6-2005

Drag Estimates for the Joined-Wing Sensor Craft

Ryan L. Craft

DRAG ESTIMATES FOR THE JOINED-WING SENSOR CRAFT

THESIS

Ryan L. Craft, Ensign, USN

AFIT/GAE/ENY/05-J02

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GAE/ENY/05-J02

DRAG ESTIMATES FOR THE JOINED-WING SENSOR CRAFT

THESIS

Presented to the Faculty

Department of Aeronautics and Astronautics

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Aeronautical Engineering

Ryan L. Craft, BS

Ensign, USN

June 2005

AFIT/GAE/ENY/05-J02

DRAG ESTIMATES FOR THE JOINED-WING SENSOR CRAFT

Ryan L. Craft, BS
Ensign, USN

Approved:

_____/signed/ _____ _____
Dr. Robert Canfield (Chairman) date

_____/signed/_____ _____
Lt Col Eric Stephen (Member) date

_____/signed/_____ _____
Dr. Ralph Anthenien (Member) date

Acknowledgements

I would like to express my sincere appreciation to my faculty advisory, Dr. Robert Canfield, for his guidance and support throughout the course of this effort. I would, also, like to thank Dr. Maxwell Blair, from the Air Force Research Laboratory, for the software support and perspective provided to me in this research.

Special thanks go to the many great friends that surrounded me this year, new and old, both civilian and military, who always kept me motivated throughout the course of this study. And of course, my most sincere appreciation goes to my family. To a mother and father who have invested their time and energy in raising a family I am very proud to be a part of, offering continuous emotional support and love.

Ryan L. Craft

Table of Contents

Page

Page

Page

List of Figures

Figure Page

Figure Page

List of Tables

Table Page

List of Symbols

Symbol	Definition
α, AOA	Freestream Angle-of-Attack
α_i	Induced Angle-of-Attack
α_L	Local Angle-of-Attack
δ	Aft Wing Root Twist Angle
η_p	Propeller Efficiency
Λ_{ib}	Inboard Wing Sweep
Λ_{ob}	Outboard Wing Sweep
λ	Taper Ratio
τ	Span Efficiency Scaling Factor
τ_{AR}	Taper Ratio Efficiency Scaling Factor
A/C	Aircraft
AR	Aspect Ratio
AW	Aft Wing
C	Specific Fuel Consumption
C_D	Drag Coefficient
C_d	Two Dimensional Drag Coefficient
C_{Do}	Zero Lift Drag Coefficient
C_{Di}	Induced Drag Coefficient
C_{DL}	Local Drag Coefficient Oriented With Local Velocity Vector
C_{Dp}	Parasite Drag Coefficient
C_{Dtotal}	Total Drag (Parasite and Induced)

C_F …….. Turbulent Mean Skin-Friction Coefficient

C_L ……. Lift Coefficient

C_l …… Two Dimensional Lift Coefficient

C_{LL} …… Local Lift Coefficient Oriented With Local Velocity Vector

C_M …… Moment Coefficient

c_m …… Mid-Chord

c_{ra} ……. Aft Root Chord

c_{rf} …….. Fore Root Chord

c_t ……... Tip Chord

d_f ……. Fuselage Diameter

D ……Drag

D_L …….. Local Drag Oriented With Local Velocity Vector

D_∞ …….. Component of Drag Oriented With Respect to Freestream Velocity Vector

e_{span} …….. Span Efficiency Factor

e_{oswald} …… Oswald Efficiency Factor

f ……. Equivalent Parasite Area

FIW ……Forward Inside Wing

FOW ……Forward Outside Wing

fuse ……. Fuselage

i ……. Mission Leg Segment Identifier

k …….. Drag Due To Lift Correction Factor

L …….. Lift

L’ ……..Airfoil Thickness Location Parameter

l_f ……..Fuselage Length

L/D ……. Lift-to-Drag Ratio

L_L ……. Local Lift Oriented With Local Velocity Vector

L_∞ ……Component of Lift Oriented With Respect to Freestream Velocity Vector

M …….. Mach Number

m …….. Mass

R ……..…… Range

r ……..Oswald Efficiency Correction Factor

Re, R_N …….. Reynolds Number

R_{LS} …….. Lifting Surface Correction Factor

R_{WF} …… Wing-Fuselage Interference Factor

S …….. Wing Planform Area

S_{ib} …… Inboard Span

S_{ob} ……. Outboard Span

S_{wet} …… Wetted Planform Area

t/c …… Thickness-to-Chord Ratio

V ……. Velocity

V_L ……. Local Velocity Vector

V_∞ ……. Velocity Relative To Freestream

W …….. Weight

w ……. Downwash

x …….. X-Coordinate Frame of Airfoil

X_{ac} ……. Location of Aerodynamic Center In X-Coordinate Frame

X_{cg} …….. Location of Center of Gravity In X-Coordinate Frame

x_{fa} ……. Fore-Aft X-Offset

z …….. Z-Coordinate Frame of Airfoil

z_{fa} ……. Fore-Aft Z-Offset

Abstract

This research studied the drag effects of the joined-wing sensor craft technology demonstrator being developed at the Air Force Research Laboratory. Although many performance parameters have been studied and evaluated for this vehicle, to date no detailed drag estimates have been conducted for the AFRL configuration. Previous performance parameters of the aircraft have been estimated based solely on a constant lift-to-drag ratio assumption. Using the Air Vehicles Technology Integration Environment created by Dr. Maxwell Blair, and supplemented by MATLAB code, this study explored three different drag prediction methods to determine accurate estimates of both parasite and induced drag. The Roskam/AVTIE Pan Air method was determined as the ideal approach to estimate drag by measuring parasite drag effects using XFOIL, a respected environment within the aviation industry to accurately predict all viscous drag effects, and determined induced drag from Pan Air, a creditable software package based on inviscid flowfield solutions about three dimensional objects. This method will be incorporated into a single design environment, in conjunction with AVTIE, in order to estimate drag and aid future AFRL joined-wing design studies incorporating wing twist, aeroelastic effects, and other geometric changes to the baseline configuration.

DRAG ESTIMATES FOR THE JOINED-WING SENSOR CRAFT

I. Introduction

1.1 Overview

The combat zone of 20 years ago differs drastically with that of today's due to the technology of unmanned aerial vehicles (UAVs) for use as primarily surveillance platforms. UAVs have proved to be especially effective in intelligence, surveillance, and reconnaissance (ISR) missions which demand continuous high altitude coverage over a span of 24 hours or more. Most famous of these aircraft are the RQ-4A Global Hawk and the RQ-1 Predator. However, these aircraft are only capable of surveying targets within plain view from the sky above. Enemies are realizing that hiding equipment under tents and treetop canopies prevents detection from the current threat of surveillance UAVs.

In order to adapt to the ever changing combat zone, the United States Air Force is investigating a new type of ISR mission. The United States is in need of a high altitude, long endurance, UAV with full 360-degree field of view coverage capable to detect equipment under foliage. Foliage penetration demands an aircraft with large sensors and antennas able to produce signals with long wavelengths. Current configurations such as the Global Hawk are not suitable for providing full 360° continuous coverage, nor foliage penetration. Another possible configuration is that of a flying wing with sensors and antennas integrated into the highly swept wings. From this possible configuration spawned the concept of the joined-wing sensor craft (Figure 1, Figure 2, and Figure 3).

