwise direction

3.2.3 Operating Conditions

The operating conditions communicate to the processor or the vortex lattice solver the gross weight of the aircraft, the operating altitude and the thrust specific fuel consumption. Each wing configuration is trimmed at the required gross weight during the aerodynamic analysis to determine the lift variation and drag variation with angle of attack and airspeed. The thrust specific fuel consumption is held constant during the simulation at the altitude specified by the user.

This section completes the first part of the optimization technique. In summary, this part is the pre-processing condition characterized by the input of the user-specific geometric parameters of the wing, and the operating conditions. The subsequent sections will elaborate on the vortex lattice solver and the performance index computation.

3.3 Vortex Lattice Solver

The vortex lattice solver is the processor and the heart of the main optimization technique. In fact, it is where the main aerodynamic analysis takes place. The vortex

lattice solver will build a 5-dimensional array composed of the variation of the initial wing's planform geometry characterized by the wing area, wingspan, taper ratio, leading edge sweep and geometric twist. For each wing configuration, the simulation will loft the wing planform, derive the panel geometry, determine the location of the bound vortex and the control point, and calculate its aerodynamic performance. The vortex lattice analysis is divided into a logical operational sequence. The first part is characterized by the choice of a singularity element. The horseshoe vortex, in this case, the singularity element, will consist of a straight bound vortex segment that models the lifting properties and of two semi-infinite trailing vortex lines that model the wake. The second part is in charge of the derivation of the panel geometry and the grid generation. In this phase, the wing is divided into a certain number of spanwise and chordwise elements or panels. The third part relates to the calculation of the influence coefficients and the application of the boundary conditions. Finally the last part deals with the calculation of the pertinent aerodynamic coefficients necessary for the optimization analysis. The subsequent sections will explain in detail how the vortex lattice analysis is performed.

3.3.1 Derivation of the Panel Geometry

The multidimensional array contains the geometric properties of the wing such as the wing area, wingspan, taper ratio, leading edge sweep and geometric twist. These values are transferred to the vortex lattice solver for the derivation of the horseshoe panel geometry, and the control point coordinates.

Each panel in the vortex lattice analysis is basically a horseshoe vortex. Its geometry is directly altered by the geometric layout of the wing and the airfoil section geometry.

The geometric derivation of each panel will use the relations given earlier in section 3.2.1 and the aerodynamic coordinate system illustrated in figure 2.10.

The horseshoe vortex has three main points that characterize its geometric layout. The first point corresponds to the beginning of the bound vortex vector. Its location with respect to the aerodynamic coordinate system is indicated by (x_{1n}, y_{1n}, z_{1n}) . The second point corresponds to the end of the bound vortex vector, indicated by the coordinates (x_{2n}, y_{2n}, z_{2n}) . The coordinates of the last point represent the position of the control point with respect to the aerodynamic coordinate system indicated by (x_m, y_m, z_m) . Again, the bound portion of the horseshoe vortex coincides with the quarter chord line of its panel, and the trailing vortices are in the plane of the wing, parallel to the x axis. Figure 3.3 illustrates the concept.

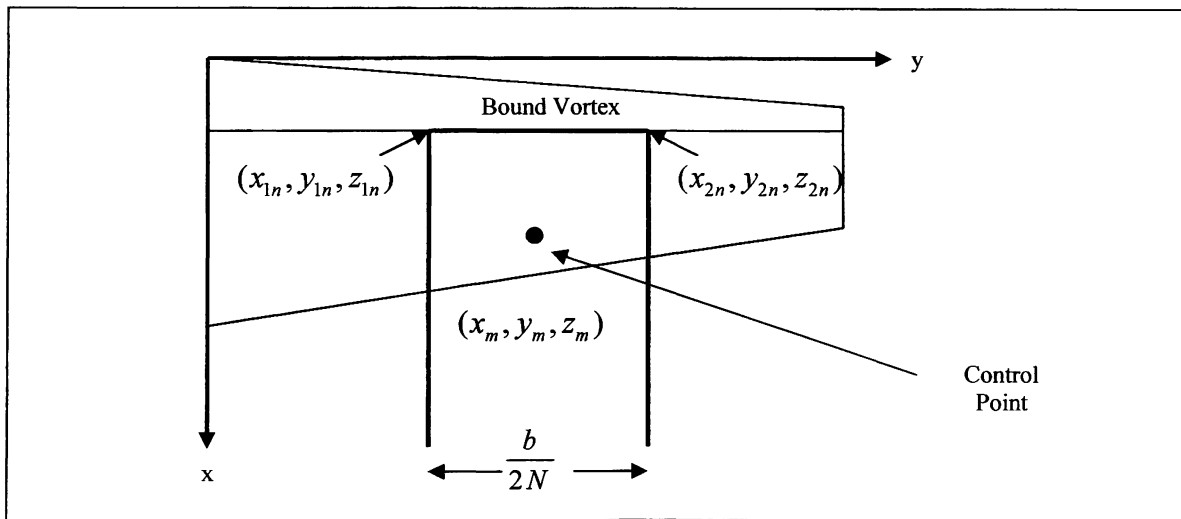


Figure 3.4: Panel representation of a wing with unswept quarter chord.

The derivation of the panel geometry is rather complex. The coordinates of the bound vortex depend on the distribution of camber in the chordwise direction as well as the linear variation of the angle of incidence along the span due to the geometric twist of the

tip airfoil. Consequently, the control point coordinates are also a function of the geometric properties of the wing, and are subject to camber variation and washout.

The application of the boundary conditions and the analysis that allows for the solution of the vortex strengths at each control point, are explained in the next section.

3.3.2 Solution of the Vortex Strengths

The definition of the bound vortex coordinates and the control point coordinates allow the planform of the wing to be divided into an $N \times P$ lattice (N spanwise divisions and P chordwise divisions). The lattice layout depends on the input of the initial geometric parameters such as the wing area S , the taper ratio λ , the wingspan b , the leading edge sweep, and the geometric twist angle. Once the vortex lattice geometry is setup, the analysis will solve for the total induced velocity at each control point. Thus for a symmetric flow, equation 2.24 becomes

$$\vec{V}_m = \left(\sum_{n=1}^N \vec{C}_{m,n} \Gamma_n \right)_S + \left(\sum_{n=1}^N \vec{C}_{m,n} \Gamma_n \right)_P \quad [3.6]$$

Where the symbols s and p represent the starboard and port wings, respectively. The solution of equation 3.17 is only possible through the application of the boundary conditions approximating the surface as a streamline. The tangency requirement states that if the flow is tangent to the wing, the component of the induced velocity normal to the wing at the control point balances the normal component of the free-stream velocity. In fact, this condition is necessary for the solution of the vortex strengths. As a result, the

simulation determines the unit normal vector and the velocity vector for each individual panel.

According to [4], the unknown circulation strengths (Γ_n) required to satisfy these tangent flow boundary conditions are determined by solving the system of simultaneous equations represented by equation 3.17. The solution involves the inversion of a matrix.

Since the flow field under consideration is symmetric with respect to the xz plane, the lift force acting at a point on the starboard wing (+y) is equal to that at the corresponding point on the port wing (-y). Because of the symmetry condition, the solution only requires to solve for the vortex strengths of the starboard wing. However, the contributions of the horseshoe vortices of the port wing at the control points of the starboard wing have to be included when solving the equations.

3.3.3 Calculation of the Aerodynamic Coefficients

Once the vortex strengths have been determined by satisfying the boundary conditions assuming that the flow is tangent to the surface at each of the control points, the first aerodynamic coefficient may be calculated.

According to the author in [4], for wings that have no dihedral over any portion of the wing, all the lift is generated by the free-stream velocity crossing the spanwise vortex filament. The lift per unit span is calculated based on the Kutta-Joukowski theorem as:

$$l = \rho_{\infty} V_{\infty} \Gamma_n \quad [3.7]$$

Since the flow is symmetric, the total lift for the wing in terms of the finite element panels is

$$L = 2\rho_{\infty}V_{\infty} \sum_{n=1}^N \Gamma_n \Delta y_n \quad [3.8]$$

where

$$\Delta y_n = \frac{b}{2N} \quad [3.9]$$

The lift coefficient, which is one of the most important aerodynamic coefficients needed for the optimization analysis, is calculated as follows

$$C_L = \frac{L}{q_{\infty}S} \quad [3.10]$$

Equation 3.24 will allow the calculation of the total lift coefficient needed for the drag polar analysis of the wing. The second coefficient pertinent to this analysis is the total drag coefficient. However, before the total drag calculation is introduced and explained, the calculation of the section lift coefficient is required to proceed further in the analysis.

Since the lift per unit span is given by equation 3.7, the section lift coefficient is calculated as follows

$$C_l = \frac{l}{\frac{1}{2}\rho_{\infty}V_{\infty}^2 c_{avg}} = \frac{2\Gamma}{U_{\infty}c_{avg}} \quad [3.11]$$

Where c_{avg} is the average chord of each panel, and can be expressed in terms of the geometric parameters of the wing and the panel number as follows:

$$c_{avg} = \frac{2}{b} \left[\left(\tan \sigma_{LE} \right) \left(\frac{b}{2} \right) + \frac{2S(\lambda-1)}{b(1+\lambda)} \right] \frac{(2n-1)b}{4N} + \left(\frac{2S}{b(1+\lambda)} \right) \left(\frac{b}{2} \right) - \left[\left(\tan \sigma_{LE} \right) \left(\frac{(2n-1)b}{4N} \right) \right] \quad [3.12]$$

The section lift coefficient of an individual panel is given by equation 3.11. Since this analysis is interested on a more general expression for the section lift distribution of the entire wing, equation 3.11 can be rewritten as.

$$C_l = \sum_{n=1}^N \frac{2\Gamma_n}{V_\infty c_{avg}} \quad [3.13]$$

In addition, the section lift coefficient is used in the calculation of the profile drag coefficient of the wing through the use the airfoil's drag polar diagram.

3.3.3.1 Computation of the Vortex Induced Drag and the Profile Drag

As mentioned earlier in section 2.3.1, the total drag of the wing is dictated by equation 2.30, where the total drag coefficient is a function of the profile drag and the drag due to lift. In this analysis, the lift-dependent drag is essentially vortex induced drag. The simulation will calculate the vortex induced drag coefficient using the following equation:

$$C_{D_i} = \frac{C_L^2}{\pi e AR} \quad [3.14]$$

Where e is the span efficiency factor, usually defined in terms of the induced drag factor δ as follows:

$$e = \frac{1}{(1 + \delta)} \quad [3.15]$$

The simulation will use a two dimensional lookup table provide by reference [12] to calculate the induced drag factor δ , which is a function of aspect ratio and taper ratio. As soon as the induced drag factor is determined, the aero analysis will determine span efficiency using equation 3.15, and the vortex induced drag coefficient using equation 3.14.

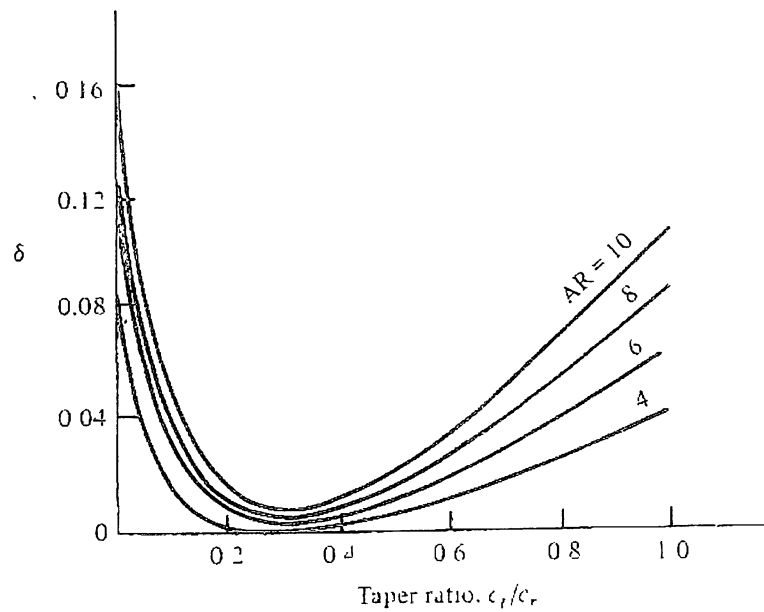


Figure 3.5: Induced drag factor δ as a function of taper ratio

The profile drag coefficient calculation involves a different process. Its calculation requires the use of empirical data which is accessed through the optimization technique using a two-dimensional lookup table. This lookup table is composed of the drag polar diagram of the selected airfoil. For this analysis, the drag polar diagram of the NACA 65-210 airfoil is used to build up the two dimensional lookup table. The vortex lattice analysis will output the search index, which is made up by the section lift coefficient, and the Reynolds number calculated at the average chord of each chordwise wing station.

The process is simple. The section lift coefficient is calculated for each individual panel through the vortex lattice routine. Furthermore, the average chord is calculated at each chordwise strip with the purpose of estimating the Reynolds number at that location. Since the taper of the wing varies, the average chord varies as a function of the span. As soon as the search index is built for a panel, the lookup table will output the section drag coefficient at each strip location according to the values calculated by the VLM routine of the section lift coefficient and Reynolds number. The process is illustrated in figure 3.6.

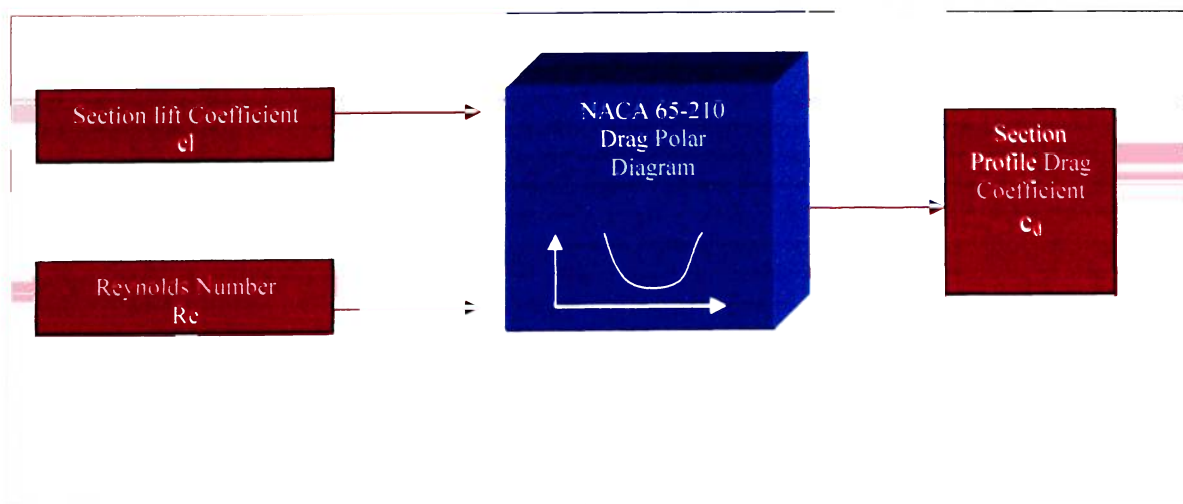


Figure 3.6: Profile Drag Calculation

The Reynolds number for each individual panel is calculated at the altitude specified by the user and by the average chord length at each spanwise location. The drag polar diagram of the NACA 65-210 airfoil is taken from reference [6] given in Appendix A.3. Its data is tabulated in the code as a two-dimensional array. The section drag coefficient is the output of the two-dimensional lookup table. This output is needed for the calculation of the total profile drag and its coefficient. The following equation is used to determine the value of the profile drag for the entire wing:

$$D_{profile} = 2 \sum_{n=1}^N \frac{1}{2} \rho V_{\infty}^2 C_{d_n} A_{p_n} \quad [3.17]$$

Where C_{dn} is the profile drag coefficient and A_p is the trapezoidal chordwise strip area. The computational routine calculates the individual areas according to the wing's geometric parameters. The process is similar to the one outlined in section 3.3.1 which deals with the derivation of the panel geometry. The total profile drag coefficient of the wing is calculated through the following equation:

$$C_d = \frac{D_{profile}}{\frac{1}{2} \rho V_{\infty}^2 S} \quad [3.18]$$

The computational routine will then calculate the minimum profile drag coefficient C_{do} for each airspeed variation based on the output by equation 3.18. As a result, the total drag coefficient needed for the wing's drag polar analysis can be calculated as follows:

$$C_D = C_{do} + C_{di} \quad [3.19]$$

Equation 3.19 completes the aerodynamic analysis and the vortex lattice solver routine. With the calculation of the total lift coefficient C_L and the total drag coefficient C_D , the multidimensional array which represents the drag polar for all the different wing configurations can be filled accordingly.

3.4 Fuel Volume and Fuel Weight Estimation

Fuel weight and the fuel volume are calculated by the simulation assuming a tapered fuel tank that spans along the wing and has a rectangular base defined by the airfoil section. The base of tapered fuel tank will start at the quarter chord and will end at the 70% chord station as illustrated in figure 3.7.

The fuel volume is calculated by integrating the cross sectional area of the tank defined by the following equation:

$$A(x) = \left[\left(\frac{a-d}{b/2} \right) x + d \right] \left[\left(\frac{(C)a - (C)d}{b/2} \right) x + (C)d \right] \quad [3.20]$$

Where $A(x)$ is expressed in terms of the geometric properties of the tank as follows: a is the product of the wing's tip chord and the airfoil thickness at the 70% chord station and b is the wingspan. Consequently, d is the product of the wing's root chord and the thickness of the airfoil at the 70% root chord station. C is a constant defined by the ratio

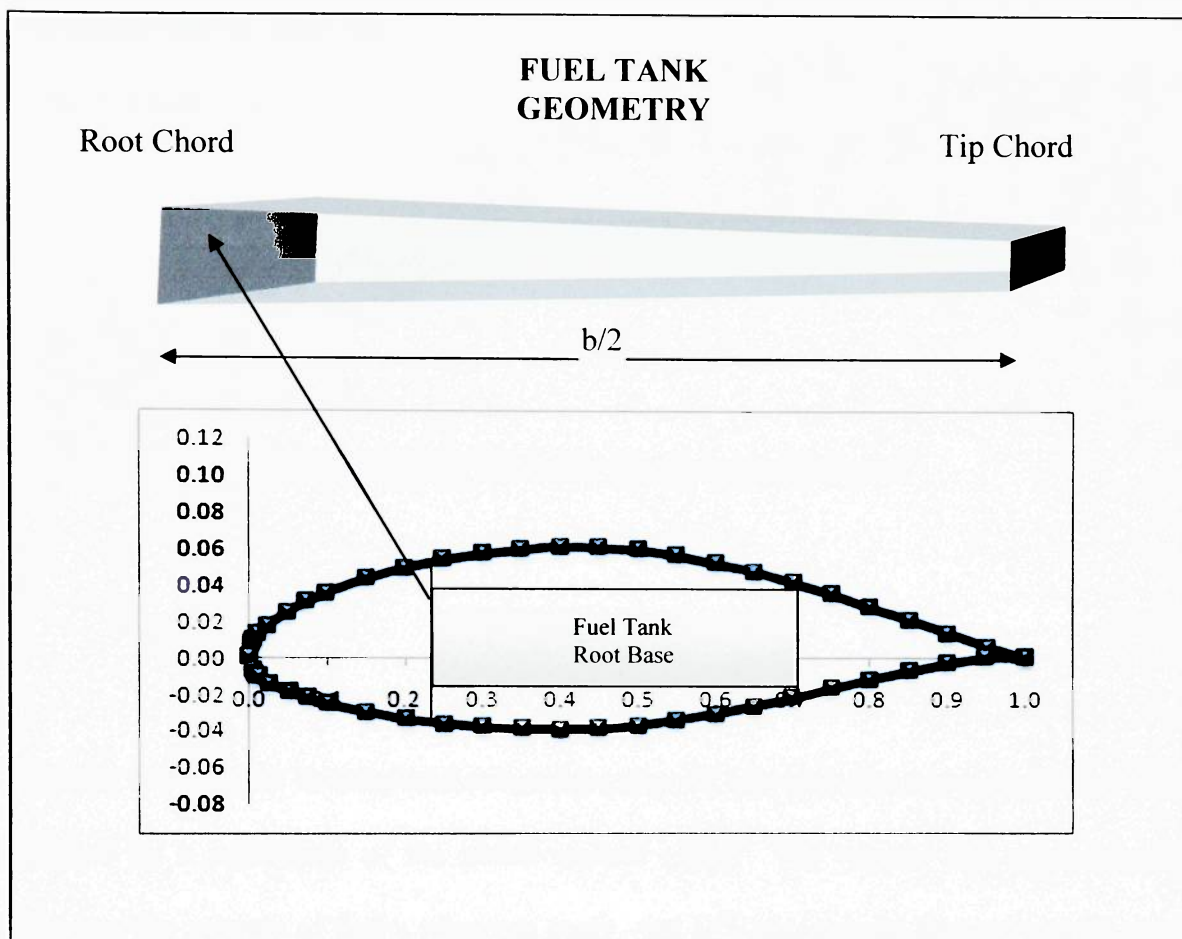


Figure 3.7: Fuel Tank Description

of the length of the base of the fuel tank and the thickness of the airfoil at the 70% root chord station.

As a result, the volume of the fuel tank can be integrated along the span of the wing to generate the following expression:

$$V_{\text{tank}} = \int_0^{b/2} A(x) dx = C \left(\frac{b/2}{3} \right) (a^2 + ad + d^2) \quad [3.21]$$

Assuming that the wing has two identical fuel tanks, the total fuel volume carried by the wing in *gallons* can be determined through the following equation:

$$V_{\text{fuel}} = 2(V_{\text{tank}}) \left(\frac{7.48 \text{gallons}}{1 \text{ft}^3} \right) \quad [3.22]$$

As a result, the total fuel weight in lbs can be determined as follows:

$$W_{\text{fuel}} = V_{\text{fuel}} (\rho_{\text{fuel}} g) \quad [3.23]$$

Overall, fuel volume is calculated assuming a tapered fuel tank that has a chordwise base dictated by a percentage of the airfoil section chord. The routine will automatically calculate the amount of fuel a wing can carry, and will estimate the fuel weight for each wing configuration. This allows a more logical approach for fuel weight estimation.

3.5 Wing Weight Estimation

The wing weight of each individual wing configuration is estimated by the simulation through the statistical group weights method described by Raymer in reference [8]. This method applies statistical equations based upon sophisticated regression analysis. Usually, these equations typify those used in conceptual design by the major airframe companies, and cover fighter/attack, transport, and general aviation aircraft. This analysis will only focus on the weight equation of the wing for a general aviation aircraft, given by the following expression:

$$W_{wing} = 0.036 S_w^{0.758} W_{fw}^{0.0035} \left(\frac{A}{\cos^2 \Lambda} \right)^{0.6} q^{0.006} \lambda^{0.04} \left(\frac{100t/c}{\cos \Lambda} \right)^{-0.3} (N_z W_{dg})^{0.49} \quad [3.24]$$

Where

S_w = trapezoidal wing area, ft²

W_{fw} = weight of fuel in wing, lb

A = aspect ratio

Λ = wing sweep at 25% MAC

q = dynamic pressure at cruise, lb/ft²

λ = taper ratio

N_z = ultimate load factor; = 1.5 × limit load factor

W_{dg} = flight design gross weight, lb

The application of this equation is subject to the following consideration. The limit load factor estimation comes from the recommendation given by reference [8]. For a general aviation aircraft of normal category, the typical limit load factor varies between 2.5 and 3.8. In chapter 4, all the details and values concerning the terminology of equation 3.7 will be explained.

3.6 Performance Computation

The performance analysis represents the post-processor routine of the optimization technique. The simulation will determine the aerodynamic performance of each wing configuration by calculating the C_L vs. alpha curve, the lift to drag ratio curve, and the drag polar curve with airspeed variation. The vortex lattice analysis acts as a virtual wind tunnel testing all possible wing configurations and calculating all the pertinent aerodynamic data needed for the optimization analysis.