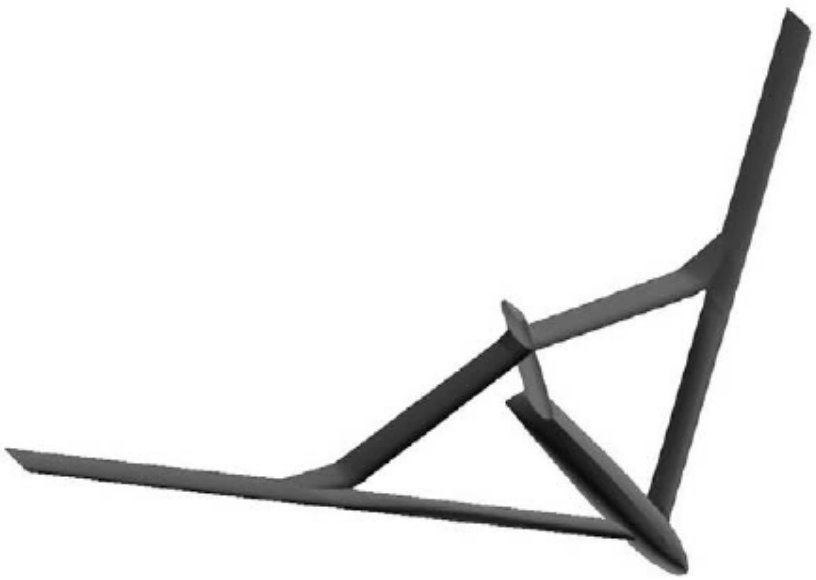

Figure 1. Typical Joined-Wing Concept Geometry

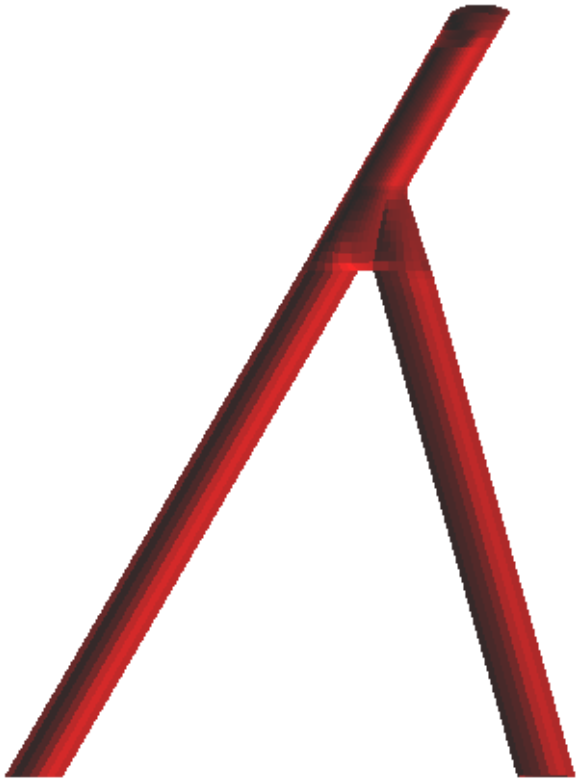

Figure 2. Top View of Proposed Right-Half Joined-Wing Geometry

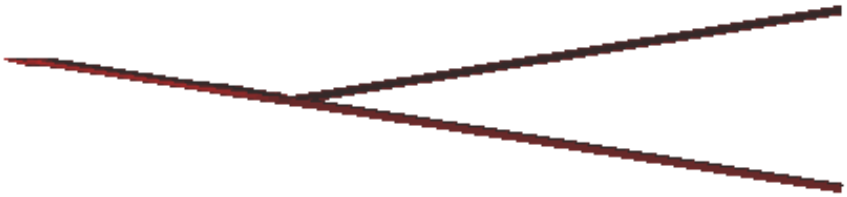

Figure 3. Front View of Proposed Right-Half Joined-Wing Geometry

The joined-wing concept is a revolutionary digression from the current world inventory of aircraft. Potential gains from such a design could lead to improved radar signature, enhanced aerodynamic performance, and a decrease in structural weight. The joined-wing aircraft typically consists of a large lifting surface, the aft wing, with forward sweep and negative dihedral, connecting the top vertical tail with the main, or fore, wing. This aft wing serves as a support strut for the cantilevered main wing and alleviates bending moments. In flight, the main wing will tend to flex up due to the production of lift and the aft wing will be subjected to axial compression throughout most of the flight profile.

The proposed joined-wing sensor craft design features an embedded radar antenna in the forward and aft wings providing a large aperture, enabling ultra high frequency (UHF) surveillance with a 360-degree field of view of a target area. UHF is a required radar frequency for foliage penetration (FOPEN) [1].

In order to decrease weight, the antenna elements are built into the composite wing structure. This Conformal Load-bearing Antenna Structure (CLAS) is a composite sandwich of graphite-epoxy, honeycomb carbon foam core, and an astroquartz skin

covering (Figure 4). Antenna elements are attached to the upper graphite-epoxy layer, while the electro-magnetically clear astroquartz layer provides environmental protection for the radar to transmit through.

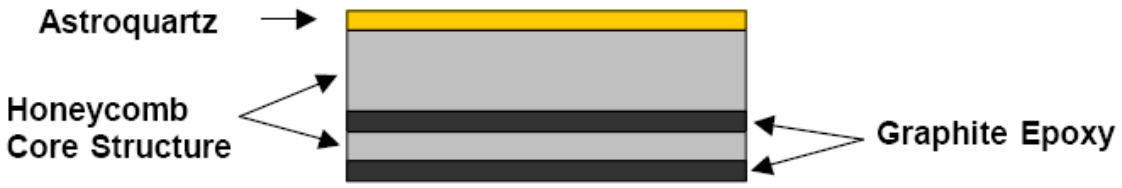

Figure 4. Conformal Load-Bearing Antenna Structure Cross Section

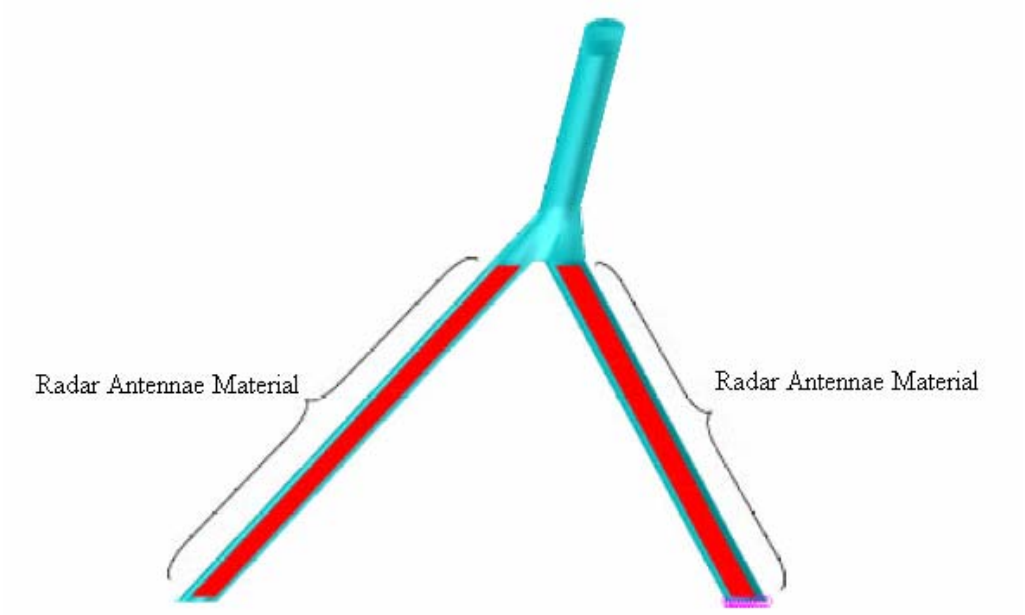

Figure 5. Radar Antennae Location

The front and aft wing sweep angles are constrained by the system's radar coverage requirements. The radar contained within the wings, shown in Figure 5, must provide 360-degrees of coverage around the vehicle.

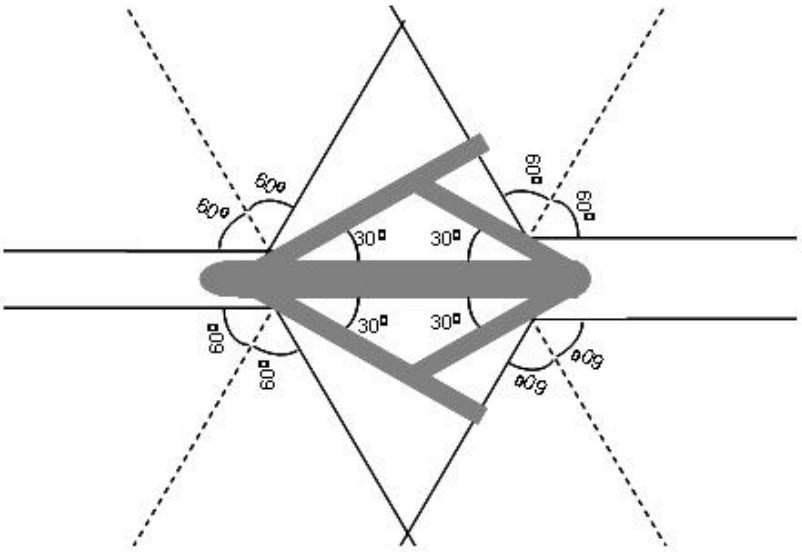

Figure 6. Maximum Wing Sweep Constraint

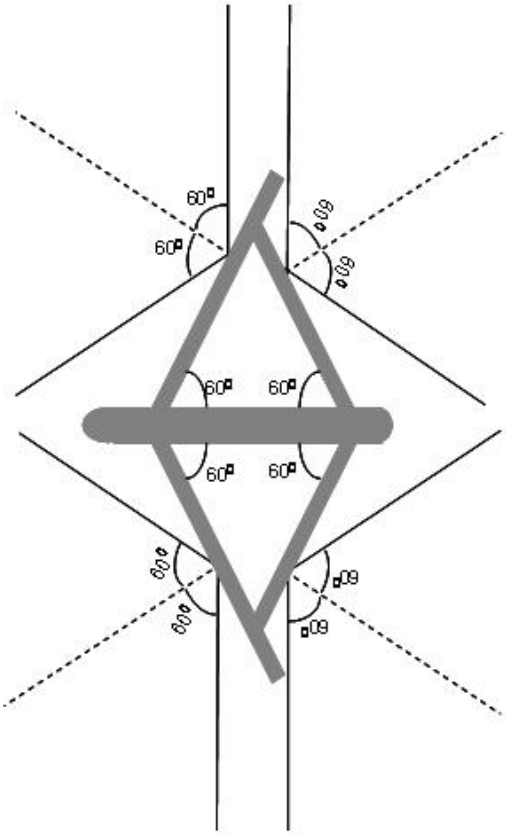

Figure 7. Minimum Wing Sweep Constraint

The maximum change in electromagnetic beam steering angle from the normal direction of the wing at which end-fire radar can properly receive/transmit is approximately 60 degrees, also known as the grazing angle. In order to prevent blind spots, possible wing sweeps range from 30 to 60 degrees (Figure 6, and Figure 7). High wing sweep allows better high speed performance; however, these high sweep angles force the weakest portion of radar coverage to lie at the aircraft's 12 o'clock position, the most probable location for targets. Less wing sweep results in better radar coverage and improved fuel consumption by increasing loitering performance, a crucial design parameter for an aircraft of this type.

1.2 Research Objectives

Prior analysis of the aerodynamic performance of the joined-wing sensor craft assumed a constant lift-to-drag (L/D) ratio of 24 throughout its flight profile. This research begins to examine the drag forces by estimating parasite and induced drag the aircraft would experience in flight. Several methods were utilized in order to accurately model both the parasite and induced drag forces on the aircraft. In addition, several models were analyzed, one base model without any wing twist from which multiple models were created utilizing wing twist in order to minimize induced drag in an effort to maximize L/D, improving fuel consumption. The ultimate objective is to develop a method to accurately evaluate drag characteristics for any joined-wing geometry. This process will be implemented into a single design environment used to integrate structural optimization with aerodynamic optimization to achieve overall vehicle system optimization. A single integrating design environment to optimize weight and drag

characteristics and analyze structural performance will aid future joined-wing aerodynamic optimization studies.

1.3 Research Focus

This research focused on aerodynamic properties of the rigid joined-wing sensor craft. Since estimating drag is difficult, multiple drag buildup methods were utilized in order to converge on an accurate drag assessment. Throughout the flight profile, the aircraft was aerodynamically trimmed using the aft wing as a pitch control surface. At each trimmed point of the flight profile, drag forces were determined. This research recognizes that all approaches to drag buildups are estimates, but the mutually consistent use of several methods will ensure more accurate results than the previous constant L/D assumption. Wing twist was applied to the baseline configuration in an effort to optimize the wing design, based on an elliptic lift distribution and decreased induced drag effects.

1.4 Methodology Overview

Multiple methods for drag estimation were utilized in order to allow comparison and convergence on the aircraft's actual L/D ratio. Roskam [2] provides very detailed pressure drag estimation in his aircraft design series that includes all drag forces, except for induced drag, at both subsonic and supersonic flight regimes. He presents several crucial characteristic trend lines that govern the drag forces that act on an aircraft. Roskam's drag buildup method was incorporated into MATLAB [3] code that interpolated between various characteristic lines in order to generate results. This method

depends only on the physical dimensions of the aircraft and compares it to actual experimental data determined from previous similar configurations in order to produce an estimate. However, the joined-wing concept is considered a radical design to the aviation industry, and generating preliminary aerodynamic conclusions based exclusively on the Roskam method will not be accepted as a genuine drag estimate.

Adaptive Modeling Language (AML) [4] was also used to supplement the drag estimates from Roskam. AML is an object oriented prototyping environment and is used here to develop a geometric model that contains all required information needed to calculate drag forces about the joined-wing aircraft. AML is characterized as a LISP-like scripted language which directs compiled object code [5]. AML user objects vary from conventional object-code (e.g. C++) in that any object component or process is automatically available from within any other object of the code. The base AML class manages automated dependency tracking on every member property (member variable) through object inheritance [5]. Dependency tracking provides a model that is always current with respect to any modifications. This attribute allows one to invoke many changes before forcing preferred consequences. For example, the mission profile, the wing span, the airfoil section and so on can be altered, thereby forcing a subsequent calculation of dependent responses.

Dr. Maxwell Blair [5] employed AML to create the Air Vehicles Technology Integration Environment (AVTIE). It enables designers to develop aerodynamic loads and perform aircraft trim calculations. AVTIE drives aerodynamic results and accounts for both parasite and induced drag effects. Although this software is fully capable of evaluating the aerodynamic characteristics of the entire vehicle, it is applied to the wing

structure only, neglecting the fuselage and vertical tail. AVTIE is the central source of wing drag estimates and relies on two other programs, XFOIL [6] and PanAir [7].

Pan Air is a program that calculates flowfield properties about arbitrary three-dimensional configurations. The program uses a higher-order panel method to solve the linearized potential flow boundary-value problem at subsonic and supersonic Mach numbers. The aerodynamic solution provides surface flow properties (flow directions, pressures, Mach number, etc.), configuration forces and moments, sectional forces and moments, and pressures. In addition, Pan Air calculates flow properties in the flow-field and flow-field streamlines and results are limited to inviscid subsonic and supersonic cases (transonic cases excluded) with attached flow.

XFOIL is a program for the design and analysis of subsonic two dimensional airfoils. It consists of a collection of menu-driven routines which perform various useful viscous functions such as boundary layer effects and transition, lift and drag predictions, drag polar calculations with fixed or varying Reynolds and/or Mach numbers, etc. The two dimensional drag data generated by XFOIL was assumed applicable up to 30 degrees of wing sweep. XFOIL provides AVTIE parasite drag values for the wing only, based on drag polar estimations. XFOIL viscous data is also used to supplement Pan Air inviscid data.