The performance analysis adds two new dimensions to the previous 5-dimensional geometric array due to virtual wind tunnel analysis. These two new dimensions correspond to the angle of attack and airspeed variation. The simulation will trim each wing for a steady level flight condition during the analysis using a multivariable search index. The search index is composed of the following key variables: wing area S , wingspan b , taper ratio λ , leading edge sweep, geometric twist angle, airspeed V_∞ , and the lift coefficient required to trim the aircraft at the gross weight specified by the user given by:

$$C_{LW} = \frac{2W}{\rho V_\infty^2 S} \quad [3.25]$$

Since the C_L vs. alpha curve is calculated for every single wing configuration, the angle of attack corresponding to the value of the lift coefficient calculated by equation 3.24 can be retrieved using the information given by the search index. This process is explained in figure 3.8:

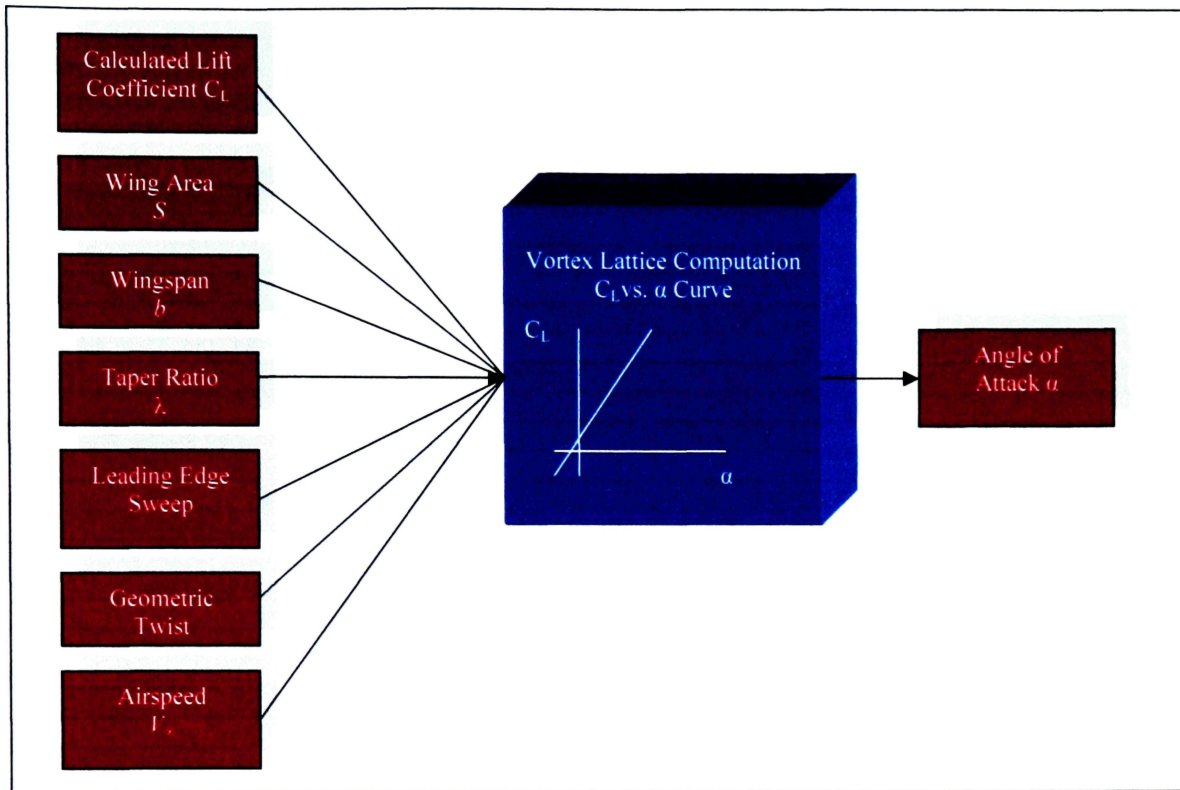


Figure 3. 8: Computational Routine for the Performance Analysis of the Wing. Part 1

The computational routine will take the output value of the angle of attack and use it to obtain the corresponding drag coefficient for that flight condition. In fact, the computational process responsible for obtaining the corresponding drag coefficient employs a similar method as the one illustrated in figure 3.8.

The total drag coefficient will have a search index of 7 dimensions. Six out of the seven dimensions will remain the same when compared to the previous search index. The new dimension in this case corresponds to the angle of attack, which is the output of the C_L vs. α curve from the previous computational routine. The search index retrieves the drag coefficient from the drag polar diagram of that particular flight condition. Figure 3.9 illustrates the lookup table concept.

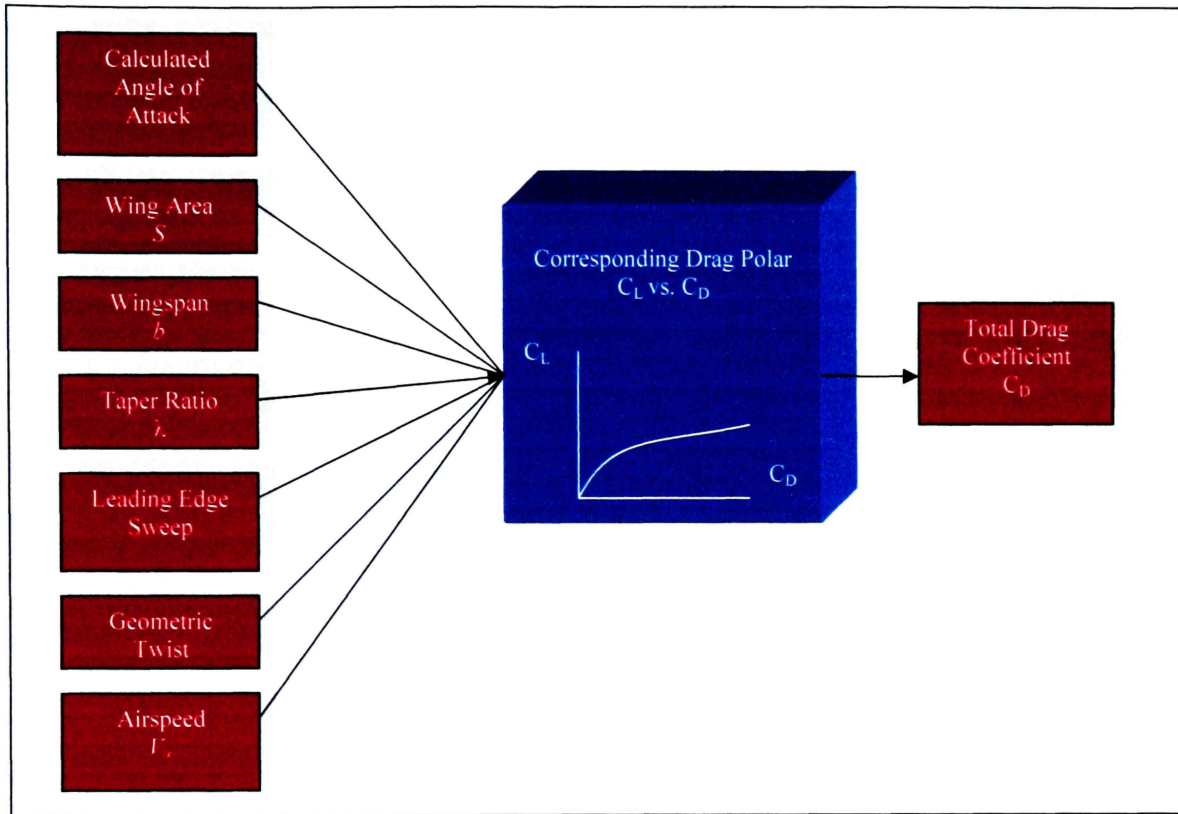


Figure 3. 9: Computational Routine for the Performance Analysis of the Wing. Part 2

The code will run and do the necessary loops until it retrieves the corresponding lift and drag coefficients. As soon as the aerodynamic coefficient analysis is done, the simulation will calculate the lift to drag ratio curve and the drag variation with velocity with the gathered aerodynamic data for each wing configuration. As mentioned in section 2.3.4, the variation of drag with velocity for a given airplane at a given altitude is nothing more than the thrust required. As a result the drag variation of the wing with velocity is calculated using the following expression:

$$T_R = D = \frac{1}{2} \rho_{\infty} V_{\infty}^2 S C_D \quad [3.26]$$

Each wing configuration will have its own thrust required curve mandated by the performance routine mentioned earlier. The actual plot of the thrust required curve for the optimal wing can be seen in Chapter 4 along with several other aerodynamic performance plots.

3.6.1 Performance Index Function

The performance index function is the main component of the optimization analysis. It acts as the search engine of the simulation, looking for the best possible wing configuration that ultimately results in the highest lift-to-drag ratio L/D at cruise velocity, maximum range and minimum wing weight. In order for the simulation to find the optimal wing, the performance index function is calculated for each wing configuration through the following expression:

$$P_{I(S,b,\lambda,\sigma Le,GT)} = (w_1) \left(\frac{\left(\frac{L}{D} \right)_{cruise(S,b,\lambda,\sigma Le,GT)}}{\left(\frac{L}{D} \right)_{cruise_{initial}}} \right) + (w_2) \left(\frac{W_{wing_{initial}}}{W_{wing(S,b,\lambda,\sigma Le,GT)}} \right) + (w_3) \left(\frac{R_{(S,b,\lambda,\sigma Le,GT)}}{R_{initial}} \right) \quad [3.27]$$

The performance index function has three main variables with a corresponding function weight (w). The first variable is the lift to drag ratio at cruise velocity and represents the aerodynamic efficiency of the wing. The second variable is wing weight which is calculated based on the statistical regression equation described earlier in section 3.5. The third variable is the theoretical maximum range which is calculated using the Breguet formulation as described in section 2.3.4. The performance index function given by

equation 3.27 allows the simulation to determine the best possible combination of lift to drag ratio at cruise velocity, minimal wing weight, and maximum range. Each wing configuration will have a calculated value of the performance index function. These values are stored in an array of five dimensions. These 5 dimensions correspond to the geometric parameters of the wing described by wing area, wingspan, taper ratio, leading edge sweep and geometric twist angle.

The computational routine will target the wing that has the highest value of the performance index function. In order to identify the geometric parameters of the wing pertaining to the highest value of the performance index function, the routine develops a search index that looks where this value is located within the five-dimensional performance array. Once this index has been determined, the simulation can extract other performance parameters such as cruise velocity, angle of attack at cruise, fuel volume and fuel weight among others.

In summary, the computational process of the optimization technique begins with the input geometry and passing conditions, followed by the vortex lattice solver, and finalizes with the performance index computation. The input geometry and passing conditions symbolize the pre-processing state of the computational routine. The vortex lattice solver denotes the main processing stage, and the performance index computation denotes the post-processing routine.

3.7 Initial Wing Properties and Program Initialization

The vortex lattice optimization program was written to be user specific. The initial wing geometry serves as the starting point of the numerical analysis. To initialize the simulation, a standard trapezoidal wing geometry is selected with a constant NACA 65-210 airfoil section. A total of 24 spanwise and 8 chordwise panels are selected providing an adequate division to correlate aerodynamic theory and model correctly the camber distribution of the wing in the chordwise direction. This provides an even division of panels, allowing the starboard wing and the port wing to have a total number of 96 panels each.

The optimization analysis will test 324 wing combinations. Each combination represents a variation of the 5 parameters that dictate the geometry of the trapezoidal wing. Table 3.1 illustrates the geometric test matrix, the range and the variation of each geometric parameter.

Table 3.1: Geometric Variation: Optimization Analysis

Geometric Parameter	Range	Variation Δ	Combination
Wing Area: S	$\{S S-2 \leq S \leq S+2\}$	$\Delta S = 2\text{ft}^2$	3
Wing Span: b	$\{b b-1 \leq b \leq b+1\}$	$\Delta b = 1\text{ft}$	3
Taper Ratio: λ	$\{\lambda \lambda-0.1 \leq \lambda \leq \lambda+0.1\}$	$\Delta\lambda = 0.1$	3
Leading Edge Sweep: σ_{LE}	$\{\sigma_{LE} 2.64^\circ \leq \sigma_{LE} \leq 8.64^\circ\}$	$\Delta \sigma_{LE} = 2.0^\circ$	4
Geometric Twist Angle: G.T	$\{G.T 0^\circ \leq G.T \leq 2^\circ\}$	$\Delta G.T = 1.0^\circ$	3
Total Wing Combinations:	$3 \times 3 \times 3 \times 4 \times 3 = 324$ combinations		

The initial test wing will have a span of 37 ft, a wing area of 153 ft², a taper ratio of 0.5 a leading edge sweep of 8.64 degrees, and a geometric twist angle of 2 degrees. The simulation will run at standard sea level, at a maximum assumed gross weight of 5200lbs and a thrust specific fuel consumption of 0.6. The performance index function will be weighed the following way: 60% for aerodynamic efficiency, 20% for wing weight efficiency and 20% for range efficiency. Figure 3.10 presents the typical initialization windows required by the program to start the simulation using the wing geometry input and the initial flight conditions.

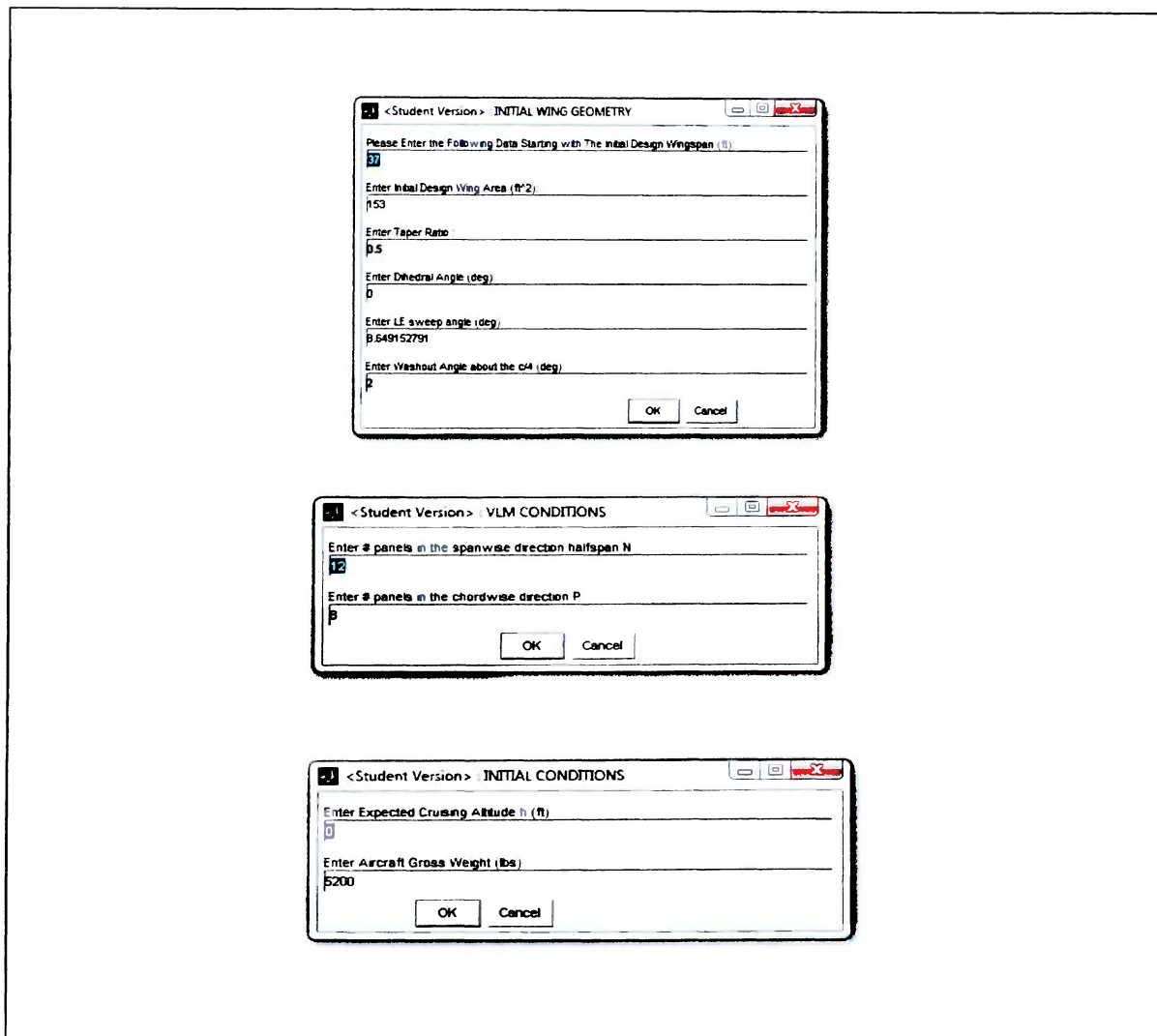


Figure 3.10: Program Initialization Windows

4. RESULTS

4.1 Optimal Wing Geometric Configuration

The results of the optimization process call for a breakdown of the geometric parameters and the actual geometric layout of the optimal wing. The results for both wings are compared and presented for clear visualization in changes regarding the wing geometry and aerodynamic properties.

The geometrical layout of the optimal wing configuration is the corresponding output of the optimal index defined by the 5 geometrical parameters described earlier in section 3.4.1. Table 4.1 presents the results for the geometric analysis of the optimal wing configuration and compares them to the initial wing parameters.

Table 4.1: Geometric analysis

Wing Reference Geometry					
Parameters	Wing Area (S) ft²	Wing Span (b) ft	Taper Ratio λ	L.E Sweep (deg)	Geometric Twist (deg)
Initial Wing	153	37	0.5	8.64°	2°
Optimal Wing	151	36	0.4	4.64°	2°
Percent Difference %	1.30%	2.7%	20%	46.3%	0%

According to table 4.1, the optimal wing experienced an overall change in planform geometry when compared to the initial wing design. There was a 1.30% decrement in wing area, a 2.7% decrement in wingspan, and a 20 % decrement in taper ratio. Consequently, there was a reduction in leading edge sweep of approximately 46%, and no change in the geometric twist angle resulting in the same configuration of washout as the optimal wing geometry. Overall, the optimal geometric configuration represents a wing that is optimized for maximum range, minimum wing weight and a higher lift to drag ratio at cruise condition. The preceding sections will explain in detail why this geometrical configuration serves as the optimal design.

4.1.1 Optimal Wing Versus Initial Wing Layout

The simulation creates the geometric layout of the optimal and the initial wing using the five geometric parameters given in table 4.1. The lofting procedure uses the relations provided in section 3.2.1 to generate the distribution of the panels along the chordwise and spanwise directions. Each vortex lattice panel has four defining points with three spatial coordinates that are function of wing area, taper ratio, wingspan, leading edge sweep, and geometric twist angle. The output of the panel geometry is given with respect to the aerodynamic coordinate system. The simulation creates an array for each wing configuration with the spatial coordinates of each defining point. The program creates the loft of the initial and the optimal wing using the information provided by this array. Figures 4.1 and 4.2 illustrate the geometric configuration of each wing and the vortex lattice arrangement. The scale in the z axis is set to amplify geometric twist and camber.

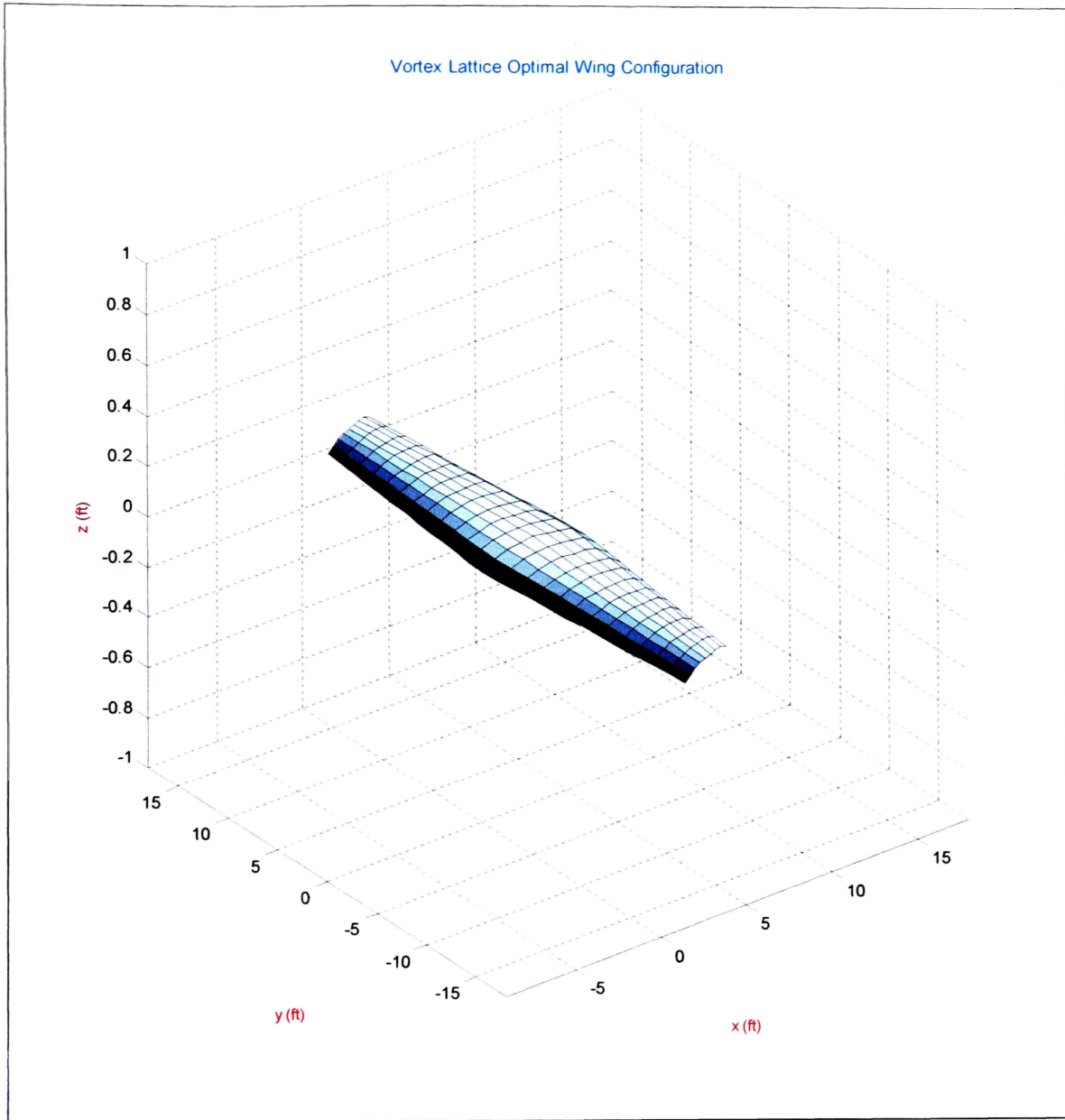


Figure 4. 1: Geometric layout of the optimal wing: Loft Analysis and Panel Configuration

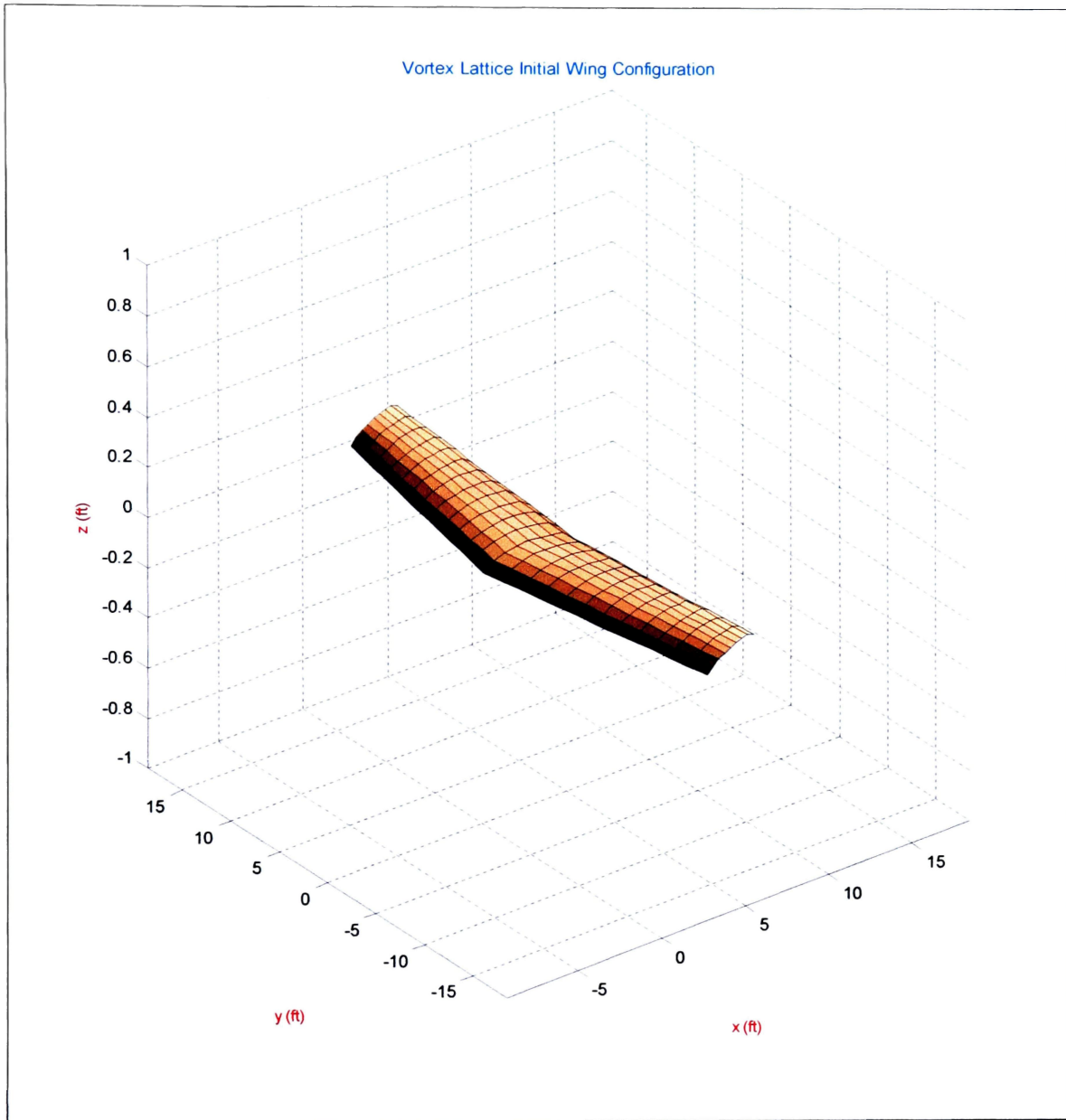


Figure 4. 2: Geometric layout of the Initial wing: Loft Analysis and Panel Configuration

4.2 Performance Index Function Analysis

The performance index function analysis shows how the different wing configurations perform in terms of the lift to drag ratio at cruise velocity, maximum theoretical range and minimum wing weight. The following figures illustrate the different surface plots that correspond to the three performance metrics (L/D, range, wing weight) of the optimization analysis as well as the performance index surface plot. The performance index function is scaled in terms of the initial wing. A value of 1 represents initial wing performance, anything higher than 1 outperforms the initial wing, anything lower than 1 is considered below the normal performance of the initial wing configuration. Three geometric variables corresponding to the optimal wing planform have been frozen for the purpose of illustrating these 3-dimensional surface plots. These geometric variables are taper ratio, leading edge sweep and geometric twist.