1.5 Assumptions and Limitations

The joined-wing sensor craft concept is being studied by a number of aircraft design companies. This study is based on the Air Force Research Laboratory (AFRL) baseline model. The most critical assumption applied to this research implied a rigid

model without any flexible wing deformations, an unrealistic assumption for this type of high aspect ratio wing aircraft. However, the procedures developed herein remain valid when aeroelastic effects are incorporated. Also, all induced drag was assumed to act on the wing structure alone and neglected the fuselage and vertical tail. Skin friction estimates are determined from the AFRL baseline model that incorporates aluminum materials, although most likely any joined-wing production aircraft would be constructed of composite type materials. Throughout each drag buildup method presented later in this study, further assumptions and limitations will be discussed with possible side effects and sources of error.

1.6 Implications

This multi-objective approach to aircraft design requires techniques that encompass all aspects of the conceptual design process. This allows the aircraft designers to observe and incorporate the interactions of aerodynamic effects. AVTIE also allows the researcher to study the effects of wing twist and its magnitude of improvement on aerodynamic performance. This research demonstrated the ability to incorporate many drag estimation methods in order to converge on more accurate L/D calculations. Another important result was an optimized wing twist distribution for the baseline rigid configuration. Potentially, AVTIE is capable of developing an optimized conceptual design for any aircraft configuration.

II. Literature Review

2.1 Introduction

This chapter summarizes the relevant joined-wing aerodynamic research already accomplished in past studies. First, it reviews characteristics that are required for such an aircraft to perform an essential mission desired by the United States Air Force. Next, it reviews the advantages obtained with this new concept and highlights some of the possible problems the design will encounter.

This chapter also discusses past research in the areas of aerodynamic analysis and structural optimization, which ultimately drives physical characteristics of the aircraft. It also makes note of differences between past research and the research presented here. In addition, this chapter reviews a proposed method of aerodynamic optimization. This chapter concludes by describing the AFRL joined-wing sensor craft configuration that is utilized in this research and its mission profile.

2.2 Requirements

The High-Altitude Long-Endurance (HALE) mission demands a large wingspan with high aspect ratio. Sustaining dynamic pressure at greater altitude within HALE missions requires increased speed, ultimately leading to transonic effects during cruise and loiter. The long slender wing design results in increased flexibility over conventional aircraft wings. This fact alone invites interest in the joined-wing concept with the aft wing serving as a support strut of the main wing.

Past research has compared the joined-wing concept with the strut-braced wing (SBW) designs. Surely, one could undergo a design investigation with a continuous spectrum of shapes ranging from an aft wing airfoil section to a SBW. In all cases, the main wing is reinforced with a second structure, which is mostly dominated by compressive loads due to upward main wing flexure. Contemporary studies [8] suggest the SBW may be a superior design over the joined-wing concept for commercial operation due to transonic effects. However, it is the airborne sensor mission that drives the study of the joined-wing vehicle, one capable of 360-degree surveillance.

2.3 Past Joined-Wing Design Work

In 1974, Miranda [9] proposed a "boxplane wing" design with claims such as improved controllability and maneuverability, low induced drag, and structural integrity. This boxwing configuration comprises the swept back fore wings, the forward swept aft wings and the interconnection of the tips of these wings by swept vertical fins for lateral stability (Figure 8).

The first concept of a joined-wing design was patented by Julian Wolkovich [10] in 1976 (Figure 9, and Figure 10). In later published studies, Wolkovich claimed the general concept of the joined-wing design provided potential weight savings and aerodynamic benefits [11]. In addition to a lighter aircraft, Wolkovich claimed a strategically designed joined-wing aircraft would exhibit several advantages over conventional aircraft, including a reduction in induced drag, higher maximum lift coefficients (C_{Lmax}), improved stability and control characteristics, and reduced parasitic drag, among other advantages [11].