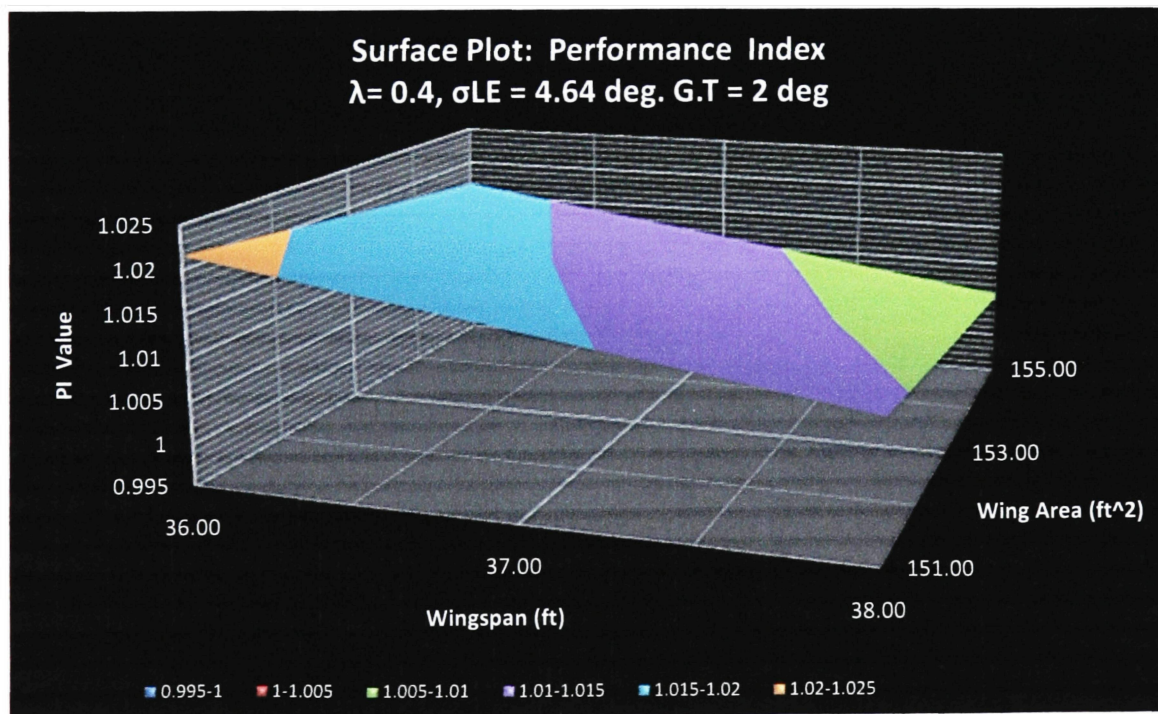


Figure 4.3: Performance Index Surface Plot: $\lambda = 0.4$, $\sigma_{LE} = 4.64$ deg, $G.T = 2$ deg

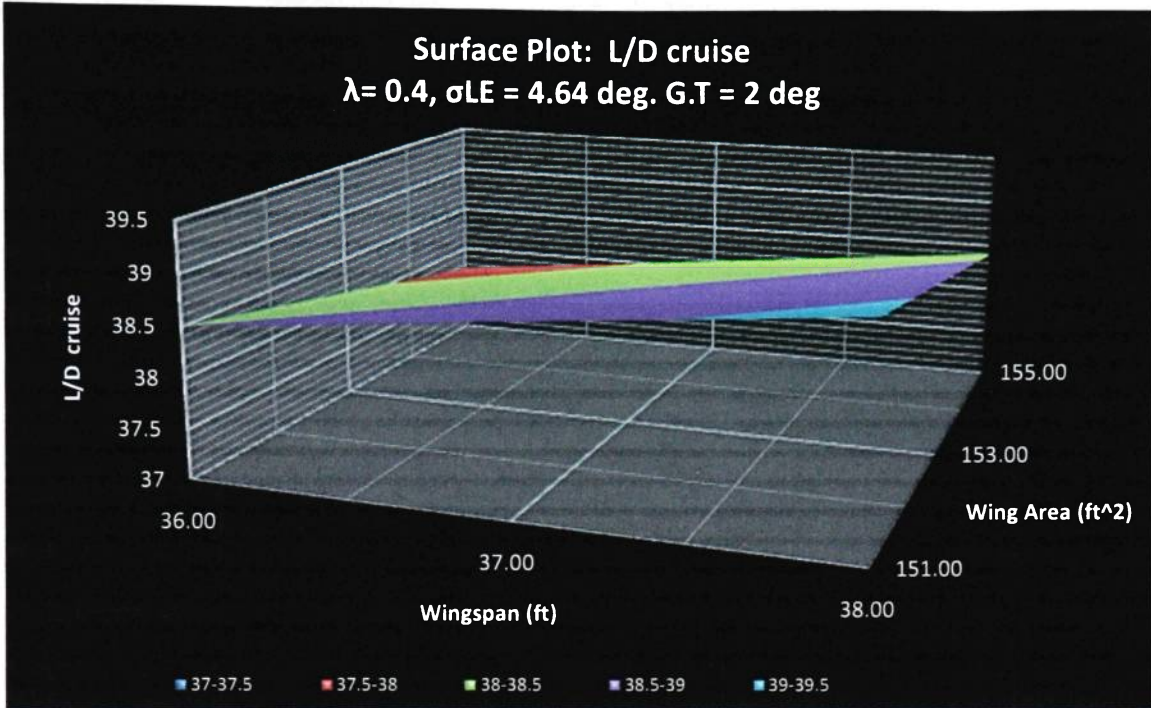


Figure 4.4: L/D Cruise Surface Plot: $\lambda = 0.4, \sigma_{LE} = 4.64 \text{ deg, G.T} = 2 \text{ deg}$

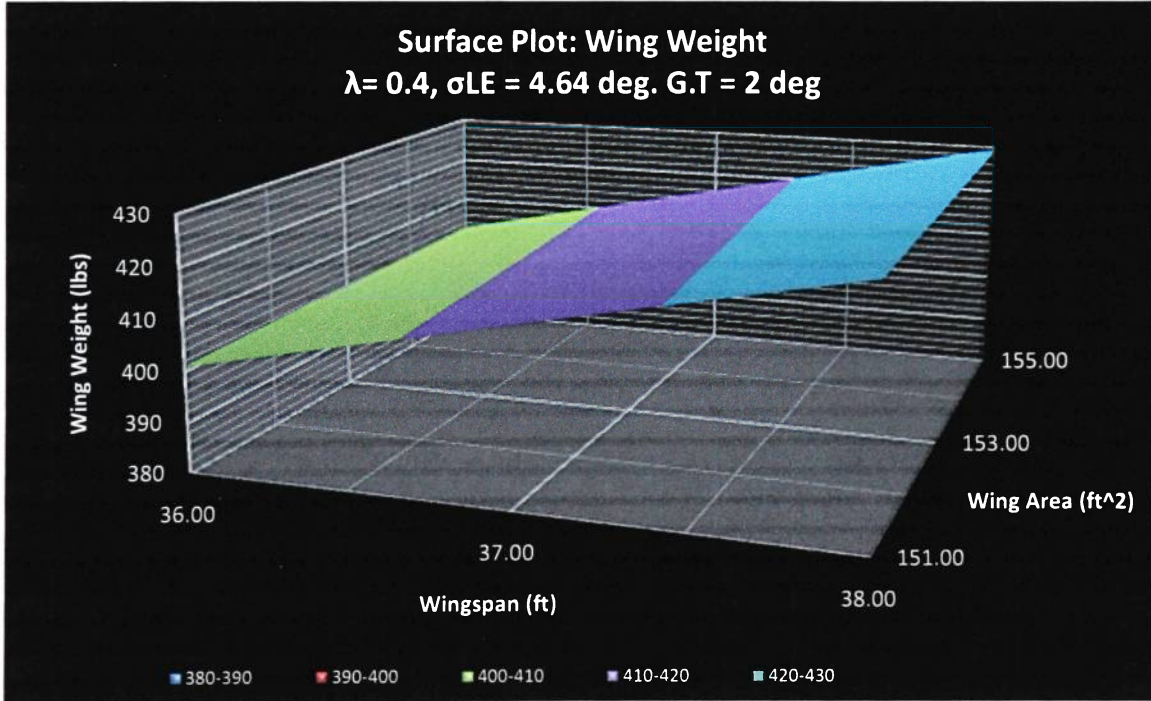


Figure 4.5: Wing Weight Surface Plot: $\lambda = 0.4, \sigma_{LE} = 4.64 \text{ deg, G.T} = 2 \text{ deg}$

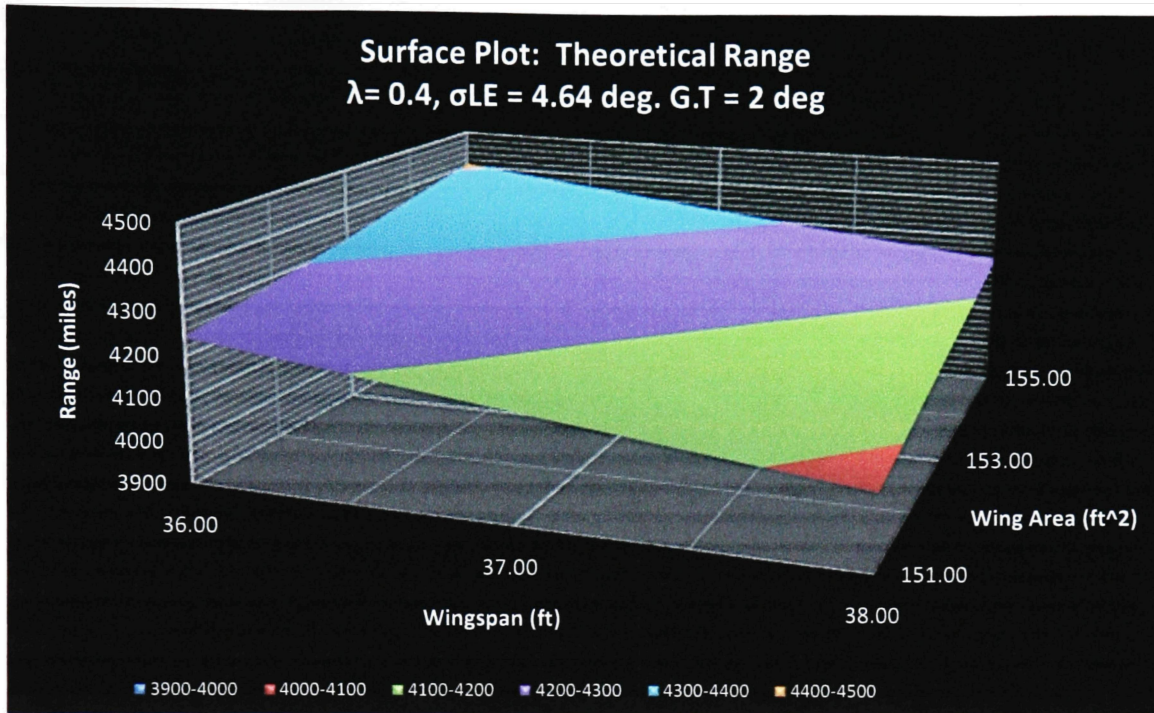


Figure 4.6: Theoretical Range Surface Plot: $\lambda = 0.4$, $\sigma_{LE} = 4.64$ deg, $G.T = 2$ deg

These surface plots show how the 5 geometric design variables (wingspan, wing area, taper ratio, leading edge sweep, and washout) affect the performance index function and each of the performance metrics. Each surface plot has three axes. The depth axis represents wing area, the horizontal axis represents wingspan, and the vertical axis represents either a performance metric or the performance index function value (PI).

Overall, the optimal wing configuration described in section 4.1 has the highest performance index function value. The subsequent sections will explain in detail why this wing configuration is deemed optimal when compared to the initial wing planform.

4.3 Aerodynamic Analysis

The optimal aerodynamic configuration of a wing that allows for maximum lift to drag ratio at cruise velocity is targeted by the simulation through the aerodynamic ratio of the performance index function. It is important to note that cruise velocity is determined when the value of $C_L^{0.5}/C_D$ reaches a maximum. As a result, the simulation performs an analysis on the variation of $C_L^{0.5}/C_D$ with velocity for each wing configuration. This aerodynamic ratio dictates the flight condition for maximum range and cruise velocity. Table 4.2 illustrates the results of the optimization analysis concerning the aerodynamic performance of each wing in terms of this ratio.

Table 4.2: Aerodynamic Ratio Analysis: Optimal Wing vs. Initial Wing

Aerodynamic Ratio Analysis		
Aerodynamic Parameters	$\left(\frac{C_L^{1/2}}{C_D}\right)_{\max}$	$V_{cruise} = V_{(C_L^{1/2}/C_D)_{\max}}$
Wing	90.40	400.64 ft/s
Initial Wing		
Optimal Wing	90.63	400.64 ft/s
Percent Increase %	0.25%	0.0%

The optimal wing configuration has the same cruise velocity of the initial wing and a slight increase of less than 1% in the maximum value of $C_L^{0.5}/C_D$.

Figure 4.7 illustrates the variation of $C_L^{0.5}/C_D$ with velocity for both wings. The optimal wing maintains similar if not equal aerodynamic ratio efficiency at higher velocities when compared to the initial wing design.

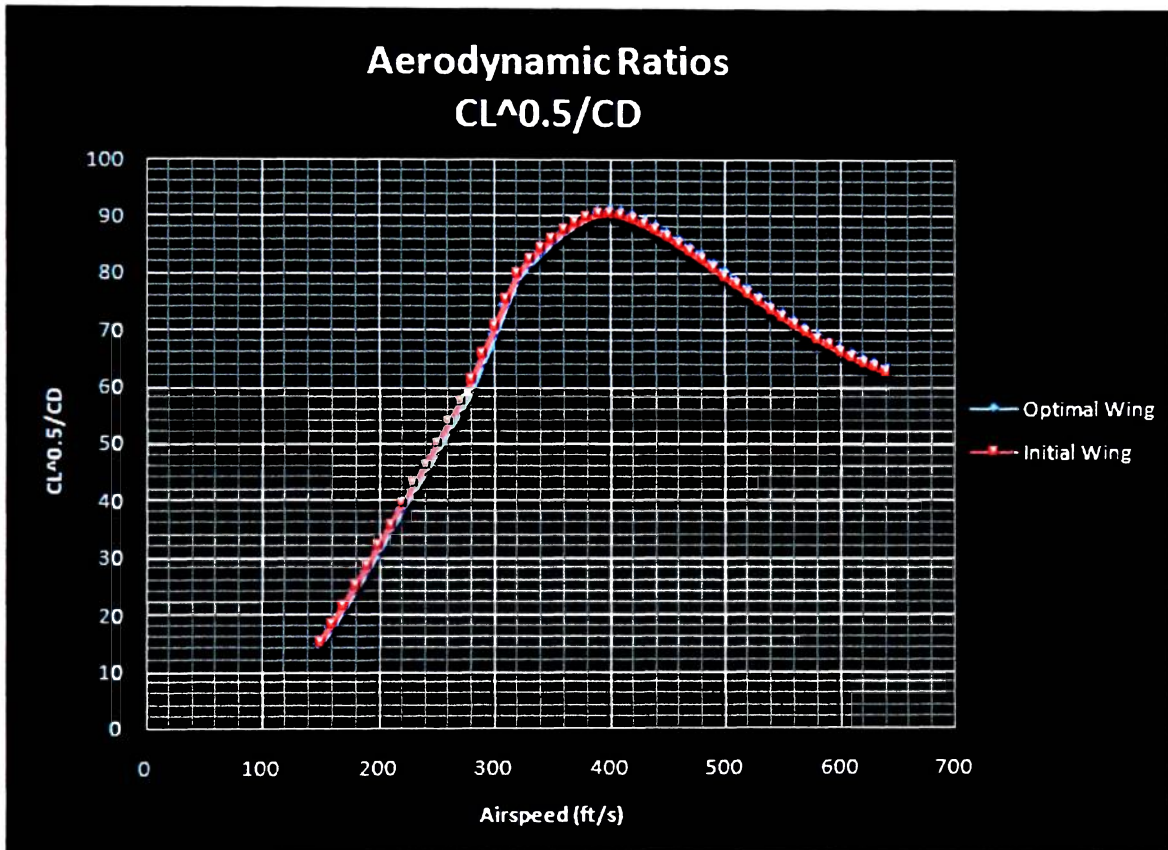
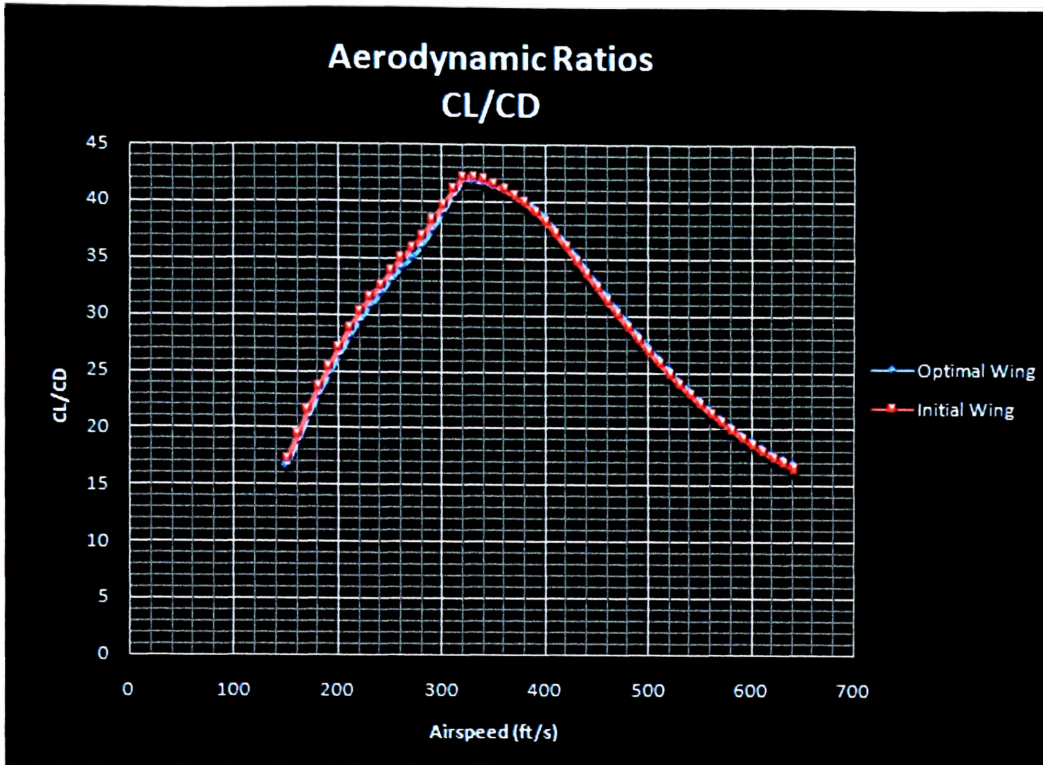
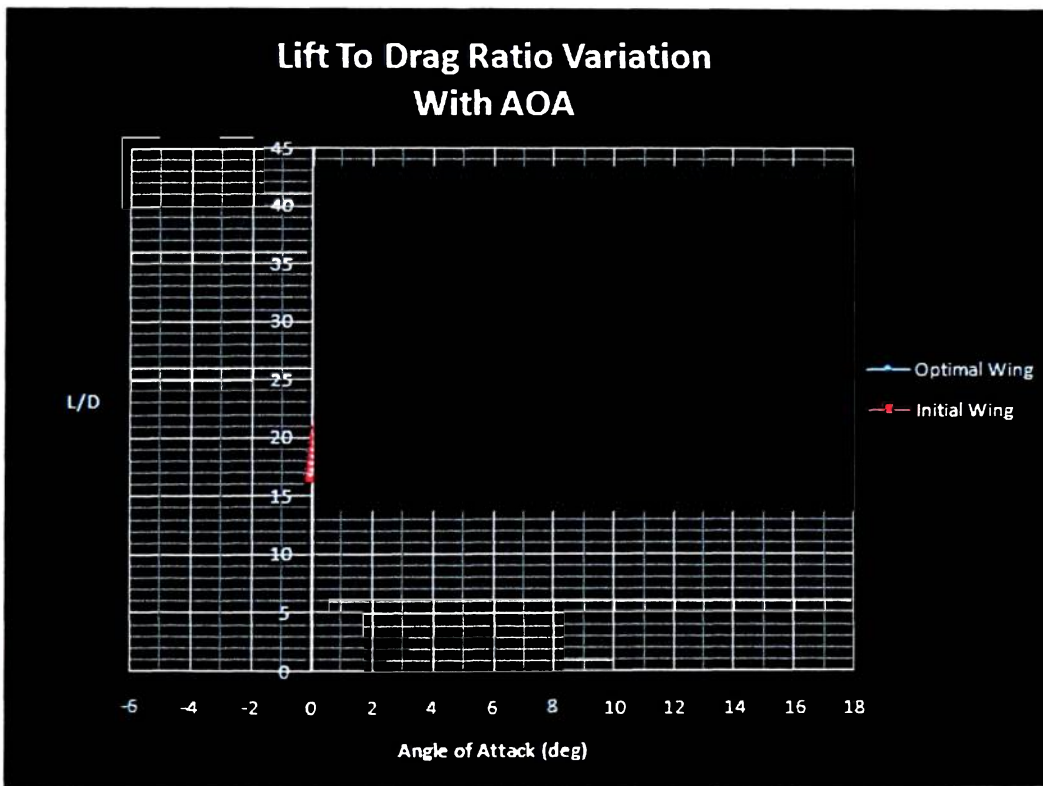


Figure 4. 7: Variation of $C_L^{0.5}/C_D$ with velocity

The lift-to-drag ratio variation with airspeed and angle of attack for both wings is presented in figures 4.8 (a) through (b). Again, the efficiency of the optimal wing shows similar behavior to that of the optimal wing during the entire velocity envelope.



(a)



(b)

Figure 4. 8: (a) L/D ratio variation with airspeed. (b) L/D ratio variation with angle of attack.

Figure 4.8 calls for the analysis of the lift to drag ratio at cruise velocity. The simulation managed to determine a wing configuration that maintained the cruise velocity of the initial wing, and allowed an increment of 0.89% in the lift to drag ratio at this condition.

Table 4.3 shows the results of the lift to drag ratio at cruise velocity for both wings.

Table 4.3: Lift to Drag Ratio Analysis at Original Cruise Velocity

Wing	Initial V_{cruise}	Lift to Drag Ratio L/D at Original Cruise Velocity	Percent Increase %
Initial	460.64 ft/s	38.16	0.89%
Optimal	460.64 ft/s	38.50	

A closer look at the drag behavior of both wings during cruise explains how the optimal wing shows a considerable drag reduction. The drag curve in this analysis is produced by the simulation through the variation of the total drag with flight velocity. Figure 4.9 shows how the optimal wing configuration shows a slight reduction in drag with increasing airspeed relative to the drag curve of the initial wing. Overall, the geometric configuration of the optimal wing presents a drag reduction that generates a positive effect on the lift to drag ratio at cruise velocity accounting for almost a 1% increment in this parameter that dictates aerodynamic efficiency.

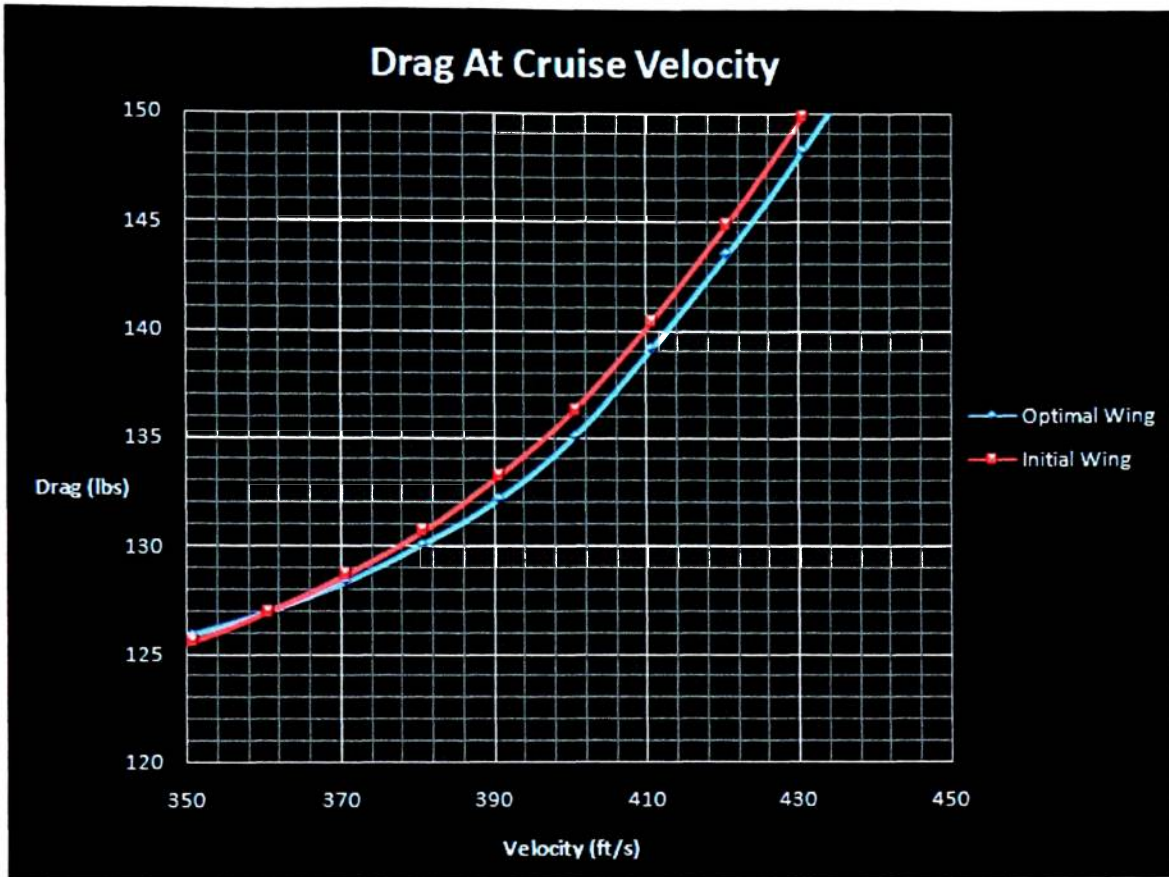


Figure 4. 9: Drag Variation with Velocity: Initial vs. Optimal Wing

Range Analysis

It is important to note that the simulation computes the range of each wing based on the simplified range equation for a jet propelled airplane (Breguet formulation) assuming a constant thrust specific fuel consumption, altitude, and fuel weight. The amount of fuel carried by each wing was calculated analytically by the program by determining the tank volume of each wing, and then determining its corresponding fuel weight per section 3.4. For this particular analysis, a fuel density of 6.4 lb/gal (JP-4) is assumed. Figure 4.10 compares the calculated fuel volume of both wings.