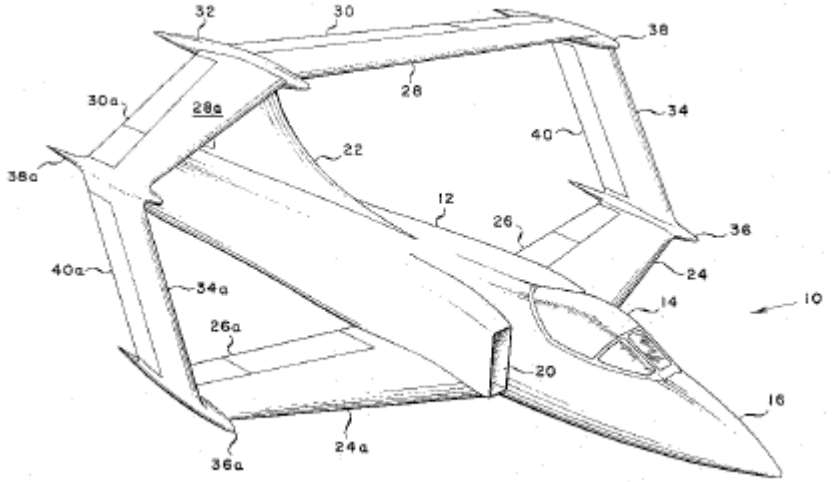

Figure 8. Boxwing Concept Airplane

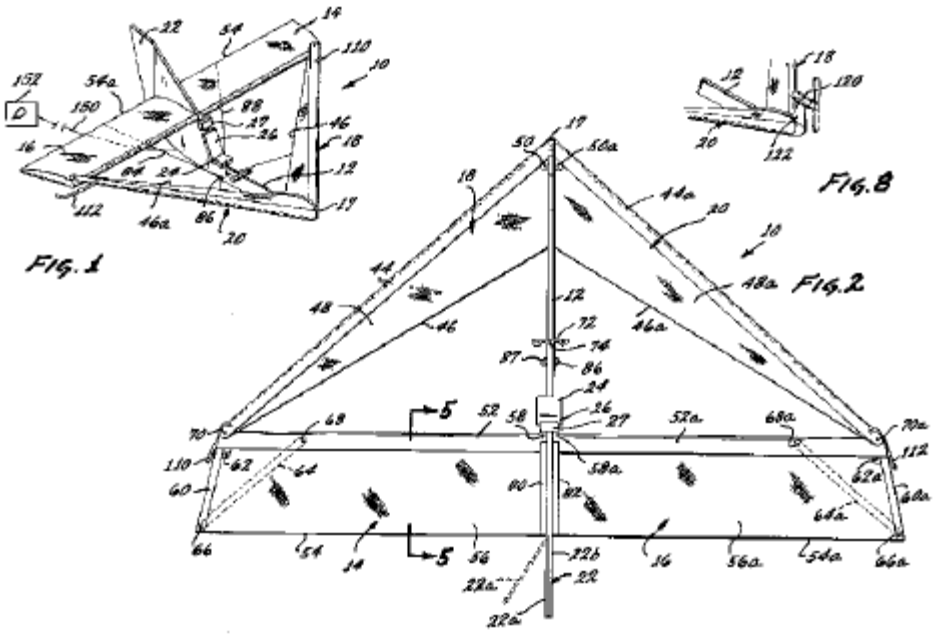

Figure 9. Wolkovich's First Joined-Wing Concept

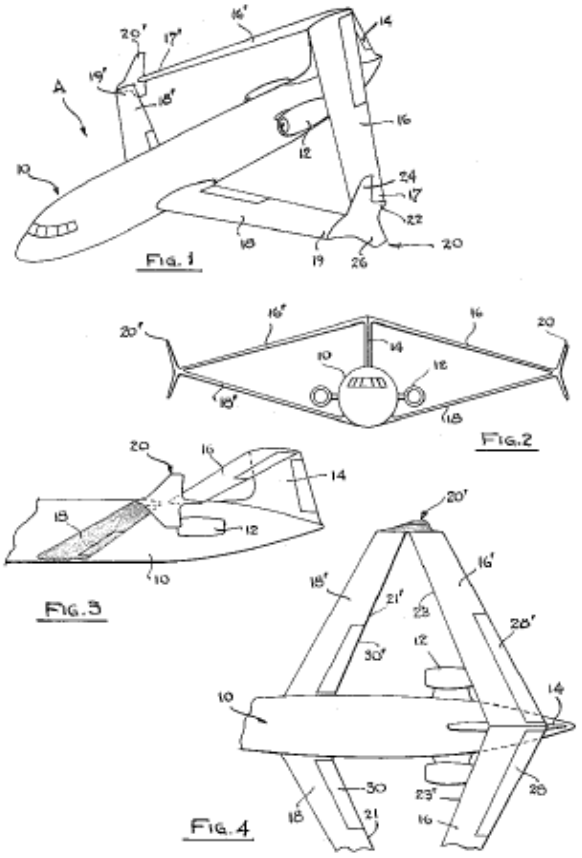

Figure 10. Wolkovich's Second Joined-Wing Concept

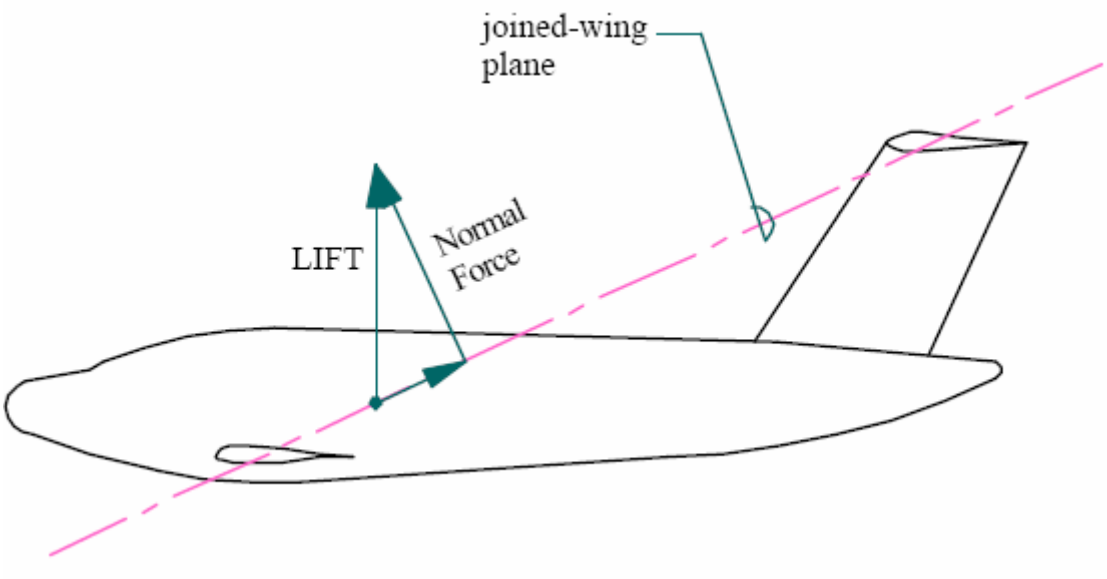

Figure 11. Lift Force Components in the Joined-Wing Plane

Wolkovich also observed that the total vertical lifting force from the forward and aft wings can be resolved into a force acting normal to and parallel to the structure of joined wing (Figure 11). The force normal to the joined wing plane creates a bending moment about the z-axis. This normal force is also a component of the drag of the aircraft, and will be discussed in detail.

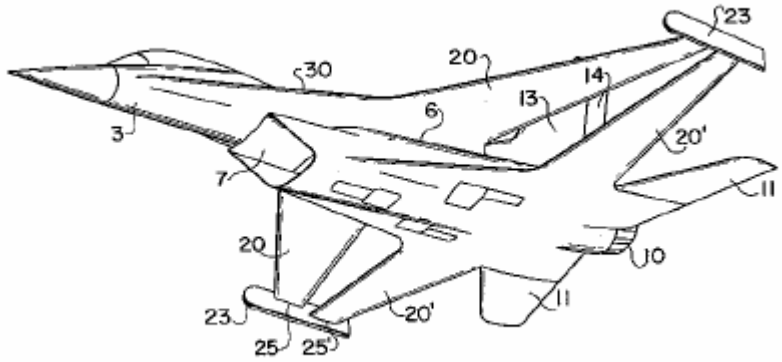

Figure 12. Superposed Wing Concept by Zimmer

An "airplane with two superposed wings" was first researched by Zimmer [12] in 1978. The characteristics of this configuration are two superposed sweptback wings, which together constitute a closed frame in a front view (Figure 12). Such wing configurations are based on the fact that induced drag is proportional to the square of the lift and inversely proportional to the geometric extension of the wing in the direction of its span and height, and can be decreased with such a design. These interrelations were first theoretically researched by Ludwig Prandtl and Max Munk.

In 1982, Samuels [13] compared the structural weight of a joined-wing with that of a Boeing 727 wing. He found that the joined-wing structure was 12 – 22% lighter than

that of a conventional configuration. Hajela and Chen [14] and Hajela [15] related the significant weight savings with an increase in the dihedral angle of a joined-wing configuration. Hajela used a fully stressed design procedure and an equivalent beam model. Miura et al. [16] states that structural weight traits of a joined-wing depend strongly on the structural arrangement and wing geometry. This study displayed that a joined-wing configuration had promising opportunities for decreasing structural weight. Wolkovich [11] claimed both structural and aerodynamic advantages including structural weight reduction, low decreased induced drag, improved transonic area distribution, high trimmed maximum lift coefficient, and reduced wetted area and parasite drag.

Frediani [17] applied the studies of the boxwing design to larger transport aircraft (Figure 13). The proposed advantages were similar to those of the joined-wing concept with reduced induced drag and structural weight savings. He also found an increase in the aircraft's damage tolerance and better characteristics of weight efficiency and fatigue life. He also addressed the issues of static aeroelastic problems such as control reversal and aerodynamic and structural load redistributions.

Early in the research of the joined-wing concept, Fairchild [18] completed a structural weight comparison between a conventional wing and the joined-wing design. Utilizing the same NACA 23012 airfoil section for both models, throughout the study he held the structural box size and thickness ratio constant. His conclusions show the joined-wing concept displayed a 50% reduction in vertical wing deflection over the conventional non-reinforced wing. Also, the study found that for aerodynamically similar configurations, the joined-wing design was approximately 12% lighter than conventional configurations.

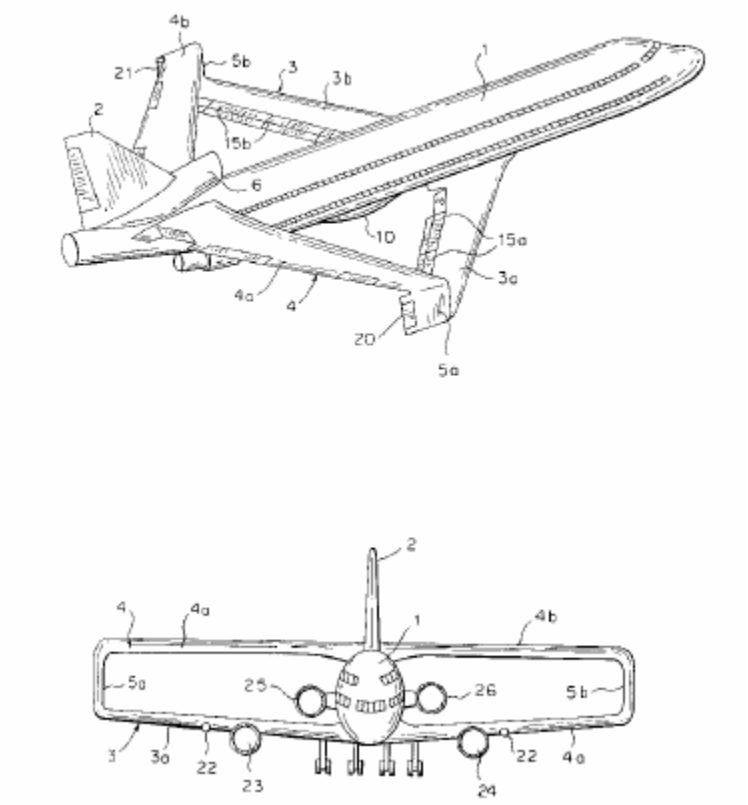

Figure 13. Frediani Box Wing Concept for Large Transport Aircraft

NASA Ames Research Center instigated studies into the possibility of developing a full scale joined-wing aircraft [19]. The proposed aircraft was to be manned, forcing many goals of the project towards good handling qualities. Smith et al. concluded the joined-wing concept decreases bending moments within the forward wing and determined a span efficiency factor greater than 1.0 [19]. The span efficiency factor is defined as the ratio of the induced drag created by an elliptical lift distribution to the actual induced drag distribution. The results of a span efficiency factor greater than one validates the previous claim of reduced drag from conventional configurations [11].

NASA Ames researchers found that even with elaborate aerodynamic design optimization, the one-sixth scale wind tunnel model exhibited instabilities near stall

angles-of-attack (AOA) in both the longitudinal and lateral frames. These unfavorable stall characteristics were improved on the wind tunnel model by installing vortilons, but a full scale demonstrator was never built.

However, Lin, Jhou, and Stearman continued the research from the NASA Ames research program, using the same wind tunnel model as the basis of their studies [20]. From this base model, the researchers studied different joint configurations attempting to optimize the union between the forward and aft wings. In total, eight different configurations were studied using Finite Element Modeling (FEM) analysis and experimental data generated in the wind tunnel. Their conclusions confirm that the best joint designs are a rigid joint, or a pinned joint with the z-axis free to rotate [20]. This supplemented studies performed by Gallman et al. [21] who concluded that a joint location at 70% of the forward wing semispan would provide a 11% reduction in drag over a conventional aircraft of similar physical dimensions.

Kroo et al. [22] used several design variables in order to develop a method to optimize a joined-wing configuration with regards to aerodynamics and structural performance. Their method utilized a vortex lattice aerodynamic code to trim the aircraft in order to achieve a minimum drag attitude. In all configurations studied, the aft wing produced a negative lift load required to trim the aircraft. Many conventional aircraft of today also require a negative lift contribution from the horizontal stabilizers in order to remain in trimmed flight. However, due to the joined-wing's unusually large horizontal control surface (the entire aft wing), the effects of producing a negative lift contribution by twisting this surface greatly increases the pareasite drag and nullified the expected reduction in induced drag.

Complementing the work presented here is the work of Lee and Weisshaar [23]. These authors provided significant insight into the important role of flutter in regards to joined-wing aircraft designs. Their models included structural optimization of laminated composite material with linear static aeroelastic and flutter constraints.

The studies of Gallman and Kroo also suggested that the potential of aft wing buckling negated possible weight savings due to structural hardening of the supportive wing. Also varying the location of the forward and aft wing joint, the authors concluded a large reduction in weight could be achieved with a wing joint located at 70% of the forward wing span [22], verifying the works of Gallman [21]. Motivated by the works of Kroo and Gallman the AFRL joined-wing concept uses a rigid joint at 70% semispan.

2.4 Recent Joined-Wing Research

Recent research on the joined-wing concept has been primarily devoted to the integration of structural and aerodynamic design. Many physical characteristics of the joined-wing design are direct results of aeroelastic effects, and the aircrafts ability to endure the aerodynamic loads it will encounter throughout flight. Livne [24] analyzed previous joined-wing research in order to provide a course for future studies. Using non-linear multi-disciplinary approaches, he explains the general joined-wing configuration creates complex interactions between structural and aerodynamic loads.

Blair and Canfield [25] continued work for the joined-wing concept with AFRL. They proposed an integrated design method for joined-wing configurations. In their studies, they chose to model a joined-wing configuration specifically for a sensor craft mission. An area of great importance to the authors was the aft wing and its

susceptibility to buckling. Realizing the aft wing will be under compression for long periods of time, they decided not to install a separate moving control surface for pitch control. Instead, in order to control longitudinal trim, they decided to twist the entire aft wing. This had the added benefit removing control surfaces from the vicinity of embedded UHF antenna. Similar to previous studies, Blair and Canfield also used a rigid wing joint for the model.

The concept started the simulated mission with an initial estimate of fuel required based on the Breguet range equation and a constant lift-to-drag ratio. The Breguet formula is given below in its normal form, where R_i is the range for the i^{th} mission segment, V is velocity, C is specific fuel consumption, L/D is the lift-to-drag ratio, and m is the mass.

$$R_i = \left(\frac{V}{C}\right)\left(\frac{L}{D}\right)\ln\left(\frac{m_{i-1}}{m_i}\right) \tag{1}$$

Blair and Canfield advised other researchers that large aft wing twist inputs created high angles-of-attack conditions, producing excessive drag and should be avoided. They also validated the works of Kroo [22] in that negative lifting force on the aft wing greatly increased drag on the aircraft.

2.5 Previous Research On The AFRL Joined-Wing Configuration

Based on prior studies by Blair and Canfield [25], research has continued on the baseline AFRL joined-wing model at the Air Force Institute of Technology (AFIT), in

conjunction with AFRL. Recently, master's students at AFIT have thoroughly studied certain design parameters and constraints of the AFRL model.

Roberts [26] analyzed aeroelastic effects and potential aft wing buckling due to aerodynamic loads. His studies demonstrate that the proposed AFRL sensor-craft is a highly coupled, multi-disciplinary design. Both linear, and non-linear, analysis of aerodynamic wing deflection resulted in a buckling safe design for all maneuver loads the model would endure throughout the flight profile.

Smallwood [27] investigated the effects of wing deflections on the conformal, load-bearing antenna arrays embedded within the wing structure. This was a multi-disciplinary effort that touched on the aerodynamic, structural, and electromagnetic design considerations that stem from this unique type of sensor integration. His studies concluded that wing deflections due to typical aerodynamic loads produce significant disturbances to the radiation pattern of conformal antenna when end-fire phasing is applied, and corrective action will be required with beam steering in order to maintain 360 degree sensor coverage.

Rasmussen [28] optimized the joined-wing configuration geometry based on aerodynamic and structural performance. Analysis was completed utilizing structural optimization, aerodynamic analyses, and response surface methodology. In total, 74 joined-wing configurations spawned from the AFRL baseline configuration and were optimized with respect to weight. Each optimized structure was determined through a change of skin, spar, and rib thickness in the wing box by determining trimmed maneuver and gust conditions for critical flight mission points. Each configuration varied one of

six key geometric variables. These included front wing sweep, aft wing sweep, outboard wing sweep, joint location, vertical offset, and thickness to chord ratio.

Sitz [29] performed an aeroelastic analysis of the joined-wing sensor craft. The analysis was completed using an aluminum structural model that was splined to an aerodynamic panel model. The force and pressure distributions were examined for the aft wing, forward inside wing, joint, and tip sections. Her studies concluded both distributions provide expected elliptical results, with the exception of the forward inside wing. This section appeared to be affected by interference from the wing joint. She also analyzed the use of control surfaces for purposes of pitch, roll, yaw, and trimming the aircraft. Results validated those calculated in previous studies.