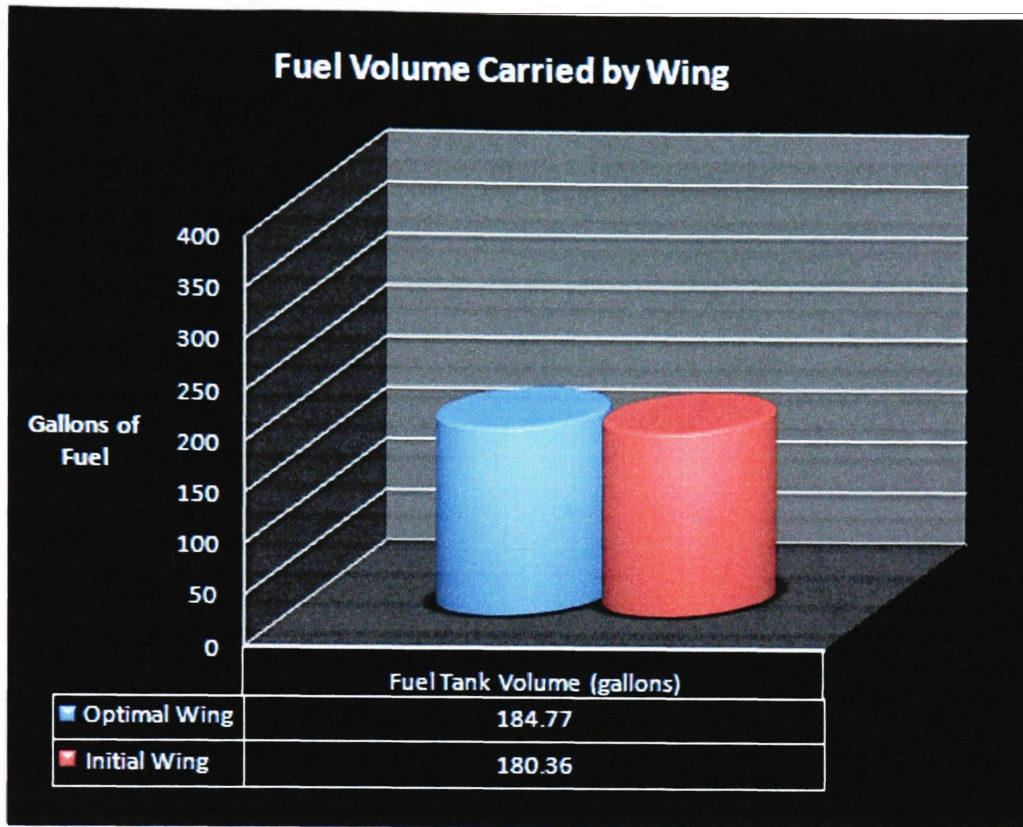


Figure 4.10: Wing Fuel Volume

The increase in fuel tank volume is given to the increment in root chord. The optimal wing can carry approximately more than 4 gallons of fuel corresponding to an overall fuel volume gain of 2.44% with respect to the initial wing. Table 4.4 presents the corresponding fuel weights for both wing configurations.

Table 4.4: Fuel Weight Comparison

Wing Type	Fuel Weight (W_f)
Initial Wing	$W_f = 1154.31$ lbs
Optimal Wing	$W_f = 1182.51$ lbs

Aerodynamic efficiency has a first order effect on range, since the theoretical maximum range for a jet airplane is obtained by flying at the corresponding velocity where $(C_L^{0.5}/C_D)$ is a maximum. The simulation calculates the maximum range for each wing through equation 4.1.

$$R = \frac{2}{c_t} \sqrt{\frac{2}{\rho_\infty S}} \left(\frac{C_L^{1/2}}{C_D} \right)_{\max} \left(W_0^{1/2} - (W_0 - W_f)^{1/2} \right) \quad [4.1]$$

Figure 4.11 shows the calculated theoretical maximum range for each wing configuration at standard sea level assuming a thrust specific fuel consumption of 0.6 lb/lbhr. The optimal wing has a maximum range of 4243.15 miles and the initial wing has a maximum range of 4097.74 miles. Overall, there is a 3.5% increase in range mainly due to the increment in fuel volume, and lift to drag ratio at that condition.

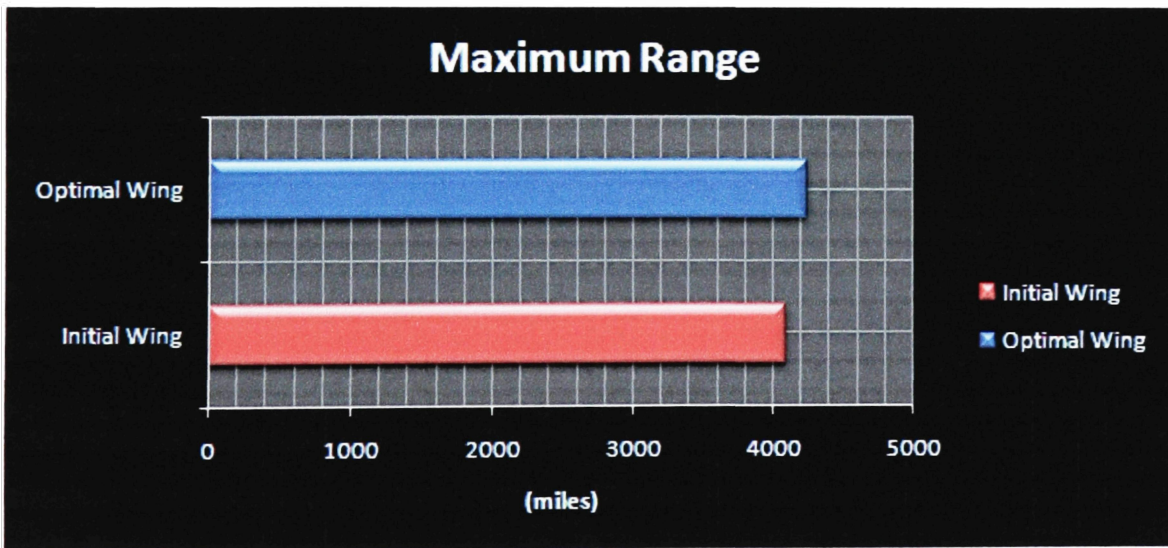


Figure 4.11: Theoretical Maximum Range Comparison

4.4 Wing Weight Comparison

The simulation calculated the wing weight of the optimal and the initial wing configurations based on the statistical regression equation described earlier in section 3.5. The following table illustrates a comparison between the initial wing weight and the optimal wing weight.

Table 4.5: Estimated Wing Weight Comparison.

Wing Type	Wing Weight (W_w)	% Percent Difference
Initial Wing	$W_w = 419.72$ lbs	4.45%
Optimal Wing	$W_w = 401.01$ lbs	



Figure 4.12: Estimated Wing Weight

Table 4.5 shows that the optimal wing has an estimated wing weight of 401.01 lbs which represents a wing weight reduction of 4.45%. This weight reduction can be accounted for by the wing area decrement seen on the optimal wing configuration.

4.5 Spanwise Distribution of the Pressure Coefficient

The simulation also computes the spanwise distribution of the pressure coefficient for each wing at the wing's cruise velocity and angle of attack. Since the vortex lattice analysis determines the lift per unit span, the pressure at each individual panel is calculated by dividing the lift force at each panel by the panel area, and then dividing the result by the dynamic pressure. Figures 4.13 and 4.14 illustrate the results.

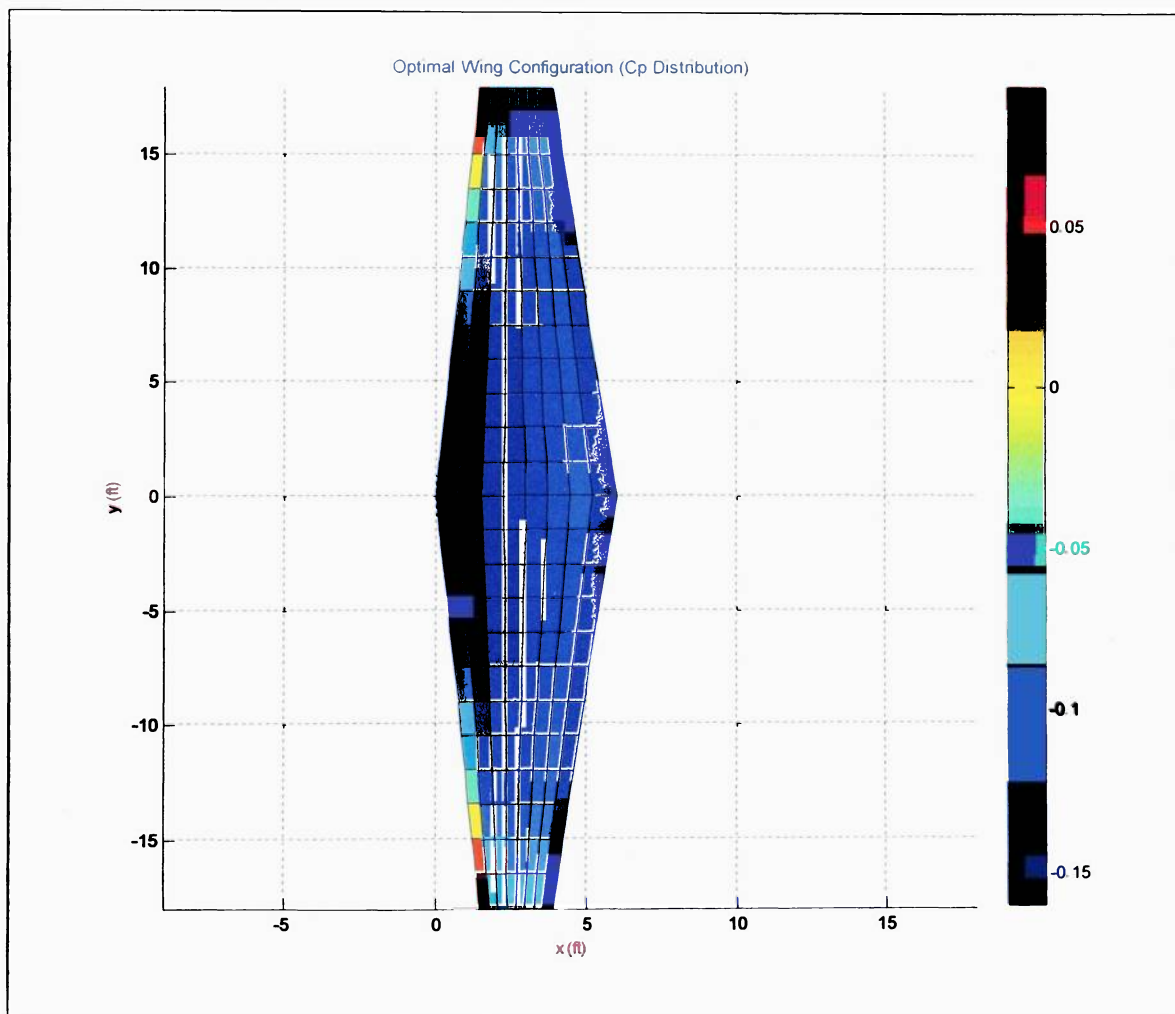


Figure 4.13: Cp Spanwise distribution: Optimal Wing

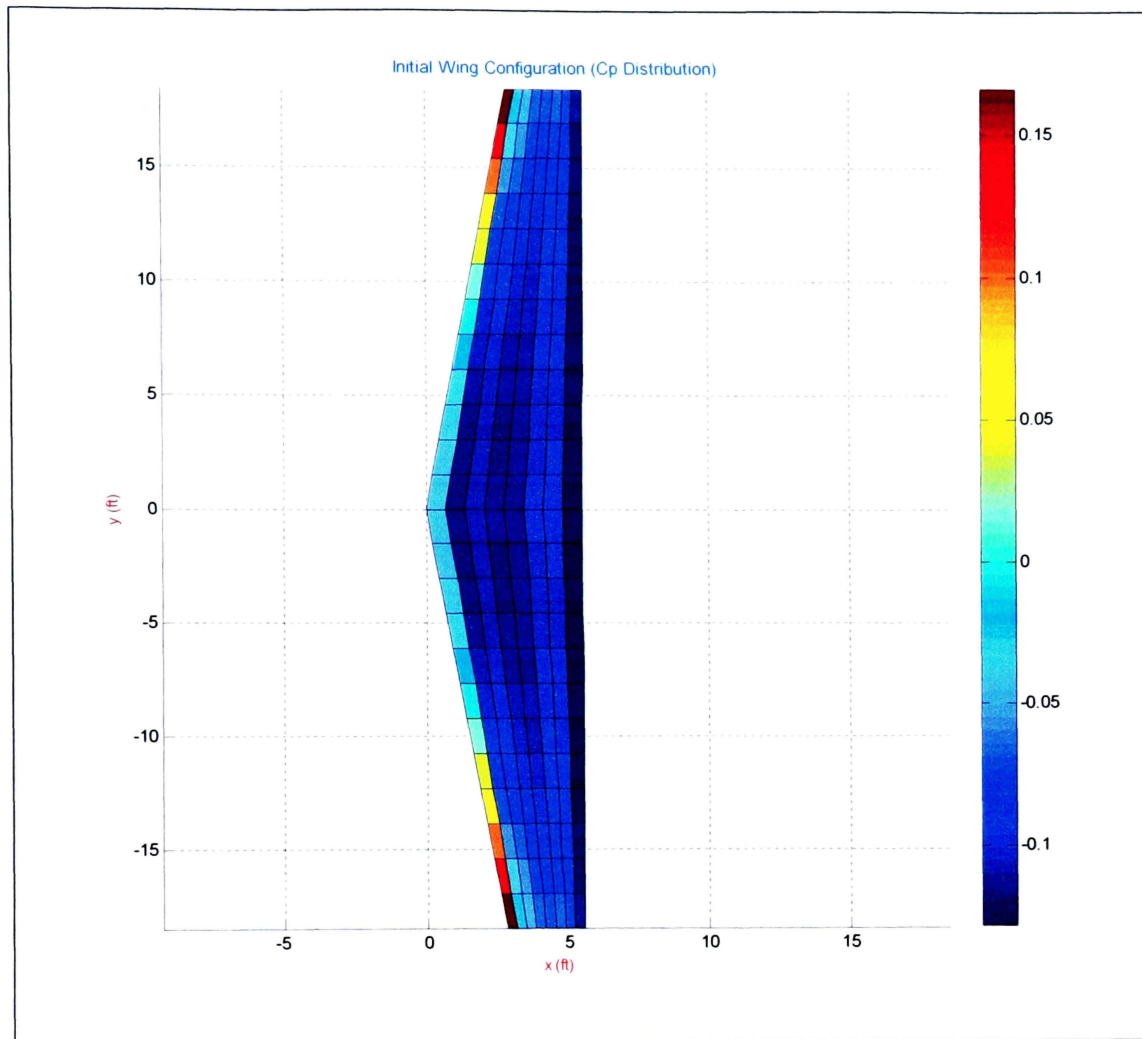


Figure 4.14: Cp Spanwise distribution: Initial Wing

There is a notable difference in the spanwise pressure distribution between both wings. The leading edge sweep of the initial wing shows a positive average pressure gradient towards the wingtips which is higher in value than the maximum positive pressure gradient on the leading edge of the optimal wing. Overall, the optimal wing shows a better pressure distribution at its cruise angle of attack.

4.6 Further Considerations: C_L vs. Alpha Curve

Figure 4.14 shows the lift coefficient variation with angle of attack for the optimal and the initial wing. Both wings have the same washout about the quarter chord resulting in approximately the same zero lift angle of attack. The lift curve slope for both wings is similar, since the optimal and the initial wing share the same wing section geometry and have similar aspect ratios differing only by approximately 2%.

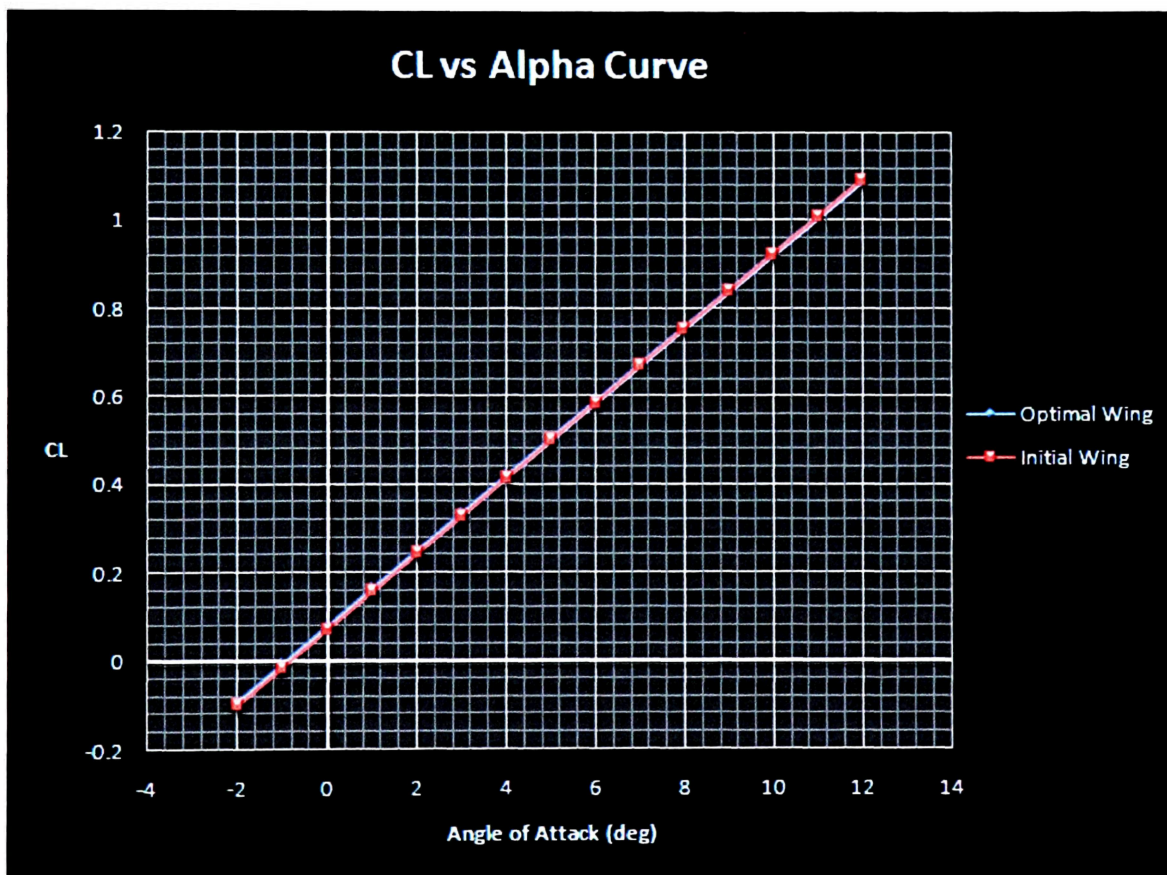


Figure 4.15: C_L vs. Alpha Curve: Initial vs. Optimal

4.7 Validation

The aerodynamic optimization process and the implementation of the vortex lattice method will be validated through the use of the NACA technical report 1422: *Experimental and Calculated Characteristics of Three Wings of NACA 64-210 and 65-210 Airfoil Sections With and Without 2° Washout*.

This NACA report conducted an investigation to determine the effects of airfoil section and washout on the experimental and calculated characteristics of 10-percent thick wings. Three wings of aspect ratio 9 and a taper ratio of 0.4 were tested in the Langley 19-foot pressure tunnel. The tests were made at a dynamic pressure of approximately 85 pounds per square foot, corresponding to a Reynolds number of approximately 4,400,000 and a Mach number of 0.17. The wing to be studied and compared on this report contains a NACA 65-210 section, and includes two configurations that have a geometric twist of 0° and 2° respectively. The output of the vortex lattice optimization routine (VORT) will be compared to the published experimental data of the C_L vs. Alpha curve and the drag polar for the wing with the NACA 65-210 airfoil section of the NACA technical report.

4.7.1 C_L vs. Alpha Curve Comparison

The geometry of the tested wing was modeled with respect to the specifications given by the NACA technical report 1422. The dynamic conditions of the test were matched to the ones of the wind tunnel through the values of Reynolds number and Mach number. The vortex lattice routine used a total of 12 spanwise panels and 8 chordwise panels to model the geometry of the wing and the variation of camber correctly. Figure

4.16 illustrates the variation of C_L vs. α for the two experimental wing configurations and compares them with the output of the vortex lattice optimization routine (VORT).

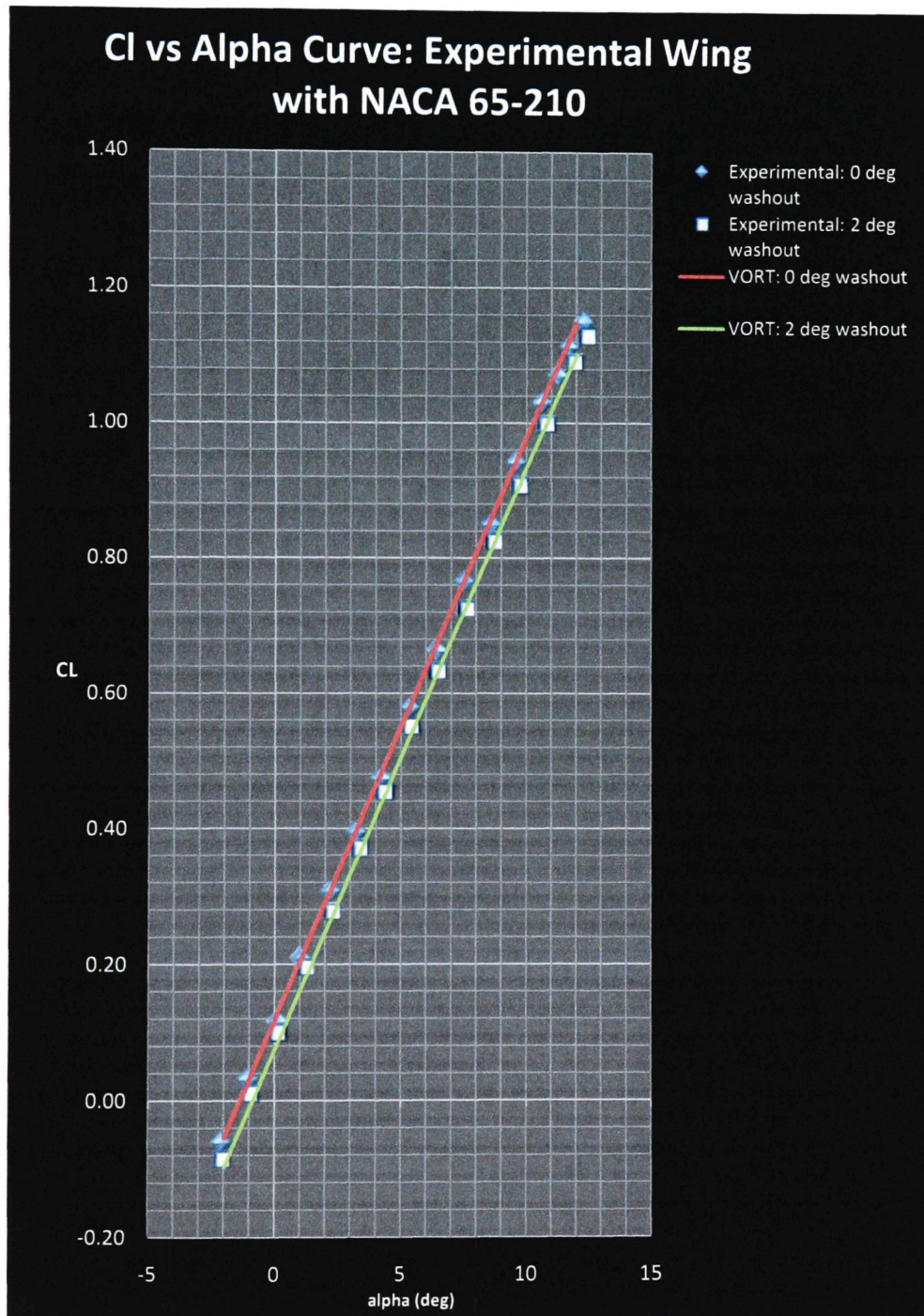


Figure 4.16: Cl vs Alpha Curve: Validation

There is excellent agreement between the experimental data and the VORT output data for both wing configurations. Table 4.6 presents a summary of the results for calculated and experimental data of the C_L vs. α curve. Experimental results are taken from Table I NACA TN No. 1422.

Table 4.6: Experimental vs. Calculated Results: C_L vs. α curve

Wing	$dC_L/d\alpha$		$\alpha_{(L=0)}$	
	Experimental	VORT	Experimental	VORT
65-210 Wing 0° washout	0.085	0.085	-1.3°	-1.4°
65-210 Wing 2° washout	0.085	0.085	-1.0°	-0.90°

The lift curve slope calculated by the vortex lattice routine (VORT) for both tested wings with NACA 65-210 airfoil sections, presents excellent numerical correlation when compared to the experimental results of the NACA technical document.

4.7.2 Drag Polar Comparison

The calculated drag coefficients by the vortex lattice routine are compared with the experimental drag coefficients for both wing configurations with NACA 65-210 airfoil

sections. The following figures illustrate the comparison between the experimental drag polar and the calculated drag polar for each wing configuration.

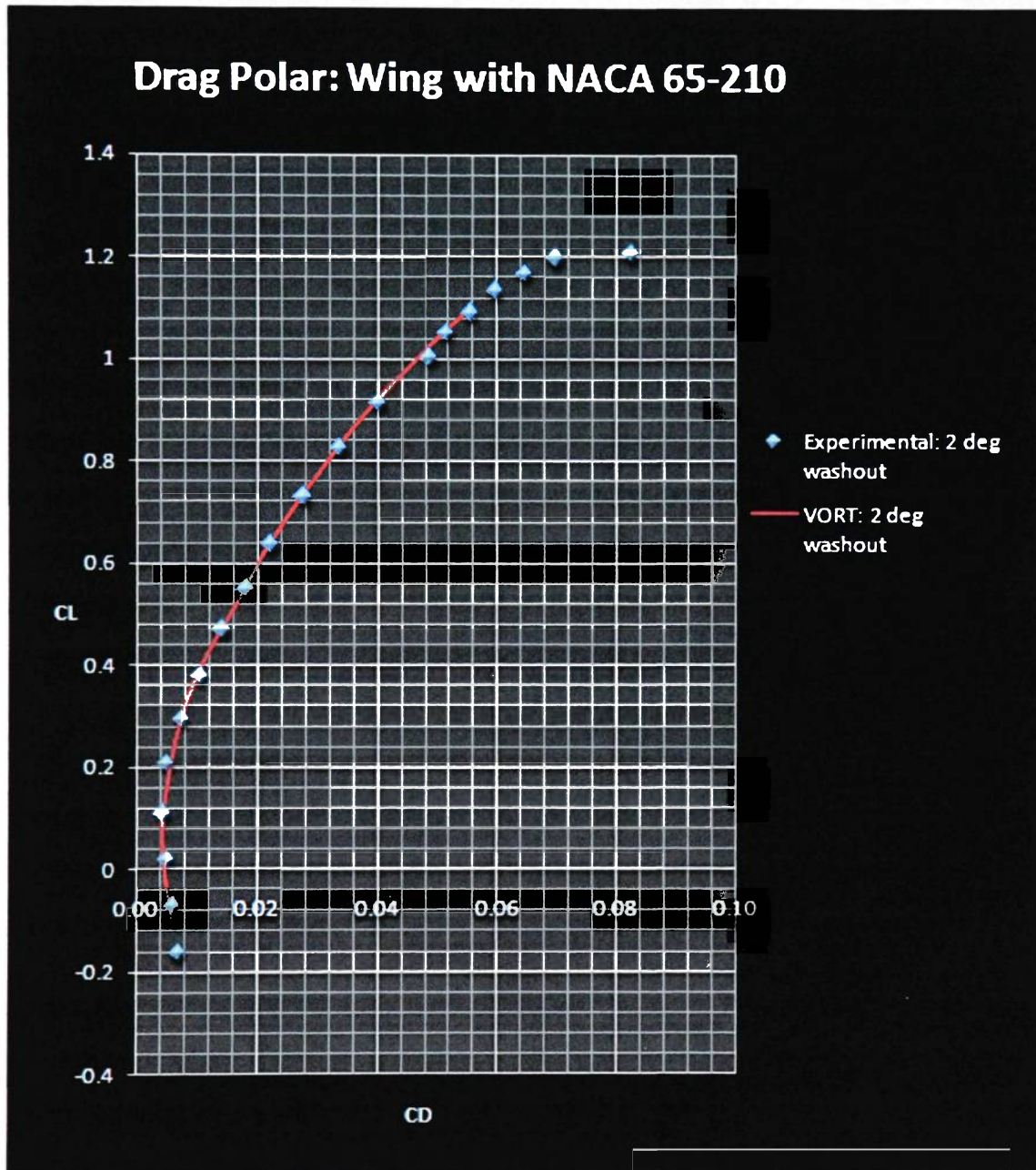


Figure 4.17: Drag Polar Diagram (0 deg washout): Experimental vs. Calculated

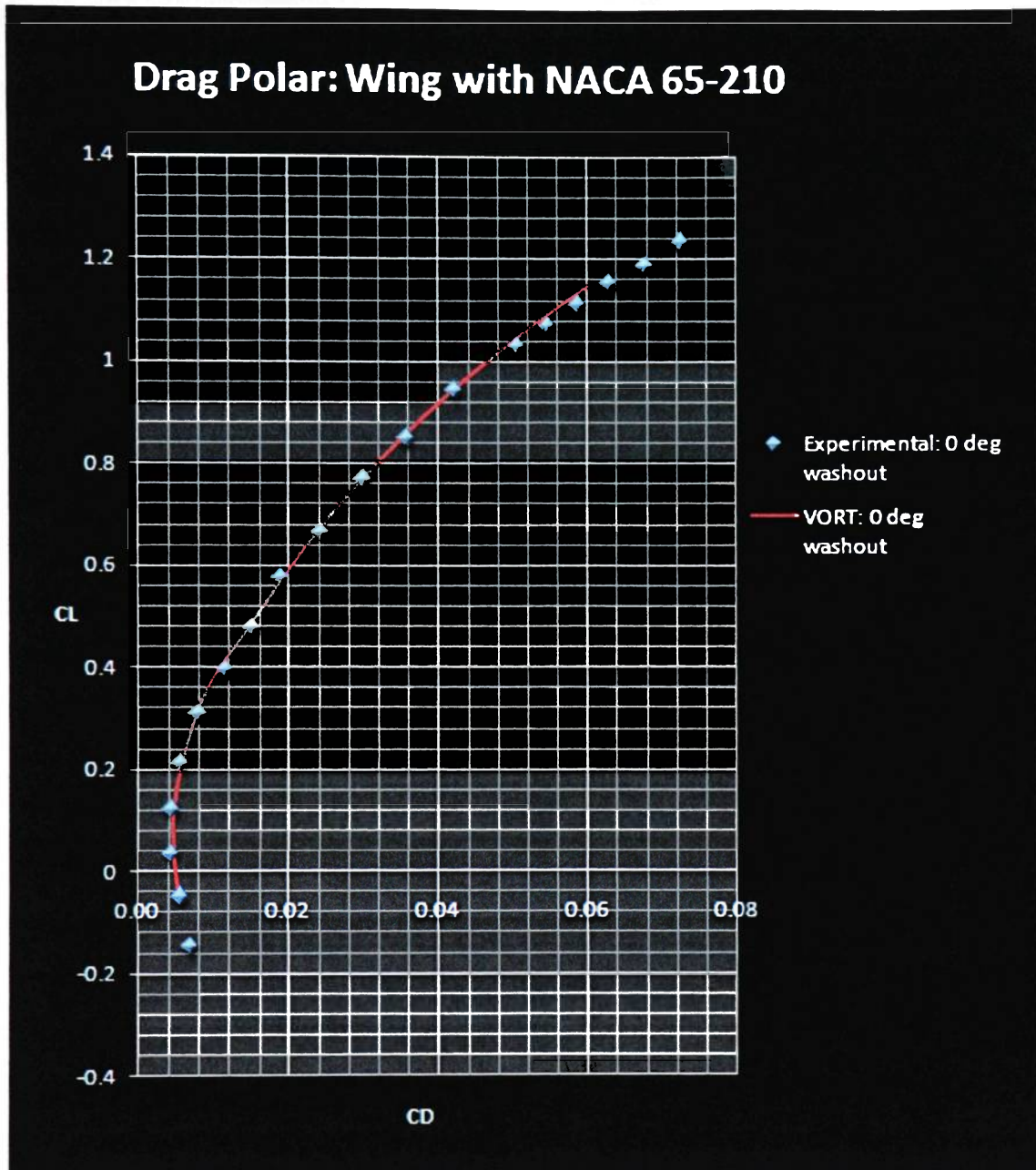


Figure 4.18: Drag Polar Diagram (2 deg washout): Experimental vs. Calculated

The calculated drag polar shown in figures 4.17 and 4.18 presents good correlation with the experimental data for both wing configurations.

The wing's profile drag is calculated using the method described in section 3.3.3.1. A two dimensional lookup table was built for this analysis using the data provided in

reference [6]. Figures 4.19 and 4.20 illustrate the variation of CD_0 with C_L for both wing configurations. VORT results are compared with the available experimental data of NACA TN 1422.

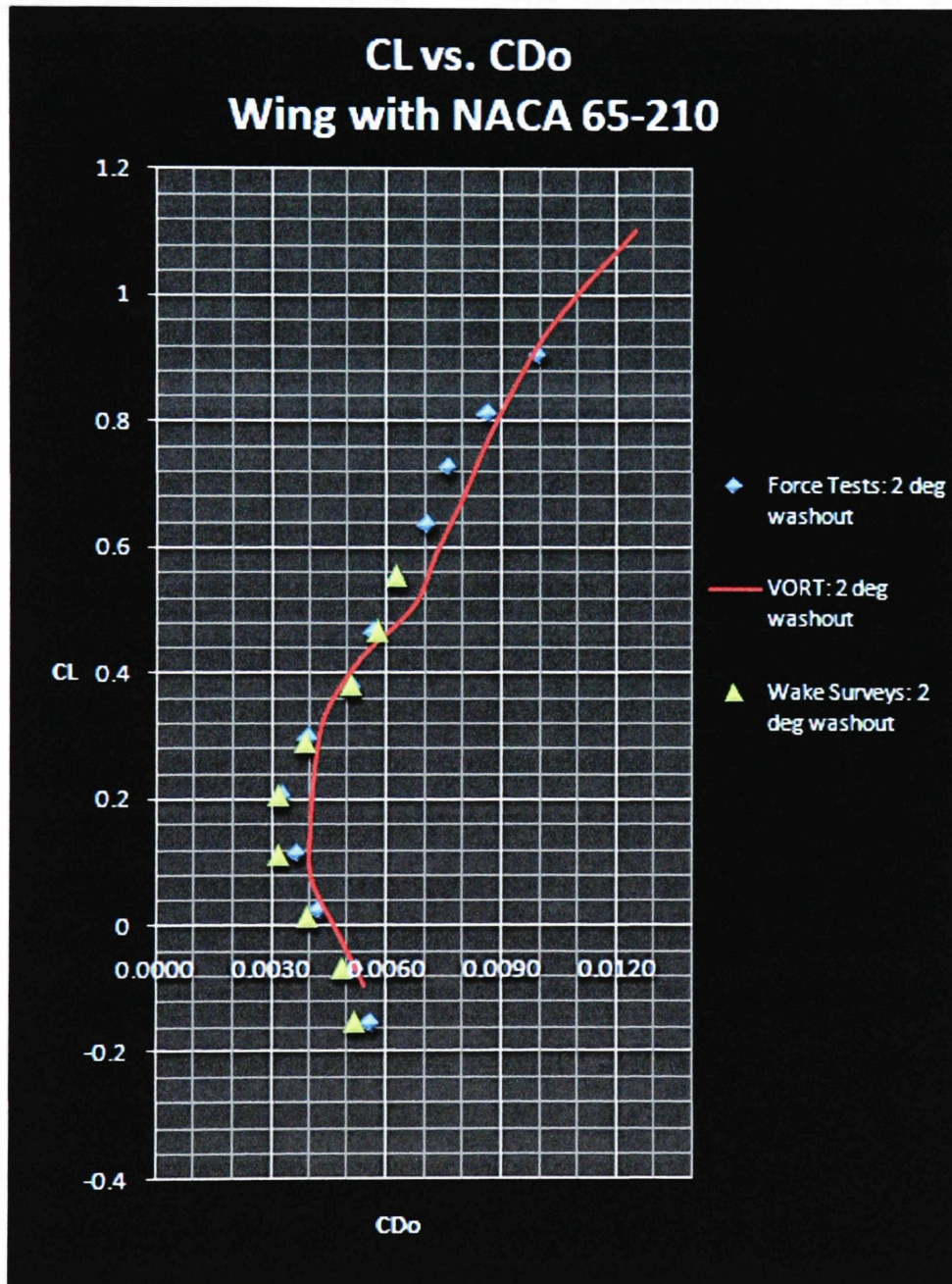


Figure 4.19: Profile Drag Variation (2 deg washout): Experimental vs. Calculated

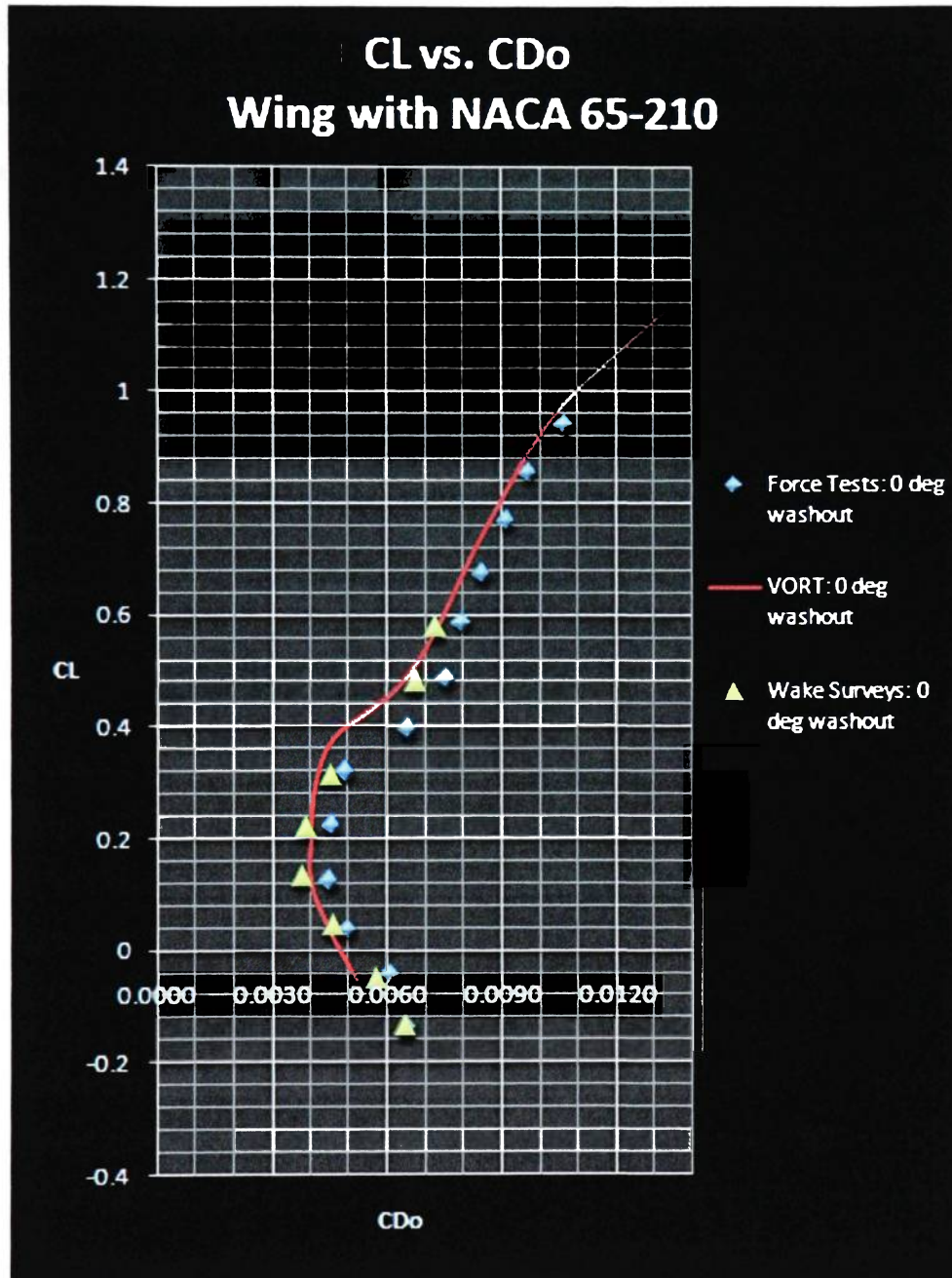


Figure 4.20: Profile Drag Variation (0 deg washout): Experimental vs. Calculated

Overall the calculated drag polar for both wings of NACA 65-210 airfoils sections behaves in a similar fashion as the experimental drag polar does. Table 4.8 shows a comparison between the experimental minimum profile drag coefficient and the calculated minimum profile drag coefficient.

Table 4.7: Comparison between Experimental and Calculated C_{D0}

Wing	C_{D0min}		
	Force Tests	Wake Surveys	VORT
65-210 Wing 0° washout	0.0044	0.0039	0.0041
65-210 Wing 2° washout	0.0040	0.0039	0.0040

In general, the aerodynamic analysis based on the vortex lattice method used during this research study, and during the optimization analysis presents valid results that correlate experimental data in published literature.

4.8 Error Analysis

Various sources of error can be traced during the numerical simulation and the aerodynamic analysis. These two sources should be treated separately, and the evaluation of their accuracy calls for an individual analysis on their behalf.

4.8.1 Factors affecting the accuracy of the Vortex Lattice Method

The vortex lattice analysis in this study is limited to irrotational, inviscid, incompressible flow. During the simulation, the airspeed variation reached Mach numbers above 0.4 just in the verge of compressible flow allowing for significant error in the approximation of the flow field over the wing. Furthermore, the analysis lacks the use of compressibility corrections and does not utilize drag divergence approximations.

4.8.2 Computational Error in the Numerical Analysis

The computational error in the numerical analysis can be traced to the mathematical interpolation and extrapolation performed during the simulation. The calculation of the profile drag coefficient uses a two-dimensional lookup table that uses a linear interpolation and extrapolation method embedded in the application. Other computational calculations such as the one performed for the dynamic viscosity coefficient and the density altitude use linear interpolation methods that allow error in the calculation.

During the performance analysis, the search index proceeds to calculate the corresponding drag coefficient through a four-dimensional look-up table that uses a binary search method and a spline interpolation method resulting in error accumulation. Overall, the error in the numerical analysis is traced to the mathematical interpolation and extrapolation performed during the simulation

5 CONCLUSIONS

The optimization technique efficiently linked the Vortex Lattice Method to successfully optimize the geometry of a generic wing with a NACA 65-210 airfoil section. The objective of the optimization analysis centered on maximizing the lift to drag ratio at cruise velocity of the initial wing configuration, while maximizing theoretical range (Breguet formulation) and minimizing wing weight. The aerodynamic optimization technique incorporated five geometric design variables such as wing area, wingspan, taper ratio, leading edge sweep and geometric twist. The simulation studied a possible total of 324 wing combinations corresponding to the constraint variation of the five geometric design variables. The wing section geometry, defined by the NACA 65-210 airfoil, was the only defining parameter that remained constant throughout the analysis.

The aerodynamic optimization tool gave as an output an optimal wing geometry having a 1.30% decrement in wing area, a 2.7% decrement in wingspan, and a 20 % decrement in taper ratio when compared to the initial wing configuration. Consequently, there was a reduction in leading edge sweep of approximately 46.3%, and no change in the geometric twist angle when compared to the initial wing configuration.

Overall, the optimal wing configuration performed as expected and presented superior aerodynamic efficiency. The lift to drag ratio at the original cruise velocity was increased by almost 1%. Theoretical maximum range increased by 3.5%, and wing weight was reduced by 4.45%.

In general, the aerodynamic optimization succeeded in optimizing the initial wing configuration to allow optimal performance during cruise while minimizing wing weight and increasing maximum theoretical range. The NACA technical document 1422 was used to validate the aerodynamic analysis used by the simulation.

6 RECOMMENDATIONS AND FUTURE WORK

The aerodynamic optimization technique proposed in this analysis could extend its application to different Mach number regimes. Even though this analysis was constraint to a subsonic level, it has the potential to adapt compressibility corrections to model flow in the transonic and supersonic regimes.

The vortex lattice analysis in this code was designed to model wings without variable wing section geometry. A solution would be to adapt an algorithm that modifies the camber distribution of the airfoil through the variation of the terms within the 6th degree polynomial approximation of the airfoil's mean camber line. The wing section geometry could eventually change in order to further increase the aerodynamic efficiency of the wing. The vortex lattice analysis could expand its application to model control surfaces such as ailerons or high lift devices like flaps or slats.

The optimization technique could incorporate a gradient method or an optimal search method that could replace the numerical brute force technique used in this analysis reducing computational time and allowing faster convergence.

REFERENCES

1. Holst, Terry. "Genetic Algorithms Applied to Multi-Objective Aerodynamic Shape Optimization." NASA/TM-212846, 2005.
2. Anderson, John. Fundamentals of Aerodynamics. New York: McGraw Hill, 2001.
3. W.H. Mason. "Aerodynamics of 3D Lifting Surfaces through Vortex Lattice Methods." Virginia Polytechnic Institute and State University, 1998. <http://www.aoe.vt.edu/~mason/Mason_f/CAtxtChap6.pdf>
4. Bertin, J. and Smith, M. Aerodynamics for Engineers. New Jersey: Prentice Hall, 1998
5. Anderson, John. Aircraft Performance and Design. New York: McGraw Hill, 1998
6. Abbot, Ira, Albert von Doenhoff. Theory of Wing Sections. New York: Dover Publications, Inc., 1959.
7. S.Chen and F.Zhang. "A Preliminary Study of Wing Aerodynamic, Structural and Aeroelastic Design and Optimization." AIAA-2002-5656, 2002.
8. Raymer, Daniel. Aircraft Design: A Conceptual Approach. Reston: American Institute of Aeronautics and Astronautics, Inc., 1999.
9. Klatz, J. and Plotkin, A.: Low Speed Aerodynamics. Cambridge University Press, Cambridge, 2001
10. Jackson, Paul. Jane's All the World's Aircraft. Cambridge University Press, UK, 2005
11. Sivells, James C. "Experimental and Calculated Characteristics of Three Wings of NACA 64-210 and 65-210 Airfoil Sections with and without 2° Washout." NACA TN. 1422, 1947.
12. McCormick, B.W. Aerodynamics, Aeronautics, and Flight Mechanics. New York: John Wiley & Sons, 1979.