2.6 Basis For Current Research

This research will continue the work of Blair and Canfield [24] and Sitz [29] with the AFRL joined-wing sensor craft model. Although these authors have thoroughly studied many performance parameters of the model, to date no detailed drag studies have been conducted on the AFRL design. All performance calculations in the AVTIE code of [24] have been based solely on a constant lift-to-drag ratio assumption. Using the AVTIE interface (Appendix D), working in conjunction with AML, XFOIL, and Pan Air, a detailed drag assessment was conducted for the joined-wing craft. The AVTIE program was utilized to determine the drag contributed by the wing alone. The wing will be responsible for the majority of the drag of the entire aircraft configuration. Fuselage and vertical tail drag were estimated in this research by the Roskam drag buildup method and added to the results from AVTIE to assess drag experienced by the whole aircraft

configuration. Lastly, wing twist was employed on the model in order to reduce induced drag and to satisfy an elliptical lift distribution, optimizing the aircraft's wing planform and improving its cruise and loiter lift-to-drag ratio.

2.7 The AFRL Joined-Wing Model

Table 1 displays the weight breakdown for the aircraft. Initial fuel estimates were derived from Equation (1) assuming a constant L/D of 24. Payload includes mission essential items such as surveillance equipment and possibly weapons.

Table 1. AFRL Joined-Wing Weight Breakdown

Component	Mass (kg)
Payload	3,550
Engine	1,760
Fuel	24,674
Wing Structure	6,780
Fuselage Structure	2,170
Tail Structure	100
Total Assumed	39,034

Figure 14 displays general joined-wing nomenclature and Table 2 shows the corresponding physical properties of the AFRL model. The propulsion system has a strong influence on the resulting vehicle design. Many propulsion systems are still candidates for the joined-wing concept; however, a turboprop in a pusher (aft) position was selected for this study.

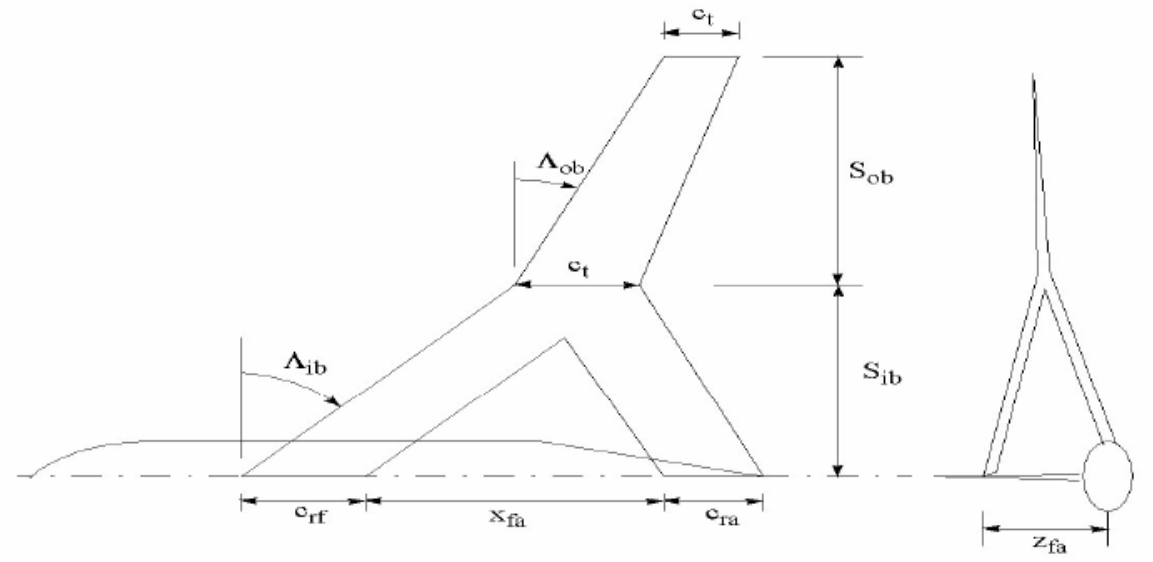

Figure 14. AFRL Joined-Wing Nomenclature

Table 2. AFRL Joined-Wing Configuration Parameters

Parameter	Symbol	SI	USCS
Inboard Span	S_{ib}	26.00 m	85.30 ft
Outboard Span	S_{ob}	8.00 m	26.25 ft
Fore Root Chord	c_{rf}	2.50 m	8.20 ft
Aft Root Chord	c_{ra}	2.50 m	8.20 ft
Mid-Chord	c_m	2.50 m	8.20 ft
Tip Chord	c_t	2.50 m	8.20 ft
Fore-Aft X-Offset	x_{fa}	19.50 m	62.34 ft
Fore-Aft Z-Offset	z_{fa}	7.00 m	22.97 ft
Inboard Wing Sweep	Λ_{ib}	30 deg	30 deg
Outboard Wing Sweep	Λ_{ob}	30 deg	30 deg
Airfoil		LRN-1015	LRN-1015
Calculated Wing Planform Area	S	143.50 m^2	1544.62 ft^2
Calculated Wing Volume		71.70 m^3	2532.06 ft^3

2.8 The LRN-1015 Airfoil

The current baseline AFRL model utilizes the LRN-1015 airfoil section throughout its wingspan, except within the joint section. This airfoil section provides exceptional aerodynamic characteristics for HALE mission oriented aircraft. The geometrical shape of the LRN-1015 airfoil is shown in Figure 15, and its XFOIL generated drag polar is shown in Figure 16.

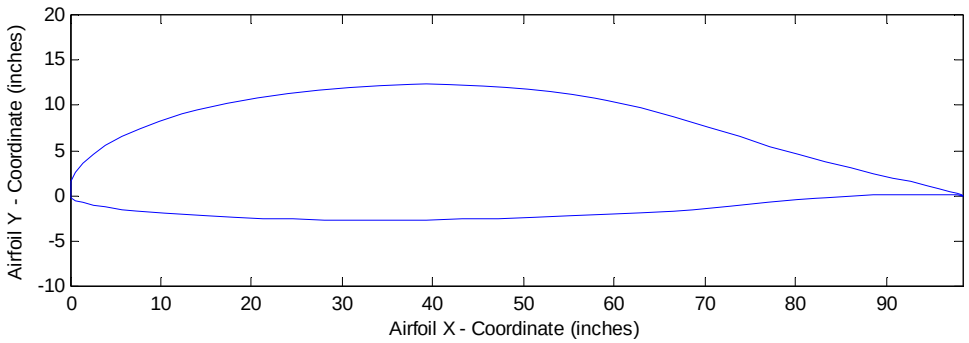

Figure 15. LRN-1015 Airfoil Geometry

The LRN-1015 airfoil drag polars in Figure 16 were generated at a Mach number of 0.50. XFOIL, being a two dimensional viscous force estimator, produces different drag estimates at different speeds. Mach numbers lower than 0.50 shifted each corresponding Reynolds number drag curve down, meaning lower drag values. Increasing Mach numbers beyond 0.50 shifted each drag curve up, resulting in higher drag values. However, the difference between Mach 0.50 and 0.60 was negligible for Reynolds numbers between 2.0e06 and 1.0e7. Since the AFRL model consistently operates within Mach numbers of 0.50 to 0.60 and Reynolds number of 2.0e06 and 1.0e07, this drag polar was assumed accurate throughout the flight profile.