APPENDIX A

MATLAB CODE

Created by: Santiago Pinzon

Content:

Appendix A.1: *VORT_executable.m*.....91

Appendix A.2: Drag polar diagram of the NACA 65-210 airfoil.....121

Appendix A.1: *VORT_executable.m* – Wing Optimization program based on Vortex

Lattice Theory.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Wing Optimization Program based on Vortex Lattice Theory %
%                               VORT_1                      %
%               Done by Santiago Pinzon                    %
%                               AE - 700                   %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
set(0,'DefaultFigureWindowStyle','docked')
clc;
cla;
clf;
clear;
close all;
fprintf('\nSantiago Pinzon, AE-700 Wing Optimization Program based on
Vortex Lattice Theory\n')
fprintf('\n Welcome to VORT_1\n')

%-----
%-----
% USER INPUT COMMAND PROMPT

prompt = {'Please Enter the Following Data Starting with The Initial
Design Wingspan (ft):','Enter Initial Design Wing Area (ft^2):','Enter
Taper Ratio :','Enter Dihedral Angle (deg):','Enter LE sweep angle
(deg)','Enter Washout Angle about the c/4 (deg)'};
name = 'INITIAL WING GEOMETRY ';
num_lines = 1;
default = {'37','153','0.5','0','8.649152791','2'};
options.Resize='on';
options.WindowStyle='normal';
answer = inputdlg(prompt,name,num_lines,default,options);

prompt_1 = {'Enter # panels in the spanwise direction halfspan
N','Enter # panels in the chordwise direction P '};
name = 'VLM CONDITIONS ';
num_lines = 1;
default = {'12','8'};
options.Resize='on';
options.WindowStyle='normal';
answer_1 = inputdlg(prompt_1,name,num_lines,default,options);

prompt_2 = {'Enter Expected Cruising Altitude h (ft)','Enter Aircraft
Gross Weight (lbs)'};
name = 'INITIAL CONDITIONS ';
num_lines = 1;
default = {'0','5200'};
options.Resize='on';
```

```

options.WindowStyle='normal';
answer_2 = inputdlg(prompt_2,name,num_lines,default,options);

prompt_3 = {'Enter Thrust Specific Fuel Consumption TSFC @ Alt
(lb/lb*h)'};
name = 'Engine Performance (Turbofan) ';
num_lines = 1;
default = {'0.60'};
options.Resize='on';
options.WindowStyle='normal';
answer_3 = inputdlg(prompt_3,name,num_lines,default,options);

% Variable Definition

% WING
b = str2double(answer{1,1}); % wingspan
S = str2double(answer{2,1}); % wing area
lambda = str2double(answer{3,1}) ; % taper ratio
phi = str2double(answer{4,1}); % dihedral angle
sweep_a =str2double(answer{5,1}) ; % Leading edge sweep
twist_a = str2double(answer{6,1}); % Washout about the c/4 of the wing

% VLM ANALYSIS
N = str2double(answer_1{1,1}); % # of panels (half span)
P = str2double(answer_1{2,1}); % # of chordwise panels

% INITIAL CONDITIONS
h = str2double(answer_2{1,1}); % Altitude
W = str2double(answer_2{2,1}); % Gross Weight

% Engine Characteristics
TSFC = str2double(answer_3{1,1});

% Limit Load Factor:

l_factor = 2.8;
N_z = l_factor*1.5;

% Initial Variables Required for Program Initial

x_percent = 0.70;
Jet_fuel_d = 6.4;

AR_initial = (b^2)/S;

R_gas = 1716;
P_atm = 2116*(1-0.0000068753*h)^5.2561;
oatr = 518.69*(1-0.0000068753*h);
rho = P_atm/(R_gas*oatr);

tic

```



```
fprintf('\n\nProgram in progress please wait.....\n\n')
```

```
%-----  
-----  
% AIRFOIL: DRAG POLAR DATA FOR NACA 64-415 (THEORY OF WING SECTIONS)  
cl_65210 = [0.00 0.10 0.20 0.30 0.40 0.50 0.60 0.70 0.80 0.90 1.00  
1.10];  
Reynolds = [3.1*10^6 6.0*10^6 9.0*10^6];  
CD_pmatrix = [0.004090 0.005693 0.005271  
0.004175 0.003837 0.003837  
0.004133 0.003880 0.003627  
0.004217 0.004006 0.003795  
0.004343 0.005229 0.005735  
0.007042 0.006663 0.006283  
0.007970 0.007000 0.006620  
0.008560 0.007633 0.007042  
0.009319 0.008392 0.007548  
0.010247 0.009024 0.008181  
0.011428 0.009952 0.009024  
0.013578 0.011006 0.010163];
```

```
%Induced Drag Factor Data: McCormick, B.W Aerodynamics  
taper_ratio = [0.042 0.200 0.400 0.600 0.800 1];  
Aspect_ratio = [4 6 8 10];  
Delta_var = [0.04022 0.06659 0.07732 0.09430  
0.00134 0.00670 0.01028 0.01564  
0.00134 0.00402 0.00804 0.01162  
0.01028 0.01877 0.02637 0.03352  
0.02547 0.03978 0.05453 0.07061  
0.04156 0.06346 0.08626 0.10726];
```

```
%-----  
-----  
%Dynamic Viscosity Table  
visc = [ 0.374 0.372 0.370 0.368 0.366 0.364 0.362 0.360 0.358 0.355  
0.353 0.351 0.349 0.347 0.345 0.343 0.341 0.339 0.337 0.335 0.332 0.330  
...  
0.328 0.326 0.324 0.322 0.319 0.317 0.315 0.313 0.311 0.308  
0.306 0.304 0.302 0.300 0.297 0.297 0.297 0.297 0.297 0.297 0.297 0.297  
0.297 0.297 0.297 0.297...  
0.297 0.297 0.297]*10^-6;  
alti = [ 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23  
24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47  
48 49 50]*10^3;  
sim Dynamic_viscosity
```

```
%-----  
-----
```

```
Cl_max = 1.4; % NACA 65-210 airfoil  
CL_maxWing = 0.9*Cl_max;% For aspect ratios higher than 5 (Raymer)  
rho_SSL = 2.3769*10^-3;% Density at sea level
```

```
%-----  
%Memory Allocation  
%-----
```

```
y1_m = zeros(P,N);  
    y1_mt = zeros(P,N);  
    le_m1 =zeros(P,N);  
    le_m1t = zeros(P,N);  
    te_m1 = zeros(P,N);  
    te_m1t = zeros(P,N);  
    c_lm = zeros(P,N);  
    c_lmt = zeros(P,N);  
    x1_m = zeros(P,N);  
    x1_mt= zeros(P,N);  
    x1_mz= zeros(P,N);  
    z1_mt = zeros(P,N);  
    z1_m= zeros(P,N);  
  
    y2_m= zeros(P,N);  
    y2_mt = zeros(P,N);  
    le_m2 =zeros(P,N);  
    le_m2t = zeros(P,N);  
    te_m2 = zeros(P,N);  
    te_m2t= zeros(P,N);  
    c_2m = zeros(P,N);  
    c_2mt = zeros(P,N);  
    x2_m = zeros(P,N);  
    x2_mt = zeros(P,N);  
    x2_mz = zeros(P,N);  
    z2_mt = zeros(P,N);  
    z2_m = zeros(P,N);  
  
    y3_m = zeros(P,N);  
    y3_mt = zeros(P,N);  
    le_m3 =zeros(P,N);  
    le_m3t = zeros(P,N);  
    te_m3 = zeros(P,N);  
    te_m3t= zeros(P,N);  
    c_3m = zeros(P,N);  
    c_3mt = zeros(P,N);  
    x3_m = zeros(P,N);  
    x3_mt = zeros(P,N);  
    x3_mz = zeros(P,N);  
    z3_mt = zeros(P,N);  
    z3_m = zeros(P,N);  
  
    y_M = zeros(P,N);  
    y_Mt = zeros(P,N);  
    le_M =zeros(P,N);  
    le_Mt = zeros(P,N);  
    te_M = zeros(P,N);  
    te_Mt= zeros(P,N);  
    chord_M = zeros(P,N);  
    chord_Mt = zeros(P,N);
```

```
x1_M = zeros(P,N);
x2_M = zeros(P,N);
z1_M = zeros(P,N);
z2_M = zeros(P,N);
x_M = zeros(P,N);
x_Mt= zeros(P,N);
x_Mz = zeros(P,N);
z_Mt = zeros(P,N);
z_M = zeros(P,N);
yp_M = zeros(P,N);
```

```
y1_N =zeros(P,N);
y1_Nt = zeros(P,N);
le_N1 =zeros(P,N);
le_N1t = zeros(P,N);
te_N1 = zeros(P,N);
te_N1t= zeros(P,N);
c_1 = zeros(P,N);
c_1Nt = zeros(P,N);
x1_1N = zeros(P,N);
x2_1N = zeros(P,N);
z1_1N = zeros(P,N);
z2_1N = zeros(P,N);
x1_N = zeros(P,N);
x1_Nt = zeros(P,N);
x1_Nz = zeros(P,N);
z1_Nt = zeros(P,N);
z1_N = zeros(P,N);
```

```
y2_N = zeros(P,N);
y2_Nt =zeros(P,N);
le_N2 =zeros(P,N);
le_N2t = zeros(P,N);
te_N2 = zeros(P,N);
te_N2t= zeros(P,N);
c_2 = zeros(P,N);
c_2Nt = zeros(P,N);
x1_2N = zeros(P,N);
x2_2N = zeros(P,N);
z1_2N = zeros(P,N);
z2_2N = zeros(P,N);
x2_N = zeros(P,N);
x2_Nt = zeros(P,N);
x2_Nz = zeros(P,N);
z2_Nt = zeros(P,N);
z2_N = zeros(P,N);
```

```
y1p_N = zeros(P,N);
y2p_N = zeros(P,N);
```

```
A1_m = zeros(1,3,N);
A2_m = zeros(1,3,N);
A3_m = zeros(1,3,N);
A4_m = zeros(1,3,N);
```

```

A5_m = zeros(1,3,N);
A6_m = zeros(1,3,N);
A7_m = zeros(1,3,N);
A8_m = zeros(1,3,N);

B1_m = zeros(1,3,N);
B2_m = zeros(1,3,N);
B3_m = zeros(1,3,N);
B4_m = zeros(1,3,N);
B5_m = zeros(1,3,N);
B6_m = zeros(1,3,N);
B7_m = zeros(1,3,N);
B8_m = zeros(1,3,N);

n1_m = zeros(1,3,N);
n2_m = zeros(1,3,N);
n3_m = zeros(1,3,N);
n4_m = zeros(1,3,N);
n5_m = zeros(1,3,N);
n6_m = zeros(1,3,N);
n7_m = zeros(1,3,N);
n8_m = zeros(1,3,N);

n1_unit = zeros(1,3,N);
n2_unit = zeros(1,3,N);
n3_unit = zeros(1,3,N);
n4_unit = zeros(1,3,N);
n5_unit = zeros(1,3,N);
n6_unit = zeros(1,3,N);
n7_unit = zeros(1,3,N);
n8_unit = zeros(1,3,N);

store_angle_attack = zeros(22,1);
store_U_inf = zeros(50,1);

v1_m = zeros(1,3,N);
n_m = zeros(P,N);
Vortex_c = zeros(1,N);

store_S2 = zeros(3,1);
store_b2 = zeros(3,1);
store_lambda = zeros(3,1);
store_sweep_a = zeros(4,1);
store_twist_a = zeros(3,1);

Cl_p = zeros(3,3,3,4,3,22,50) ;
Cd_s = zeros(3,3,3,4,3,22,50) ;
Cd_induced = zeros(3,3,3,4,3,22,50) ;
Cd_profile = zeros(3,3,3,4,3,22,50) ;
Cd_total = zeros(3,3,3,4,3,22,50) ;

TR_result = zeros(50,3,3,3,4,3) ;
PR_result = zeros(50,3,3,3,4,3) ;
L_D_result = zeros(50,3,3,3,4,3) ;
CL_result = zeros(50,3,3,3,4,3) ;

```

```

Cl_w_result = zeros(50,3,3,3,4,3) ;
Cd_w_result = zeros(50,3,3,3,4,3) ;
CL_result2 = zeros(50,3,3,3,4,3) ;
AOA_R_result = zeros(50,3,3,3,4,3) ;
Fuel_weight_result = zeros(50,3,3,3,4,3) ;
Fuel_volume_g_result= zeros(50,3,3,3,4,3) ;

% -----
%
% OPTIMIZATION ANALYSIS %
% -----

%Calculation of Minimum passing velocity for Altitude%
S_input = str2double(answer{2,1});

U_input = sqrt((2*W)/(rho*(S_input)*CL_maxWing));

h1 = waitbar(0,'Please wait... Vortex Lattice Analysis in Progress');
S = str2double(answer{2,1})-4;
for z = 1:1:3
    S = S+2;
    store_S2(z,1)= S;
    waitbar(z/3)

b = str2double(answer{1,1})-2;
for q = 1:1:3
    b = b+1;
    store_b2(q,1) = b;

lambda = str2double(answer{3,1})-0.2 ;
for T = 1:1:3
    lambda = lambda+0.1;
    store_lambda(T,1) = lambda;

sweep_a =str2double(answer{5,1})-8 ;
for p = 1:1:4
    sweep_a = sweep_a+2;
    store_sweep_a(p,1)= sweep_a;

twist_a = str2double(answer{6,1})-3;
for u = 1:1:3
    twist_a = twist_a+1;
    store_twist_a(u,1) = twist_a;

for o=1:P
    for m=1:N

```

```

%Panel Vectors
y1_m(o,m) = ((m-1)*b)/(2*N);
y1_mt(o,m) = b/(2);
le_m1(o,m) = (tan((sweep_a*(pi/180)))*y1_m(o,m));
le_m1t(o,m) = (tan((sweep_a*(pi/180)))*y1_mt(o,m));
te_m1(o,m) =
(2/b)*(((tan(sweep_a*(pi/180)))*(b/2)+((2*S)/(b*(1+lambda)))*(lambda-
1))*y1_m(o,m)+((2*S)/(b*(1+lambda)))*(b/2));
te_m1t(o,m)=
(2/b)*(((tan(sweep_a*(pi/180)))*(b/2)+((2*S)/(b*(1+lambda)))*(lambda-
1))*y1_mt(o,m)+((2*S)/(b*(1+lambda)))*(b/2));
c_lm(o,m) = te_m1(o,m)-le_m1(o,m);
c_lm1t(o,m) = te_m1t(o,m)-le_m1t(o,m);
x1_m(o,m) = ((o-1)*(c_lm(o,m)/P))+le_m1(o,m);
x1_mt(o,m) = ((o-1)*(c_lm1t(o,m)/P));
x1_mz(o,m) = ((o-1)*(c_lm(o,m)/P));
z1_mt(o,m) =
((3/4*(c_lm1t(o,m)*tand(twist_a)))+(1/4*c_lm1t(o,m)*tand(twist_a)))/(c_lm1t
(o,m))*x1_mt(o,m)-(1/4*c_lm1t(o,m)*tand(twist_a));
z1_m(o,m) = ((tand(phi))*((m-
1)*b)/(2*N))+((z1_mt(o,m)/y1_mt(o,m))*y1_m(o,m))+ c_lm(o,m)*((-
0.7825*((x1_mz(o,m)/c_lm(o,m))^6)+(2.2021*((x1_mz(o,m)/c_lm(o,m))^5)
-
(2.3763*((x1_mz(o,m)/c_lm(o,m))^4)+(1.2458*((x1_mz(o,m)/c_lm(o,m))^3))
-(0.3676*((x1_mz(o,m)/c_lm(o,m))^2)+(0.0782*((x1_mz(o,m)/c_lm(o,m)))+
0.0001));

y2_m(o,m) = ((m-1)*b)/(2*N);
y2_mt(o,m) = b/(2);
le_m2(o,m) = (tan((sweep_a*(pi/180)))*y2_m(o,m));
le_m2t(o,m) = (tan((sweep_a*(pi/180)))*y2_mt(o,m));
te_m2(o,m) =
(2/b)*(((tan(sweep_a*(pi/180)))*(b/2)+((2*S)/(b*(1+lambda)))*(lambda-
1))*y2_m(o,m)+((2*S)/(b*(1+lambda)))*(b/2));
te_m2t(o,m)=
(2/b)*(((tan(sweep_a*(pi/180)))*(b/2)+((2*S)/(b*(1+lambda)))*(lambda-
1))*y2_mt(o,m)+((2*S)/(b*(1+lambda)))*(b/2));
c_2m(o,m) = te_m2(o,m)-le_m2(o,m);
c_2mt(o,m) = te_m2t(o,m)-le_m2t(o,m);
x2_m(o,m) = (o*(c_2m(o,m)/P))+le_m2(o,m);
x2_mt(o,m) = (o*(c_2mt(o,m)/P));
x2_mz(o,m) = (o*(c_2m(o,m)/P));
z2_mt(o,m) =
((3/4*(c_2mt(o,m)*tand(twist_a)))+(1/4*c_2mt(o,m)*tand(twist_a)))/(c_2mt
(o,m))*x2_mt(o,m)-(1/4*c_2mt(o,m)*tand(twist_a));
z2_m(o,m) = ((tand(phi))*((m-
1)*b)/(2*N))+((z2_mt(o,m)/y2_mt(o,m))*y2_m(o,m))+ c_2m(o,m)*((-
0.7825*((x2_mz(o,m)/c_2m(o,m))^6)+(2.2021*((x2_mz(o,m)/c_2m(o,m))^5)
-
(2.3763*((x2_mz(o,m)/c_2m(o,m))^4)+(1.2458*((x2_mz(o,m)/c_2m(o,m))^3))
-(0.3676*((x2_mz(o,m)/c_2m(o,m))^2)+(0.0782*((x2_mz(o,m)/c_2m(o,m)))+
0.0001));

y3_m(o,m) = (m*b)/(2*N);
y3_mt(o,m) = b/(2);

```

```

le_m3(o,m) = (tan((sweep_a*(pi/180)))*y3_m(o,m));
le_m3t(o,m) = (tan((sweep_a*(pi/180)))*y3_mt(o,m));
te_m3(o,m) =
(2/b)*(((tan(sweep_a*(pi/180)))*(b/2)+((2*S)/(b*(1+lambda))))*(lambda-
1))*y3_m(o,m)+((2*S)/(b*(1+lambda)))*(b/2));
te_m3t(o,m)=
(2/b)*(((tan(sweep_a*(pi/180)))*(b/2)+((2*S)/(b*(1+lambda))))*(lambda-
1))*y3_mt(o,m)+((2*S)/(b*(1+lambda)))*(b/2));
c_3m(o,m) = te_m3(o,m)-le_m3(o,m);
c_3mt(o,m) = te_m3t(o,m)-le_m3t(o,m);
x3_m(o,m) = ((o-1)*(c_3m(o,m)/P))+le_m3(o,m);
x3_mt(o,m) = ((o-1)*(c_3mt(o,m)/P));
x3_mz(o,m) = ((o-1)*(c_3m(o,m)/P));
z3_mt(o,m) =
((3/4*(c_3mt(o,m)*tand(twist_a))+(1/4*c_3mt(o,m)*tand(twist_a)))/(c_3mt
(o,m)))*x3_mt(o,m)-(1/4*c_3mt(o,m)*tand(twist_a));
z3_m(o,m) = ((tand(phi))*(m*b)/(2*N))+
((z3_mt(o,m)/y3_mt(o,m))*y3_m(o,m)) + c_3m(o,m)*((-
0.7825*((x3_mz(o,m)/c_3m(o,m))^6))+2.2021*((x3_mz(o,m)/c_3m(o,m))^5))
-
(2.3763*((x3_mz(o,m)/c_3m(o,m))^4))+1.2458*((x3_mz(o,m)/c_3m(o,m))^3))
-(0.3676*((x3_mz(o,m)/c_3m(o,m))^2))+0.0782*((x3_mz(o,m)/c_3m(o,m))))+
0.0001));

```

```

end
end

```

```

for o=1:P
for m=1:N
%Control Point Coordinates
y_M(o,m) = (((2*m)-1)*b)/(4*N);
y_Mt(o,m) = b/2);
le_M(o,m) = (tan((sweep_a*(pi/180)))*y_M(o,m));
le_Mt(o,m) = (tan((sweep_a*(pi/180)))*y_Mt(o,m));
te_M(o,m) =
(2/b)*(((tan(sweep_a*(pi/180)))*(b/2)+((2*S)/(b*(1+lambda))))*(lambda-
1))*y_M(o,m)+((2*S)/(b*(1+lambda)))*(b/2));
te_Mt(o,m)=
(2/b)*(((tan(sweep_a*(pi/180)))*(b/2)+((2*S)/(b*(1+lambda))))*(lambda-
1))*y_Mt(o,m)+((2*S)/(b*(1+lambda)))*(b/2));
chord_M(o,m) = te_M(o,m)-le_M(o,m);
chord_Mt(o,m) = te_Mt(o,m)-le_Mt(o,m);
x1_M(o,m) = ((o-1)*(chord_M(o,m)/P));
x2_M(o,m) = ((o)*(chord_M(o,m)/P));
z1_M(o,m) = ((tand(phi))*(((2*m)-1)*b)/(4*N))+
chord_M(o,m)*((-
0.7825*((x1_M(o,m)/chord_M(o,m))^6))+2.2021*((x1_M(o,m)/chord_M(o,m))^
5)) -
(2.3763*((x1_M(o,m)/chord_M(o,m))^4))+1.2458*((x1_M(o,m)/chord_M(o,m)
^3))-
(0.3676*((x1_M(o,m)/chord_M(o,m))^2))+0.0782*((x1_M(o,m)/chord_M(o,m)
))+0.0001));
z2_M(o,m) = ((tand(phi))*(((2*m)-1)*b)/(4*N))+
chord_M(o,m)*((-

```

```

0.7825*((x2_M(o,m)/chord_M(o,m))^6)+(2.2021*((x2_M(o,m)/chord_M(o,m))^
5) -
(2.3763*((x2_M(o,m)/chord_M(o,m))^4)+(1.2458*((x2_M(o,m)/chord_M(o,m)
^3))-
(0.3676*((x2_M(o,m)/chord_M(o,m))^2)+(0.0782*((x2_M(o,m)/chord_M(o,m)
))+ 0.0001));
x_M(o,m) = ((3/4+(o-1))*(chord_M(o,m)/P))+le_M(o,m);
x_Mt(o,m) = ((3/4+(o-1))*(chord_Mt(o,m)/P));
x_Mz(o,m) = ((3/4+(o-1))*(chord_M(o,m)/P));
z_Mt(o,m) =
((3/4*(chord_Mt(o,m)*tand(twist_a))+(1/4*chord_Mt(o,m)*tand(twist_a)))/
(chord_Mt(o,m)))*x_Mt(o,m)-(1/4*chord_Mt(o,m)*tand(twist_a));
z_M(o,m) = (((tand(phi))*(((2*m)-
1)*b)/(4*N)))+((z_Mt(o,m)/y_Mt(o,m))*y_M(o,m))+ (((z2_M(o,m)-
z1_M(o,m))/(x2_M(o,m)-x1_M(o,m)))*x_Mz(o,m)+(z1_M(o,m)-(((z2_M(o,m)-
z1_M(o,m))/(x2_M(o,m)-x1_M(o,m)))*x1_M(o,m)))));
yp_M(o,m) = -(((2*m)-1)*b)/(4*N);

end
end

x_m = size(N*P,1);
y_m = size(N*P,1);
z_m = size(N*P,1);
yp_m = size(N*P,1);

```

```

for m=1:N

```

```

x_m(m,1) = x_M(1,m);
x_m(m+N,1) = x_M(2,m);
x_m(m+2*N,1) = x_M(3,m);
x_m(m+3*N,1) = x_M(4,m);
x_m(m+4*N,1) = x_M(5,m);
x_m(m+5*N,1) = x_M(6,m);
x_m(m+6*N,1) = x_M(7,m);
x_m(m+7*N,1) = x_M(8,m);
y_m(m,1) = y_M(1,m);
y_m(m+N,1) = y_M(2,m);
y_m(m+2*N,1) = y_M(3,m);
y_m(m+3*N,1) = y_M(4,m);
y_m(m+4*N,1) = y_M(5,m);
y_m(m+5*N,1) = y_M(6,m);
y_m(m+6*N,1) = y_M(7,m);
y_m(m+7*N,1) = y_M(8,m);
z_m(m,1) = z_M(1,m);
z_m(m+N,1) = z_M(2,m);
z_m(m+2*N,1) = z_M(3,m);
z_m(m+3*N,1) = z_M(4,m);
z_m(m+4*N,1) = z_M(5,m);
z_m(m+5*N,1) = z_M(6,m);
z_m(m+6*N,1) = z_M(7,m);
z_m(m+7*N,1) = z_M(8,m);
yp_m(m,1) = yp_M(1,m);
yp_m(m+N,1) = yp_M(2,m);
yp_m(m+2*N,1) = yp_M(3,m);

```



```

yp_m(m+3*N,1) = yp_M(4,m);
yp_m(m+4*N,1) = yp_M(5,m);
yp_m(m+5*N,1) = yp_M(6,m);
yp_m(m+6*N,1) = yp_M(7,m);
yp_m(m+7*N,1) = yp_M(8,m);

```

```
end
```

```
for o=1:P
```

```
for n=1:N
```

```
%Panel Coordinates: Bound Vortex
```

```

y1_N(o,n) = ((n-1)*b)/(2*N);
y1_Nt(o,n) = b/2;
le_N1(o,n) = (tan((sweep_a*(pi/180)))*y1_N(o,n));
le_N1t(o,n) = (tan((sweep_a*(pi/180)))*y1_Nt(o,n));
te_N1(o,n) =
(2/b)*((tan(sweep_a*(pi/180)))*(b/2)+((2*S)/(b*(1+lambda)))*(lambda-
1))*y1_N(o,n)+((2*S)/(b*(1+lambda)))*(b/2));
te_N1t(o,n)=
(2/b)*((tan(sweep_a*(pi/180)))*(b/2)+((2*S)/(b*(1+lambda)))*(lambda-
1))*y1_Nt(o,n)+((2*S)/(b*(1+lambda)))*(b/2));
c_1(o,n) = te_N1(o,n)-le_N1(o,n);
c_1Nt(o,n) = te_N1t(o,n)-le_N1t(o,n);
x1_1N(o,n) = ((o-1)*(c_1(o,n)/P));
x2_1N(o,n) = ((o)*(c_1(o,n)/P));
z1_1N(o,n) = (((tand(phi))*(n-1)*b)/(2*N))+ c_1(o,n)*((-
0.7825*((x1_1N(o,n)/c_1(o,n))^6)+(2.2021*((x1_1N(o,n)/c_1(o,n))^5) -
(2.3763*((x1_1N(o,n)/c_1(o,n))^4)+(1.2458*((x1_1N(o,n)/c_1(o,n))^3))-
(0.3676*((x1_1N(o,n)/c_1(o,n))^2)+(0.0782*((x1_1N(o,n)/c_1(o,n))))+
0.0001));
z2_1N(o,n) = (((tand(phi))*(n-1)*b)/(2*N))+ c_1(o,n)*((-
0.7825*((x2_1N(o,n)/c_1(o,n))^6)+(2.2021*((x2_1N(o,n)/c_1(o,n))^5) -
(2.3763*((x2_1N(o,n)/c_1(o,n))^4)+(1.2458*((x2_1N(o,n)/c_1(o,n))^3))-
(0.3676*((x2_1N(o,n)/c_1(o,n))^2)+(0.0782*((x2_1N(o,n)/c_1(o,n))))+
0.0001));
x1_N(o,n) = ((1/4+(o-1))*(c_1(o,n)/P))+le_N1(o,n);
x1_Nt(o,n) = ((1/4+(o-1))*(c_1Nt(o,n)/P));
x1_Nz(o,n) = ((1/4+(o-1))*(c_1(o,n)/P));
z1_Nt(o,n) =
((3/4*(c_1Nt(o,n)*tand(twist_a))+1/4*c_1Nt(o,n)*tand(twist_a))/(c_1Nt
(o,n))*x1_Nt(o,n)-(1/4*c_1Nt(o,n)*tand(twist_a));
z1_N(o,n) = (((tand(phi))*((n-
1)*b)/(2*N)))+((z1_Nt(o,n)/y1_Nt(o,n))*y1_N(o,n)) + ( ((z2_1N(o,n)-
z1_1N(o,n))/(x2_1N(o,n)-x1_1N(o,n)))*x1_Nz(o,n)+(z1_1N(o,n)-
((z2_1N(o,n)-z1_1N(o,n))/(x2_1N(o,n)-x1_1N(o,n)))*x1_1N(o,n))));

y2_N(o,n) = (n*b)/(2*N);
y2_Nt(o,n) = b/2;
le_N2(o,n) = (tan((sweep_a*(pi/180)))*y2_N(o,n));
le_N2t(o,n) = (tan((sweep_a*(pi/180)))*y2_Nt(o,n));
te_N2(o,n) =
(2/b)*((tan(sweep_a*(pi/180)))*(b/2)+((2*S)/(b*(1+lambda)))*(lambda-
1))*y2_N(o,n)+((2*S)/(b*(1+lambda)))*(b/2));

```

```

        te_N2t(o,n)=
(2/b)*(((tan(sweep_a*(pi/180)))*(b/2)+((2*S)/(b*(1+lambda)))*(lambda-
1))*y2_Nt(o,n)+((2*S)/(b*(1+lambda)))*(b/2));
        c_2(o,n) = te_N2(o,n)-le_N2(o,n);
        c_2Nt(o,n) = te_N2t(o,n)-le_N2t(o,n);
        x1_2N(o,n) = ((o-1)*(c_2(o,n)/P));
        x2_2N(o,n) = ((o)*(c_2(o,n)/P));
        z1_2N(o,n) = (((tand(phi))*((n-1)*b)/(2*N)))+ c_2(o,n)*((-
0.7825*((x1_2N(o,n)/c_2(o,n))^6)+(2.2021*((x1_2N(o,n)/c_2(o,n))^5)) -
(2.3763*((x1_2N(o,n)/c_2(o,n))^4)+(1.2458*((x1_2N(o,n)/c_2(o,n))^3))-
(0.3676*((x1_2N(o,n)/c_2(o,n))^2)+(0.0782*((x1_2N(o,n)/c_2(o,n))))+
0.0001));
        z2_2N(o,n) = (((tand(phi))*((n-1)*b)/(2*N)))+ c_2(o,n)*((-
0.7825*((x2_2N(o,n)/c_2(o,n))^6)+(2.2021*((x2_2N(o,n)/c_2(o,n))^5)) -
(2.3763*((x2_2N(o,n)/c_2(o,n))^4)+(1.2458*((x2_2N(o,n)/c_2(o,n))^3))-
(0.3676*((x2_2N(o,n)/c_2(o,n))^2)+(0.0782*((x2_2N(o,n)/c_2(o,n))))+
0.0001));
        x2_N(o,n) = ((1/4+(o-1))*(c_2(o,n)/P))+le_N2(o,n);
        x2_Nt(o,n) = ((1/4+(o-1))*(c_2Nt(o,n)/P));
        x2_Nz(o,n) = ((1/4+(o-1))*(c_2(o,n)/P));
        z2_Nt(o,n) =
((3/4*(c_2Nt(o,n)*tand(twist_a)))+(1/4*c_2Nt(o,n)*tand(twist_a)))/(c_2Nt
(o,n))*x2_Nt(o,n)-(1/4*c_2Nt(o,n)*tand(twist_a));
        z2_N(o,n) =
(((tand(phi))*((n*b)/(2*N)))+((z2_Nt(o,n)/y2_Nt(o,n))*y2_N(o,n)) + (
((z2_2N(o,n)-z1_2N(o,n))/(x2_2N(o,n)-
x1_2N(o,n)))*x2_Nz(o,n)+(z1_2N(o,n)-((z2_2N(o,n)-
z1_2N(o,n))/(x2_2N(o,n)-x1_2N(o,n)))*x1_2N(o,n)))));

        %Panel Coordinates: Bound Vortex (port side change)
        y1p_N(o,n) = -(((n-1)*b)/(2*N));
        y2p_N(o,n) = -(n*b)/(2*N);

    end
end

x1_n = size(N*P,1);
y1_n = size(N*P,1);
z1_n = size(N*P,1);
y1p_n = size(N*P,1);
x2_n = size(N*P,1);
y2_n = size(N*P,1);
z2_n = size(N*P,1);
y2p_n = size(N*P,1);

for n=1:N

        x1_n(n,1) = x1_N(1,n);
        x1_n(n+N,1) = x1_N(2,n);
        x1_n(n+2*N,1) = x1_N(3,n);
        x1_n(n+3*N,1) = x1_N(4,n);
        x1_n(n+4*N,1) = x1_N(5,n);

```

$x1_n(n+5*N, 1) = x1_N(6, n);$
 $x1_n(n+6*N, 1) = x1_N(7, n);$
 $x1_n(n+7*N, 1) = x1_N(8, n);$

$y1_n(n, 1) = y1_N(1, n);$
 $y1_n(n+N, 1) = y1_N(2, n);$
 $y1_n(n+2*N, 1) = y1_N(3, n);$
 $y1_n(n+3*N, 1) = y1_N(4, n);$
 $y1_n(n+4*N, 1) = y1_N(5, n);$
 $y1_n(n+5*N, 1) = y1_N(6, n);$
 $y1_n(n+6*N, 1) = y1_N(7, n);$
 $y1_n(n+7*N, 1) = y1_N(8, n);$

$z1_n(n, 1) = z1_N(1, n);$
 $z1_n(n+N, 1) = z1_N(2, n);$
 $z1_n(n+2*N, 1) = z1_N(3, n);$
 $z1_n(n+3*N, 1) = z1_N(4, n);$
 $z1_n(n+4*N, 1) = z1_N(5, n);$
 $z1_n(n+5*N, 1) = z1_N(6, n);$
 $z1_n(n+6*N, 1) = z1_N(7, n);$
 $z1_n(n+7*N, 1) = z1_N(8, n);$

$y1p_n(n, 1) = y1p_N(1, n);$
 $y1p_n(n+N, 1) = y1p_N(2, n);$
 $y1p_n(n+2*N, 1) = y1p_N(3, n);$
 $y1p_n(n+3*N, 1) = y1p_N(4, n);$
 $y1p_n(n+4*N, 1) = y1p_N(5, n);$
 $y1p_n(n+5*N, 1) = y1p_N(6, n);$
 $y1p_n(n+6*N, 1) = y1p_N(7, n);$
 $y1p_n(n+7*N, 1) = y1p_N(8, n);$

$x2_n(n, 1) = x2_N(1, n);$
 $x2_n(n+N, 1) = x2_N(2, n);$
 $x2_n(n+2*N, 1) = x2_N(3, n);$
 $x2_n(n+3*N, 1) = x2_N(4, n);$
 $x2_n(n+4*N, 1) = x2_N(5, n);$
 $x2_n(n+5*N, 1) = x2_N(6, n);$
 $x2_n(n+6*N, 1) = x2_N(7, n);$
 $x2_n(n+7*N, 1) = x2_N(8, n);$

$y2_n(n, 1) = y2_N(1, n);$
 $y2_n(n+N, 1) = y2_N(2, n);$
 $y2_n(n+2*N, 1) = y2_N(3, n);$
 $y2_n(n+3*N, 1) = y2_N(4, n);$
 $y2_n(n+4*N, 1) = y2_N(5, n);$
 $y2_n(n+5*N, 1) = y2_N(6, n);$
 $y2_n(n+6*N, 1) = y2_N(7, n);$
 $y2_n(n+7*N, 1) = y2_N(8, n);$

$z2_n(n, 1) = z2_N(1, n);$
 $z2_n(n+N, 1) = z2_N(2, n);$
 $z2_n(n+2*N, 1) = z2_N(3, n);$
 $z2_n(n+3*N, 1) = z2_N(4, n);$
 $z2_n(n+4*N, 1) = z2_N(5, n);$
 $z2_n(n+5*N, 1) = z2_N(6, n);$
 $z2_n(n+6*N, 1) = z2_N(7, n);$

```

z2_n(n+7*N,1) = z2_N(8,n);

y2p_n(n,1) = y2p_N(1,n);
y2p_n(n+N,1) = y2p_N(2,n);
y2p_n(n+2*N,1) = y2p_N(3,n);
y2p_n(n+3*N,1) = y2p_N(4,n);
y2p_n(n+4*N,1) = y2p_N(5,n);
y2p_n(n+5*N,1) = y2p_N(6,n);
y2p_n(n+6*N,1) = y2p_N(7,n);
y2p_n(n+7*N,1) = y2p_N(8,n);

end

for m =1:N

    A1_m(:, :, m) = [x2_m(1,m)-x1_m(1,m) y2_m(1,m)-y1_m(1,m) z2_m(1,m)-
z1_m(1,m)];
    A2_m(:, :, m) = [x2_m(2,m)-x1_m(2,m) y2_m(2,m)-y1_m(2,m) z2_m(2,m)-
z1_m(2,m)];
    A3_m(:, :, m) = [x2_m(3,m)-x1_m(3,m) y2_m(3,m)-y1_m(3,m) z2_m(3,m)-
z1_m(3,m)];
    A4_m(:, :, m) = [x2_m(4,m)-x1_m(4,m) y2_m(4,m)-y1_m(4,m) z2_m(4,m)-
z1_m(4,m)];
    A5_m(:, :, m) = [x2_m(5,m)-x1_m(5,m) y2_m(5,m)-y1_m(5,m) z2_m(5,m)-
z1_m(5,m)];
    A6_m(:, :, m) = [x2_m(6,m)-x1_m(6,m) y2_m(6,m)-y1_m(6,m) z2_m(6,m)-
z1_m(6,m)];
    A7_m(:, :, m) = [x2_m(7,m)-x1_m(7,m) y2_m(7,m)-y1_m(7,m) z2_m(7,m)-
z1_m(7,m)];
    A8_m(:, :, m) = [x2_m(8,m)-x1_m(8,m) y2_m(8,m)-y1_m(8,m) z2_m(8,m)-
z1_m(8,m)];

    B1_m(:, :, m) = [x3_m(1,m)-x1_m(1,m) y3_m(1,m)-y1_m(1,m) z3_m(1,m)-
z1_m(1,m)];
    B2_m(:, :, m) = [x3_m(2,m)-x1_m(2,m) y3_m(2,m)-y1_m(2,m) z3_m(2,m)-
z1_m(2,m)];
    B3_m(:, :, m) = [x3_m(3,m)-x1_m(3,m) y3_m(3,m)-y1_m(3,m) z3_m(3,m)-
z1_m(3,m)];
    B4_m(:, :, m) = [x3_m(4,m)-x1_m(4,m) y3_m(4,m)-y1_m(4,m) z3_m(4,m)-
z1_m(4,m)];
    B5_m(:, :, m) = [x3_m(5,m)-x1_m(5,m) y3_m(5,m)-y1_m(5,m) z3_m(5,m)-
z1_m(5,m)];
    B6_m(:, :, m) = [x3_m(6,m)-x1_m(6,m) y3_m(6,m)-y1_m(6,m) z3_m(6,m)-
z1_m(6,m)];
    B7_m(:, :, m) = [x3_m(7,m)-x1_m(7,m) y3_m(7,m)-y1_m(7,m) z3_m(7,m)-
z1_m(7,m)];
    B8_m(:, :, m) = [x3_m(8,m)-x1_m(8,m) y3_m(8,m)-y1_m(8,m) z3_m(8,m)-
z1_m(8,m)];

    n1_m(:, :, m) = cross(A1_m(:, :, m), B1_m(:, :, m));
    n2_m(:, :, m) = cross(A2_m(:, :, m), B2_m(:, :, m));
    n3_m(:, :, m) = cross(A3_m(:, :, m), B3_m(:, :, m));
    n4_m(:, :, m) = cross(A4_m(:, :, m), B4_m(:, :, m));
    n5_m(:, :, m) = cross(A5_m(:, :, m), B5_m(:, :, m));
    n6_m(:, :, m) = cross(A6_m(:, :, m), B6_m(:, :, m));
    n7_m(:, :, m) = cross(A7_m(:, :, m), B7_m(:, :, m));

```

```

n8_m(:, :, m) = cross(A8_m(:, :, m), B8_m(:, :, m));

n1_unit(:, :, m) = (n1_m(:, :, m) ./ (sqrt((n1_m(1, 1, m)^2) + (n1_m(1, 2, m)^2) + (n1_m(1, 3, m)^2))));

n2_unit(:, :, m) = (n2_m(:, :, m) ./ (sqrt((n2_m(1, 1, m)^2) + (n2_m(1, 2, m)^2) + (n2_m(1, 3, m)^2))));

n3_unit(:, :, m) = (n3_m(:, :, m) ./ (sqrt((n3_m(1, 1, m)^2) + (n3_m(1, 2, m)^2) + (n3_m(1, 3, m)^2))));

n4_unit(:, :, m) = (n4_m(:, :, m) ./ (sqrt((n4_m(1, 1, m)^2) + (n4_m(1, 2, m)^2) + (n4_m(1, 3, m)^2))));

n5_unit(:, :, m) = (n5_m(:, :, m) ./ (sqrt((n5_m(1, 1, m)^2) + (n5_m(1, 2, m)^2) + (n5_m(1, 3, m)^2))));

n6_unit(:, :, m) = (n6_m(:, :, m) ./ (sqrt((n6_m(1, 1, m)^2) + (n6_m(1, 2, m)^2) + (n6_m(1, 3, m)^2))));

n7_unit(:, :, m) = (n7_m(:, :, m) ./ (sqrt((n7_m(1, 1, m)^2) + (n7_m(1, 2, m)^2) + (n7_m(1, 3, m)^2))));

n8_unit(:, :, m) = (n8_m(:, :, m) ./ (sqrt((n8_m(1, 1, m)^2) + (n8_m(1, 2, m)^2) + (n8_m(1, 3, m)^2))));

```

```

end

```

```

%Vortex Velocity Definition in a typical horseshoe vortex

```

```

% Starboard Wing

```

```

Fac_12_i = size(1, N*P);
Fac_12_j = size(1, N*P);
Fac_12_k = size(1, N*P);
Fac2_12 = size(1, N*P);

```

```

V_12 = size(1, N*P);

```

```

V1_inf_i = size(1, N*P);
V1_inf_j = size(1, N*P);
V1_inf_k = size(1, N*P);
V1_inf = size(1, N*P);

```

```

V2_inf_i = size(1, N*P);
V2_inf_j = size(1, N*P);
V2_inf_k = size(1, N*P);
V2_inf = size(1, N*P);

```

```

V_s = size(1, N*P);

```

```

%Port Wing
Facp_12_i = size(1,N*P);
Facp_12_j = size(1,N*P);
Facp_12_k = size(1,N*P);
Fac2p_12 = size(1,N*P);

V_12p = size (1,N*P);

V1p_inf_i = size(1,N*P);
V1p_inf_j = size(1,N*P);
V1p_inf_k = size(1,N*P);
V1p_inf = size (1,N*P);

V2p_inf_i = size(1,N*P);
V2p_inf_j = size(1,N*P);
V2p_inf_k = size(1,N*P);
V2p_inf = size (1,N*P);

V_p = size(1,N*P);

% Downwash Velocity function definition

w_n = size (1,N*P);

% Boundary Condition function definition

BC = size (1,N*P);
chord = size (1,N*P);

% Trailing Vortices for Induced Drag Calculation

Vt_s = size(1,N*P);
Vt_p = size(1,N*P);
w_i = size(1,N*P);

% Main Program Loop for VLM analysis

for m=1:N*P
    for n=1:N*P

        %STARBOARD WING
        % Primary Vector Bound Vortex (induced velocity calculation)
        Fac_12_i(m,n) = ((y_m(m)-y1_n(n))*(z_m(m)-z2_n(n))-(y_m(m)-
y2_n(n))*(z_m(m)-z1_n(n)))/(((y_m(m)-y1_n(n))*(z_m(m)-z2_n(n))-
(y_m(m)-y2_n(n))*(z_m(m)-z1_n(n)))^2)+(((x_m(m)-x1_n(n))*(z_m(m)-
z2_n(n))-(x_m(m)-x2_n(n))...
*(z_m(m)-z1_n(n)))^2)+(((x_m(m)-x1_n(n))*(y_m(m)-y2_n(n))-
(x_m(m)-x2_n(n))*(y_m(m)-y1_n(n)))^2));

        Fac_12_j(m,n) = -((x_m(m)-x1_n(n))*(z_m(m)-z2_n(n))-(x_m(m)-
x2_n(n))*(z_m(m)-z1_n(n)))/(((y_m(m)-y1_n(n))*(z_m(m)-z2_n(n))-
(y_m(m)-y2_n(n))*(z_m(m)-z1_n(n)))^2)+(((x_m(m)-x1_n(n))*(z_m(m)-
z2_n(n))-(x_m(m)-x2_n(n))...

```

```

* (z_m(m)-z1_n(n)) ^2)+(((x_m(m)-x1_n(n)) * (y_m(m)-y2_n(n)) -
(x_m(m)-x2_n(n)) * (y_m(m)-y1_n(n))) ^2));

```

```

Fac_12_k(m,n) = ((x_m(m)-x1_n(n)) * (y_m(m)-y2_n(n)) - (x_m(m)-
x2_n(n)) * (y_m(m)-y1_n(n))) / (((y_m(m)-y1_n(n)) * (z_m(m)-z2_n(n)) -
(y_m(m)-y2_n(n)) * (z_m(m)-z1_n(n))) ^2)+(((x_m(m)-x1_n(n)) * (z_m(m)-
z2_n(n)) - (x_m(m)-x2_n(n)) ...
* (z_m(m)-z1_n(n)) ^2)+(((x_m(m)-x1_n(n)) * (y_m(m)-y2_n(n)) -
(x_m(m)-x2_n(n)) * (y_m(m)-y1_n(n))) ^2));

```

```

Fac2_12(m,n) = (((x2_n(n)-x1_n(n)) * (x_m(m)-x1_n(n)) + (y2_n(n)-
y1_n(n)) * (y_m(m)-y1_n(n)) + (z2_n(n)-z1_n(n)) * (z_m(m)-
z1_n(n))) / (((x_m(m)-x1_n(n)) ^2)+((y_m(m)-y1_n(n)) ^2)+((z_m(m)-
z1_n(n)) ^2)) ^0.5)) ...
- (((x2_n(n)-x1_n(n)) * (x_m(m)-x2_n(n)) + (y2_n(n)-
y1_n(n)) * (y_m(m)-y2_n(n)) + (z2_n(n)-z1_n(n)) * (z_m(m)-
z2_n(n))) / (((x_m(m)-x2_n(n)) ^2)+((y_m(m)-y2_n(n)) ^2)+((z_m(m)-
z2_n(n)) ^2)) ^0.5));

```

```

V_12(m,n) = ((Fac_12_i(m,n) + Fac_12_j(m,n) + Fac_12_k(m,n)) *
Fac2_12(m,n));

```

```

% Secondary Vectors (induced velocity calculation) Trailing
Vortices

```

```

V1_inf_i(m,n) = 0;
V1_inf_j(m,n) = ((z_m(m)-z1_n(n)) / (((z_m(m)-
z1_n(n)) ^2)+((y1_n(n)-y_m(m)) ^2))) * (1+((x_m(m)-x1_n(n)) / (((x_m(m)-
x1_n(n)) ^2)+((y_m(m)-y1_n(n)) ^2)+((z_m(m)-z1_n(n)) ^2)) ^0.5));
V1_inf_k(m,n) = ((y1_n(n)-y_m(m)) / (((z_m(m)-
z1_n(n)) ^2)+((y1_n(n)-y_m(m)) ^2))) * (1+((x_m(m)-x1_n(n)) / (((x_m(m)-
x1_n(n)) ^2)+((y_m(m)-y1_n(n)) ^2)+((z_m(m)-z1_n(n)) ^2)) ^0.5));

```

```

V1_inf(m,n) = (V1_inf_i(m,n) + V1_inf_j(m,n) + V1_inf_k(m,n));

```

```

V2_inf_i(m,n) = 0;
V2_inf_j(m,n) = -((z_m(m)-z2_n(n)) / (((z_m(m)-
z2_n(n)) ^2)+((y2_n(n)-y_m(m)) ^2))) * (1+((x_m(m)-x2_n(n)) / (((x_m(m)-
x2_n(n)) ^2)+((y_m(m)-y2_n(n)) ^2)+((z_m(m)-z2_n(n)) ^2)) ^0.5));
V2_inf_k(m,n) = -((y2_n(n)-y_m(m)) / (((z_m(m)-
z2_n(n)) ^2)+((y2_n(n)-y_m(m)) ^2))) * (1+((x_m(m)-x2_n(n)) / (((x_m(m)-
x2_n(n)) ^2)+((y_m(m)-y2_n(n)) ^2)+((z_m(m)-z2_n(n)) ^2)) ^0.5));

```

```

V2_inf(m,n) = (V2_inf_i(m,n) + V2_inf_j(m,n) + V2_inf_k(m,n));

```

```

% Induced Downwash by trailing vortices

```

```

Vt_s(m,n) = V1_inf(m,n) + V2_inf(m,n);

```

```

% Total Induced Velocity at control point (x, y, z) Starboard
Wing

```

```

V_s(m,n) = V_12(m,n) + V1_inf(m,n) + V2_inf(m,n);

```

%PORT WING

% Primary Vector Bound Vortex (induced velocity calculation)

```
Facp_12_i(m,n) = ((y_m(m)-y1p_n(n))*(z_m(m)-z2_n(n))-(y_m(m)-y2p_n(n))*(z_m(m)-z1_n(n)))/(((y_m(m)-y1p_n(n))*(z_m(m)-z2_n(n))-(y_m(m)-y2p_n(n))*(z_m(m)-z1_n(n)))^2)+(((x_m(m)-x1_n(n))*(z_m(m)-z2_n(n))-(x_m(m)-x2_n(n))...*(z_m(m)-z1_n(n)))^2)+(((x_m(m)-x1_n(n))*(y_m(m)-y2p_n(n))-(x_m(m)-x2_n(n))*(y_m(m)-y1p_n(n)))^2));
```

```
Facp_12_j(m,n) = -((x_m(m)-x1_n(n))*(z_m(m)-z2_n(n))-(x_m(m)-x2_n(n))*(z_m(m)-z1_n(n)))/(((y_m(m)-y1p_n(n))*(z_m(m)-z2_n(n))-(y_m(m)-y2p_n(n))*(z_m(m)-z1_n(n)))^2)+(((x_m(m)-x1_n(n))*(z_m(m)-z2_n(n))-(x_m(m)-x2_n(n))...*(z_m(m)-z1_n(n)))^2)+(((x_m(m)-x1_n(n))*(y_m(m)-y2p_n(n))-(x_m(m)-x2_n(n))*(y_m(m)-y1p_n(n)))^2));
```

```
Facp_12_k(m,n) = ((x_m(m)-x1_n(n))*(y_m(m)-y2p_n(n))-(x_m(m)-x2_n(n))*(y_m(m)-y1p_n(n)))/(((y_m(m)-y1p_n(n))*(z_m(m)-z2_n(n))-(y_m(m)-y2p_n(n))*(z_m(m)-z1_n(n)))^2)+(((x_m(m)-x1_n(n))*(z_m(m)-z2_n(n))-(x_m(m)-x2_n(n))...*(z_m(m)-z1_n(n)))^2)+(((x_m(m)-x1_n(n))*(y_m(m)-y2p_n(n))-(x_m(m)-x2_n(n))*(y_m(m)-y1p_n(n)))^2));
```

```
Fac2p_12(m,n) = (((x2_n(n)-x1_n(n))*(x_m(m)-x1_n(n)))+(y2p_n(n)-y1p_n(n))*(y_m(m)-y1p_n(n))+(z2_n(n)-z1_n(n))*(z_m(m)-z1_n(n)))/(((x_m(m)-x1_n(n))^2)+((y_m(m)-y1p_n(n))^2)+((z_m(m)-z1_n(n))^2))^0.5)... - (((x2_n(n)-x1_n(n))*(x_m(m)-x2_n(n)))+(y2p_n(n)-y1p_n(n))*(y_m(m)-y2p_n(n)))+(z2_n(n)-z1_n(n))*(z_m(m)-z2_n(n)))/(((x_m(m)-x2_n(n))^2)+((y_m(m)-y2p_n(n))^2)+((z_m(m)-z2_n(n))^2))^0.5));
```

```
V_12p(m,n) = ((Facp_12_i(m,n) + Facp_12_j(m,n) + Facp_12_k(m,n)) * Fac2p_12(m,n));
```

% Secondary Vectors (induced velocity calculation) Trailing vortices

```
V1p_inf_i(m,n) = 0;  
V1p_inf_j(m,n) = ((z_m(m)-z1_n(n))/(((z_m(m)-z1_n(n))^2)+(y1p_n(n)-y_m(m))^2))* (1+((x_m(m)-x1_n(n))/(((x_m(m)-x1_n(n))^2)+((y_m(m)-y1p_n(n))^2)+((z_m(m)-z1_n(n))^2))^0.5));  
V1p_inf_k(m,n) = ((y1p_n(n)-y_m(m))/(((z_m(m)-z1_n(n))^2)+(y1p_n(n)-y_m(m))^2))* (1+((x_m(m)-x1_n(n))/(((x_m(m)-x1_n(n))^2)+((y_m(m)-y1p_n(n))^2)+((z_m(m)-z1_n(n))^2))^0.5));
```

```
V1p_inf(m,n) = (V1p_inf_i(m,n) + V1p_inf_j(m,n) + V1p_inf_k(m,n));
```

```
V2p_inf_i(m,n) = 0;  
V2p_inf_j(m,n) = -((z_m(m)-z2_n(n))/(((z_m(m)-z2_n(n))^2)+(y2p_n(n)-y_m(m))^2))* (1+((x_m(m)-x2_n(n))/(((x_m(m)-x2_n(n))^2)+((y_m(m)-y2p_n(n))^2)+((z_m(m)-z2_n(n))^2))^0.5));
```



```

        V2p_inf_k(m,n) = -((y2p_n(n)-y_m(m))/((z_m(m)-
z2_n(n))^2)+(y2p_n(n)-y_m(m))^2))* (1+((x_m(m)-x2_n(n))/((x_m(m)-
x2_n(n))^2)+((y_m(m)-y2p_n(n))^2)+((z_m(m)-z2_n(n))^2))^0.5));

        V2p_inf(m,n) = (V2p_inf_i(m,n) + V2p_inf_j(m,n) +
V2p_inf_k(m,n));

        % Induced Downwash by trailing vortices

        Vt_p(m,n) = -1*(V1p_inf(m,n)+ V2p_inf(m,n));

        % Total induced downwash velocity by port and starboard wing
trailing
        % vortices

        w_i (m,n) = Vt_s(m,n) + Vt_p(m,n);

        % Total Induced Velocity at control point (x, y, z) by the Port
Wing

        V_p(m,n) = -1*(V_l2p(m,n) + V1p_inf(m,n) + V2p_inf(m,n));

        %Total Induced Velocity at control point by starboard wing and
port wing

        w_n(m,n) = V_s(m,n) + V_p(m,n);

        end

    end

    angle_attack =-3 ;
    for i = 1:1:22
        angle_attack =angle_attack+1;
        store_angle_attack(i,1)= angle_attack;

        U_inf = U_input - 10;
        for f =1:1:50
            U_inf = U_inf+10;
            store_U_inf(f,1)= U_inf;

            for m =1:N

                v1_m(:, :,m) = [U_inf*(cosd(angle_attack)) 0
U_inf*(sind(angle_attack))];
                n_m(1,m)= dot(n1_unit(:, :,m),v1_m(:, :,m));
                n_m(2,m)= dot(n2_unit(:, :,m),v1_m(:, :,m));
                n_m(3,m)= dot(n3_unit(:, :,m),v1_m(:, :,m));
                n_m(4,m)= dot(n4_unit(:, :,m),v1_m(:, :,m));
                n_m(5,m)= dot(n5_unit(:, :,m),v1_m(:, :,m));
                n_m(6,m)= dot(n6_unit(:, :,m),v1_m(:, :,m));
            end
        end
    end

```



```

AOA_To=store_angle_attack;
S_polar_o = store_S2;
b_table_o = store_b2';
U_table = store_U_inf;
lambda_table_o = store_lambda;
store_sweep_o = store_sweep_a;
store_twist_o = store_twist_a;

h2 = waitbar(0,'Please wait... Optimization now in Progress');
U_inf_o = U_input - 10;
for f =1:1:50
    U_inf_o = U_inf_o+10;
    store_U_inf(f,1)= U_inf_o;
    waitbar(f/50)

S = str2double(answer{2,1})-4;
for z = 1:1:3
    S = S+2;
    store_S2(z,1)= S;

    b = str2double(answer{1,1})-2;
    for q = 1:1:3
        b = b+1;
        store_b2(q,1) = b;

        lambda = str2double(answer{3,1})-0.2 ;
        for T = 1:1:3
            lambda = lambda+0.1;
            store_lambda(T,1) = lambda;

            sweep_a =str2double(answer{5,1})-8 ;
            for p = 1:1:4
                sweep_a = sweep_a+2;
                store_sweep_a(p,1)= sweep_a;

                twist_a = str2double(answer{6,1})-3;
                for u = 1:1:3
                    twist_a = twist_a+1;
                    store_twist_a(u,1) = twist_a;

                    Cl_w = (2*W)/(rho*(U_inf_o)^2*S);
                    Cl_slope = ((Cl_p(z,q,T,p,u,10,1)-
Cl_p(z,q,T,p,u,8,1))/(store_angle_attack(10,1)-
store_angle_attack(8,1)));
                    AOA_R = ((Cl_w-(Cl_p(z,q,T,p,u,10,1)-
((Cl_slope)*(store_angle_attack(10,1)))))/(Cl_slope));

                    %Vortex Induced Drag Method 2
                    delta_factor =
interp2(Aspect_ratio,taper_ratio,Delta_var,(b^2/S),lambda);

```

```

e_span = 1/(1+delta_factor);
CDi = Cl_w^2/(pi*(b^2/S)*e_span);
D_i = (0.5*rho*((U_inf_o)^2)*CDi*S);

% Profile Drag Calculation

[Re,cl,area_p]=
total_drag(S,b,lambda,AOA_R,U_inf_o,N,P,phi,sweep_a,twist_a,nu,rho);
sim Lookup_2D_validation;

%Average Chord Calculation and Panel Area

d_n = (0.5*rho*(U_inf_o^2)*((cd_p) .* (area_p)));

D_p = 2*(sum(d_n));

Cd_p = (D_p/(0.5*rho*((U_inf_o)^2)*S));

% Total Drag Calculation

Cd_w = CDi + Cd_p;

% Thrust Required and Power Required

TR = (0.5*rho*(U_inf_o)^2*S*Cd_w);
PR = (TR*U_inf_o)/550;

%Aerodynamic Coefficients for performance evaluation

L_D = Cl_w/Cd_w;
Cl_R = (Cl_w^0.5)/Cd_w;
Cl_R2 = (Cl_w^(3/2))/Cd_w;

% Calculation of Wing Fuel Volume: Gas tank assumed as a tapered
rectangular box

c_root_chord = (2*S)/(b*(1+lambda));
c_tip_chord = lambda*c_root_chord ;
upper_airfoil = (-4.1755*(x_percent)^6) + (13.552*(x_percent)^5) -
(16.892*(x_percent)^4) + (10.37*(x_percent)^3) - (3.6613*(x_percent)^2)
+ (0.7962*(x_percent)) + 0.0094;
lower_airfoil = (2.8139*(x_percent)^6) - (9.4191*(x_percent)^5) +
(11.884*(x_percent)^4) - (7.3145*(x_percent)^3) +
(2.5587*(x_percent)^2) - (0.516*(x_percent)) - 0.0064;

side_b = c_root_chord*(upper_airfoil-lower_airfoil);
side_a = c_tip_chord*(upper_airfoil-lower_airfoil);
K_cons = (x_percent-0.25)/(upper_airfoil-lower_airfoil);

```

```

    Fuel_volume = K_cons*((b/2)/3)*((side_a^2) + (side_a*side_b) +
((side_b^2)));
    Fuel_volume_g = 2*(Fuel_volume*7.4805);
    Fuel_weight = (Fuel_volume_g*Jet_fuel_d);

```

```

%Performance Results

```

```

    TR_result(f,z,q,T,p,u) = TR;
    PR_result(f,z,q,T,p,u) = PR;
    L_D_result(f,z,q,T,p,u) = L_D;
    CL_result(f,z,q,T,p,u) = CL_R;
    Cl_w_result(f,z,q,T,p,u) = Cl_w;
    Cd_w_result(f,z,q,T,p,u) = Cd_w;
    CL_result2(f,z,q,T,p,u) = CL_R2;
    AOA_R_result(f,z,q,T,p,u) = AOA_R;
    Fuel_weight_result(f,z,q,T,p,u) = Fuel_weight;
    Fuel_volume_g_result(f,z,q,T,p,u) = Fuel_volume_g;

```

```

        end
    end
end
end
end
end

```

```

%-----%
% INITIAL WING ANALYSIS AND PERFORMANCE

```

```

S_initial = store_S2(2);
b_initial = store_b2(2);
lambda_initial = store_lambda(2);
Sweep_initial = store_sweep_a(4);
Twist_initial = store_twist_a(3);

```

```

% Best L_D_cruise = A;
graph_L_D_i = L_D_result(:,2,2,2,4,3);
graph_AOA_R_i = AOA_R_result(:,2,2,2,4,3);

```

```

%Aerodynamic Coefficients
graph_CL_result_i = CL_result(:,2,2,2,4,3);
graph_CL_result2_i = CL_result2(:,2,2,2,4,3);

```

```

%Thrust Required Curve
graph_TR_result_i = TR_result(:,2,2,2,4,3);

```

```

%Power Required Curve
graph_PR_result_i = PR_result(:,2,2,2,4,3);

```

```

% Max Lift to drag Ratio @ cruise
[Best_L_D_cruise_i_product,index_i] = max(CL_result(:,2,2,2,4,3));

% Max Lift to Drag Ratio at Cruise velocity

Best_L_D_cruise_i = L_D_result(index_i,2,2,2,4,3);

% Calculation of Cruise Velocity
V_opt_cruise_i = store_U_inf(index_i);

% Calculation of angle of attack at cruise

AOA_m_i = AOA_R_result(index_i,2,2,2,4,3);

% Best Fuel volume:
Best_Fuel_volume_g_i = Fuel_volume_g_result(1,2,2,2,4,3);

% Best Fuel weight
Best_Fuel_weight_i = Fuel_weight_result(1,2,2,2,4,3);

% Calculation of Maximum Range for Best Wing Assuming constant altitude
RANGE_new_i =
((2*3600)/TSFC)*(sqrt(2/(rho*S_initial)))*(Best_L_D_cruise_i_product)*
((W^0.5-(W-Best_Fuel_weight_i)^0.5))/5280);

% Geometric Plot Function
[xl_m_geo_i, yl_m_geo_i, zl_m_geo_i, yl_m_s_geo_i]=
Geometry_VORT_initial(S_initial,b_initial,lambda_initial,N,P,phi,Sweep_
initial,Twist_initial);

% Cl vs Alpha Curve Function
[Cl_result_initial,
store_angle_attack_initial]=i_wing_performance(S_initial,b_initial,lamb
da_initial,V_opt_cruise_i,N,P,phi,Sweep_initial,Twist_initial,nu,rho);

% Section Lift Coefficient Distribution at Cruise Velocity

[cl_f_i,y_b_initial]=
section_lift_initial(S_initial,b_initial,lambda_initial,AOA_m_i,V_opt_c
ruise_i,P,phi,Sweep_initial,Twist_initial,nu,rho);

%-----
----
%Calculation of max sqrtCl/Cd for all wing configurations

for Z_1 =1:z
    for Q_1 =1:q
        for w_1 =1:T
            for X_1=1:p
                for U_1=1:u

                    Aero_ratio_final(Z_1,Q_1,w_1,X_1,U_1) =
(L_D_result(index_i,Z_1,Q_1,w_1,X_1,U_1));

```

```

end
end
end
end
end
-----
% Calculation of Wing Weight based on Statistical Group Weights Method
S = str2double(answer{2,1})-4;
for z_1 = 1:1:3
    S = S+2;
    store_S2(z,1)= S;

    b = str2double(answer{1,1})-2;
    for q_1 = 1:1:3
        b = b+1;
        store_b2(q,1) = b;

        lambda = str2double(answer{3,1})-0.2 ;
        for T_1 = 1:1:3
            lambda = lambda+0.1;
            store_lambda(T,1) = lambda;

            sweep_a =str2double(answer{5,1})-8 ;
            for p_1 = 1:1:4
                sweep_a = sweep_a+2;
                store_sweep_a(p,1)= sweep_a;

                twist_a = str2double(answer{6,1})-3;
                for u_1 = 1:1:3
                    twist_a = twist_a+1;
                    store_twist_a(u,1) = twist_a;

                    sweep_mac = atand(tan((sweep_a*(pi/180)))-
((1+lambda)/((b^2/S)*(1+lambda))));
                    Wing_weight =
(0.036*(S^0.758))*(Fuel_weight_result(1,z_1,q_1,T_1,p_1,u_1)^0.0035)*((
((b^2)/S)/(cosd(sweep_mac))^2)^0.6)*((0.5*rho*V_opt_cruise_i^2)^0.006)*
(lambda^0.04)*((10/cosd(sweep_mac))^-0.3)*(N_z*W)^0.49;
                    Wing_weight_result(z_1,q_1,T_1,p_1,u_1) =
Wing_weight;

                    RANGE_result(z_1,q_1,T_1,p_1,u_1) =
(((2*3600)/TSFC)*(sqrt(2/(rho*S)))*(max(CL_result(:,z_1,q_1,T_1,p_1,u_1
))))*(W^0.5-(W-Fuel_weight_result(1,z_1,q_1,T_1,p_1,u_1))^0.5))/5280);
                    end
                end
            end
        end
    end
end
-----
%Calculation of Performance Index Function

```

```

for Z_2 =1:z
    for Q_2 =1:q
        for w_2 =1:T
            for X_2=1:p
                for U_2=1:u

                    performance_index(Z_2,Q_2,w_2,X_2,U_2) =
0.2*(Wing_weight_result(2,2,2,4,3)/Wing_weight_result(Z_2,Q_2,w_2,X_2,U
_2))+
0.6*(Aero_ratio_final(Z_2,Q_2,w_2,X_2,U_2)/Aero_ratio_final(2,2,2,4,3))
+0.2*(RANGE_result(Z_2,Q_2,w_2,X_2,U_2)/RANGE_new_i);

                end
            end
        end
    end
end

%-----
close(h2)
% OPTIMIZATION ANALYSIS

%OPTIMIZATION FOR BEST L/D,Range & Stall Speed

[A_1,V] = max(max(max(max(max(performance_index)))));
[A_2,B]=max(max(max(max(performance_index(:, :, :, :, V)))));
[A_3,C]=max(max(max(performance_index(:, :, :, B, V)))));
[A_4,D]=max(max(performance_index(:, :, C, B, V)));
[A_5,E]=max(performance_index(:, D, C, B, V));

S_best = store_S2(E);
b_best = store_b2(D);
lambda_best = store_lambda(C);
Sweep_best = store_sweep_a(B);
Twist_best = store_twist_a(V);
% Optimal Aspect Ratio
AR_best = (b_best^2)/(S_best);
% Best L/D cruise = A;
graph_L_D = L_D_result(:,E,D,C,B,V);
graph_AOA_R = AOA_R_result(:,E,D,C,B,V);

%Aerodynamic Coefficients
graph_CL_result = CL_result(:,E,D,C,B,V);
graph_CL_result2 = CL_result2(:,E,D,C,B,V);

%Thrust Required Curve
graph_TR_result = TR_result(:,E,D,C,B,V);

%Power Required Curve

```



```

graph_PR_result = PR_result(:,E,D,C,B,V);

% Max Lift to drag Ratio and velocity Product @ cruise
[Best_L_D_cruise_product,index_opt] = max(CL_result(:,E,D,C,B,V));

% Max Lift to Drag Ratio at Cruise velocity

Best_L_D_cruise = L_D_result(index_opt,E,D,C,B,V);

% Calculation of Cruise Velocity
V_opt_cruise = store_U_inf(index_opt);

% Calculation of angle of attack at cruise

AOA_m = AOA_R_result(index_opt,E,D,C,B,V);

% Best Fuel volume:
Best_Fuel_volume_g = Fuel_volume_g_result(1,E,D,C,B,V);

% Best Fuel weight
Best_Fuel_weight = Fuel_weight_result(1,E,D,C,B,V);

% Calculation of Maximum Range for Best Wing Assuming constant altitude
RANGE_new =
((2*3600)/TSFC)*(sqrt(2/(rho*S_best)))*(Best_L_D_cruise_product)*((W^0
.5- (W-Best_Fuel_weight)^0.5))/5280);

% Geometric Plot Function
[xl_m_geo, yl_m_geo, zl_m_geo, yl_m_s_geo]=
Geometry_VORT_optimal(S_best,b_best,lambda_best,N,P,phi,Sweep_best,Twis
t_best);

% Cl vs Alpha Curve Function
[Cl_result_optimal,
store_angle_attack_opt]=optimal_wing_performance(S_best,b_best,lambda_b
est,V_opt_cruise,N,P,phi,Sweep_best,Twist_best,nu,rho);

% Section Lift Coefficient Distribution at Cruise Velocity
[cl_f_opt,y_b_opt]=
section_lift_optimal(S_best,b_best,lambda_best,AOA_m,V_opt_cruise,P,phi
,Sweep_best,Twist_best,nu,rho);

% Max Lift to Drag Ratio of Optimal Wing at Initial Wing's Cruise
Velocity

Best_L_D_cruise_previous = L_D_result(index_i,E,D,C,B,V);

% Wing Weights Result
Wing_weight_opt = Wing_weight_result(E,D,C,B,V);
Wing_weight_init = Wing_weight_result(2,2,2,4,3);

```

toc

```
fprintf('\n\nGEOMETRIC DESCRIPTION OF WINGS\n\n')

fprintf('\n\nGEOMETRIC PARAMETERS OF THE INITIAL WING\n\n')
fprintf('\nAirfoil Used : NACA 65 - 210 \n')
fprintf('\nWing Area S = %6.3f ft^2\n',S_initial)
fprintf('\nWingspan b = %6.3f ft\n',b_initial)
fprintf('\nTaper Ratio = %6.3f \n',lambda_initial)
fprintf('\nLeading Edge Sweep = %6.3f deg\n',Sweep_initial)
fprintf('\nGeometric Twist = %6.3f deg\n',Twist_initial)
fprintf('\nAspect Ratio AR = %6.3f \n',AR_initial)
fprintf('\n# of Panels Half Span used N =%6.f \n',N)
fprintf('\n# of Panels in the chordwise direction used P =%6.f \n',P)

fprintf('\n\nGEOMETRIC PARAMETERS OF THE OPTIMAL WING\n\n')
fprintf('\nAirfoil Used : NACA 65 - 210 \n')
fprintf('\nWing Area S = %6.3f ft^2\n',S_best)
fprintf('\nWingspan b = %6.3f ft\n',b_best)
fprintf('\nTaper Ratio = %6.3f \n',lambda_best)
fprintf('\nLeading Edge Sweep = %6.3f deg\n',Sweep_best)
fprintf('\nGeometric Twist = %6.3f deg\n',Twist_best)
fprintf('\nAspect Ratio AR = %6.3f \n',AR_best)
fprintf('\n# of Panels Half Span used N =%6.f \n',N)
fprintf('\n# of Panels in the chordwise direction used P =%6.f \n',P)

fprintf('\n\nPERFORMANCE PARAMETERS OF WINGS\n\n')

fprintf('\n\nPERFORMANCE OF INITIAL WING\n\n')
fprintf('\n Lift to Drag Ratio at Cruise Velocity = %6.2f
\n',Best_L_D_cruise_i)
fprintf('\n Lift to Drag Ratio and Cruise Velocity product = %6.2f
\n',Best_L_D_cruise_i_product)
fprintf('\n Cruise Velocity = %6.2f ft/s \n',V_opt_cruise_i)
fprintf('\n Angle of Attack at Cruise Velocity = %6.2f deg \n',AOA_m_i)
fprintf('\n Fuel Volume Carried by Wing = %6.2f gallons
\n',Best_Fuel_volume_g_i)
fprintf('\n Fuel Weight = %6.2f lbs \n',Best_Fuel_weight_i)
fprintf('\n Wing Weight = %6.2f lbs \n',Wing_weight_init)
fprintf('\n Range = %6.2f miles \n',RANGE_new_i)

fprintf('\n\nPERFORMANCE OF OPTIMAL WING\n\n')
fprintf('\n Lift to Drag Ratio at Optimal Cruise Velocity = %6.2f
\n',Best_L_D_cruise)
fprintf('\n Lift to Drag Ratio at Original Cruise Velocity = %6.2f
\n',Best_L_D_cruise_previous)
fprintf('\n Lift to Drag Ratio and Cruise Velocity product = %6.2f
\n',Best_L_D_cruise_product)
fprintf('\n Cruise Velocity = %6.2f ft/s \n',V_opt_cruise)
fprintf('\n Angle of Attack at Cruise Velocity = %6.2f deg \n',AOA_m)
fprintf('\n Fuel Volume Carried by Wing = %6.2f gallons
\n',Best_Fuel_volume_g)
fprintf('\n Fuel Weight = %6.2f lbs \n',Best_Fuel_weight)
fprintf('\n Wing Weight = %6.2f lbs \n',Wing_weight_opt)
```

```

fprintf('\n Range = %6.2f miles \n',RANGE_new)

% Figures
figure (1)
    surf(xl_m_geo,y1_m_geo,z1_m_geo)
    axis([-b_best/4 b_best/2 -b_best/2 b_best/2 -1 1])
    grid on
    xlabel({'x (ft)'},'FontSize',9,'color','r')
    ylabel({'y (ft)'},'FontSize',9,'color','r')
    zlabel({'z (ft)'},'FontSize',9,'color','r')
    title('Vortex Lattice Optimal Wing Configuration','color',
'b','FontSize',10 )
    hold on
    surf(xl_m_geo, y1_m_s_geo,z1_m_geo)
    colormap Bone

figure (2)
    surf(xl_m_geo_i,y1_m_geo_i,z1_m_geo_i)
    axis([-b_initial/4 b_initial/2 -b_initial/2 b_initial/2 -1 1])
    grid on
    xlabel({'x (ft)'},'FontSize',9,'color','r')
    ylabel({'y (ft)'},'FontSize',9,'color','r')
    zlabel({'z (ft)'},'FontSize',9,'color','r')
    title('Vortex Lattice Initial Wing Configuration','color',
'b','FontSize',10 )
    hold on
    surf(xl_m_geo_i, y1_m_s_geo_i,z1_m_geo_i)
    colormap Copper

figure (3)
    plot
    (store_U_inf,graph_CL_result,store_U_inf,graph_L_D,store_U_inf,graph_CL
_result2)
    axis([store_U_inf(1) store_U_inf(50) 0 Best_L_D_cruise_product+5 ])
    xlabel({'V (ft/s) '},'FontSize',9,'color','r')
    ylabel({'CL/Cd CL^(1/2)/Cd CL^(3/2)/Cd},'FontSize',9,'color',
'r')
    title('Aerodynamic Ratios vs Airspeed Optimal Wing','color',
'b','FontSize',10 )
    grid on

figure (4)
    plot
    (store_U_inf,graph_CL_result_i,store_U_inf,graph_L_D_i,store_U_inf,grap
h_CL_result2_i)
    axis([store_U_inf(1) store_U_inf(50) 0 Best_L_D_cruise_product+5 ])
    xlabel({'V (ft/s) '},'FontSize',9,'color','r')
    ylabel({'CL/Cd CL^(1/2)/Cd CL^(3/2)/Cd},'FontSize',9,'color',
'r')
    title('Aerodynamic Ratios vs Airspeed Initial Wing','color',
'b','FontSize',10 )
    grid on

```

```

figure (5)
    plot (graph_AOA_R,graph_L_D)
    xlabel({'AOA (deg) '},'FontSize',9,'color', 'r')
    ylabel({'L/D'},'FontSize',9,'color', 'r')
    title('Lift to Drag Variation with Angle of Attack Optimal
Wing','color', 'b','FontSize',10 )
    grid on

figure (6)
    plot (graph_AOA_R_i,graph_L_D_i)
    xlabel({'AOA (deg) '},'FontSize',9,'color', 'r')
    ylabel({'L/D'},'FontSize',9,'color', 'r')
    title('Lift to Drag Variation with Angle of Attack Initial
Wing','color', 'b','FontSize',10 )
    grid on

figure (7)
    plot (store_angle_attack_opt,Cl_result_optimal,'b')
    axis([store_angle_attack_opt(1) store_angle_attack_opt(12)
Cl_result_optimal(1) Cl_result_optimal(12)])
    xlabel({'Alpha (deg) '},'FontSize',9,'color', 'r')
    ylabel({'CL'},'FontSize',9,'color', 'r')
    title('CL vs Alpha Curve','color', 'b','FontSize',10 )
    grid on
    hold on
    plot (store_angle_attack_initial,Cl_result_initial,'r')
    grid on
    legend('Optimal Wing','Initial Wing')

figure (8)
    plot (store_U_inf,graph_TR_result)
    xlabel({'V (ft/s) '},'FontSize',9,'color', 'r')
    ylabel({'Drag (lbs)'},'FontSize',9,'color', 'r')
    title('Drag Variation with Airspeed Optimal Wing','color',
'b','FontSize',10 )
    grid on

figure (9)
    plot (store_U_inf,graph_TR_result_i)
    xlabel({'V (ft/s) '},'FontSize',9,'color', 'r')
    ylabel({'Drag (lbs)'},'FontSize',9,'color', 'r')
    title('Drag Variation with Airspeed Initial Wing','color',
'b','FontSize',10 )
    grid on

```

Appendix A.2: Drag polar diagram of the NACA 65-210 airfoil. Taken from Theory of Wing Sections: Abbott and Doenhoff. (pg 613). Matlab uses this diagram for the calculation of the profile drag coefficient.