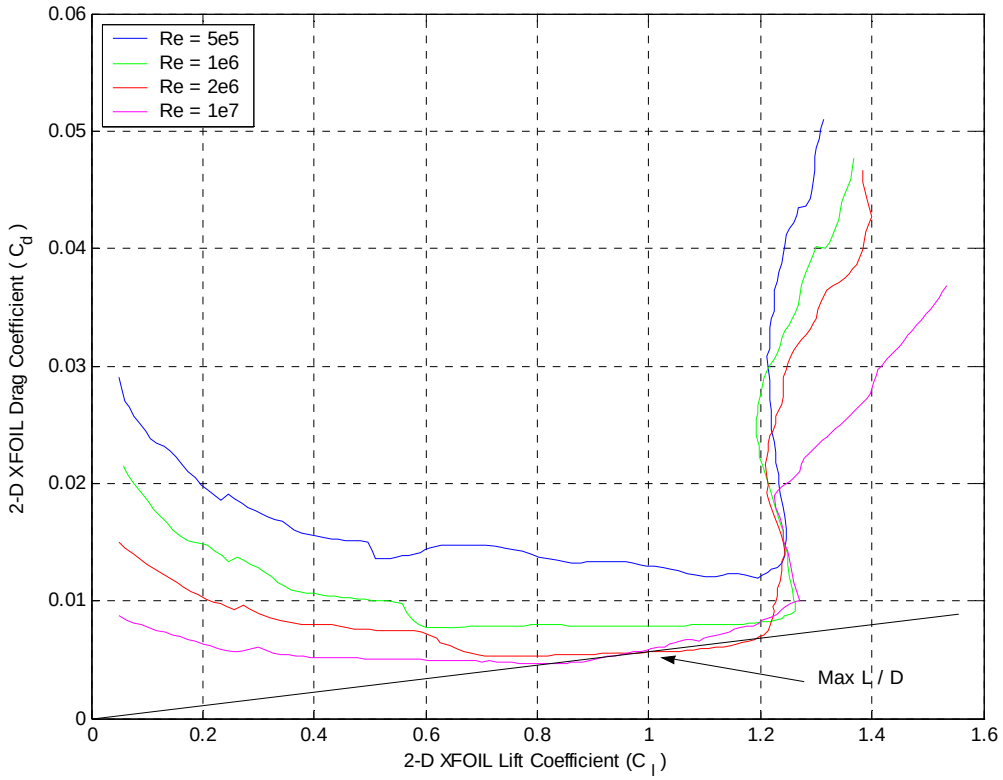

Figure 16. Two-Dimensional LRN-1015 Airfoil Drag Polar

2.9 The AFRL Mission Profile

Previous research has been based on a four point mission profile consisting of three segments (ingress, loiter, egress). The mission profile reflects the current Global Hawk surveillance mission requirements (Table 3).

The more points used in the mission profile, the more accurate the results at a cost of computational time. Initial calculations concluded that utilizing just three segments of a flight profile produced erroneous results and adding a few points increased accuracy significantly. Therefore, three more points were added to the baseline mission profile resulting in a six segment profile. Also, several trade studies were conducted in order to

optimize fuel consumption with this configuration at these flight conditions and the baseline profile was slightly modified to incorporate the results. Throughout this research, the seven-point mission profile shown in Table 4 was used for the AFRL model drag assessment.

Table 3. Baseline AFRL Mission Profile

Mission Leg	Range (miles)	Duration	Altitude (ft)	Velocity (Mach)
Ingress	3000	N/A	50,000	0.60
Loiter	N/A	24 hours	65,000	0.60
Egress	3000	N/A	50,000	0.60

Table 4. Modified AFRL Mission Profile

Measured	Ingress		Loiter			Egress	
Parameter	Point 1	Point 2	Point 3	Point 4	Point 5	Point 6	Point 7
Time (hrs)	0.67	4.83	9.00	21.00	33.00	35.00	41.33
Range (miles)	0	1,526	3,080	7,634	12,266	13,039	15,442
Altitude (ft)	50,000	56,500	60,000	66,500	70,000	60,000	50,000
Velocity (fps)	532.4	542.0	551.7	561.4	571.1	561.4	551.7
Mach	0.55	0.56	0.57	0.58	0.59	0.58	0.57
Re_{wing}	5.4e06	4.0e06	3.4e06	2.6e06	2.2e06	3.5e06	5.5e06

2.10 The AFRL Joined-Wing Joint Section Geometry

The wing joint section of the AFRL model was expected to create problems throughout this study due to its complex airfoil geometry. The model displays a poor unification between the forward and aft wing airfoil sections. The baseline configuration utilized a simple merging of the two airfoils, creating a single airfoil consisting of two

LRN-1015 sections connected end-to-end as shown in Figure 17. This ultimately leads to poor flow solutions about this section and high disturbances (Figure 18), resulting in abrupt changes in aerodynamic parameters.

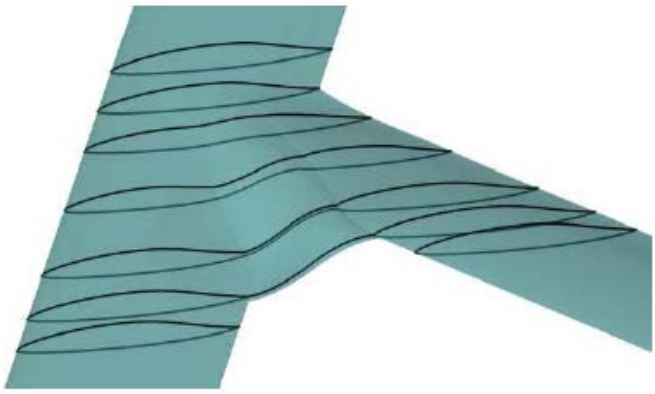

Figure 17. AFRL Configuration Wing Joint Section [30]

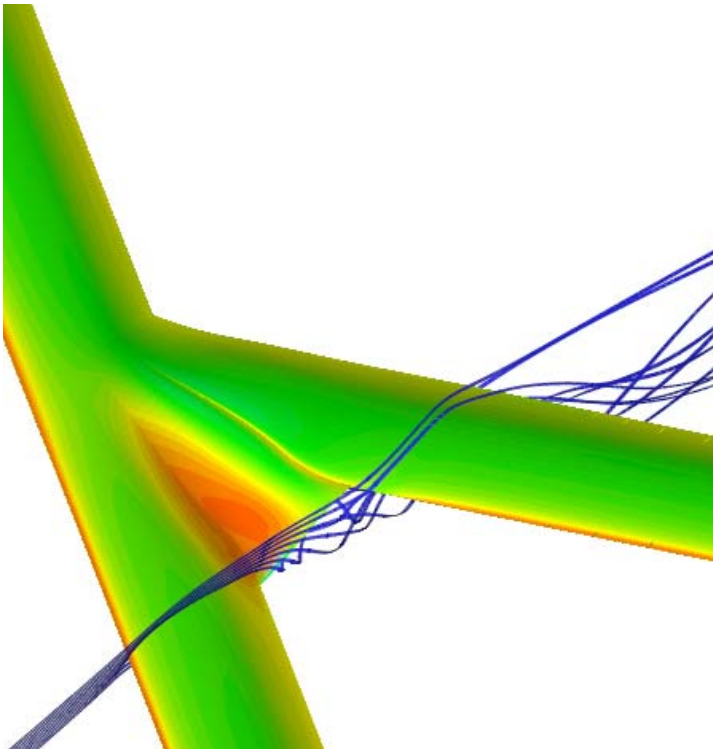

Figure 18. AFRL Wing Joint CFD Solution (Contours Colored by Pressure) [30]

III. Methodology

3.1 Introduction

This chapter presents in detail the methodology for each of the drag buildup methods used in this research. It will thoroughly discuss the assumptions applied in each process and possible errors that the results could display. First it will describe the AVTIE and Pan Air environments in detail and the trimming process utilized throughout the mission profile. Caution was exercised when working with the AVTIE environment. Modifications to the environment requires complex understanding of object oriented software programming. The software calculated the forces acting on the model using various methods. Therefore, two different methods will be extrapolated from the AVTIE results. Overall, three main methods were utilized in order to determine the drag on the aircraft. These methods are the Roskam method (R), the Roskam/AVTIE strip method (RAs), and the Roskam/AVTIE Pan Air method (RApa).

The Roskam method will be based solely on the drag buildup procedure within the Roskam aircraft design series [2]. This method estimates parasite drag effects on the entire aircraft configuration. Since the AVTIE model consists of the wing only, the next two methods combine fuselage and vertical tail drag estimates from Roskam with the wing drag results from AVTIE. The Roskam/AVTIE strip method divides the wing structure into individual strips and sums the forces acting on each panel to determine the total averaged lift throughout each panel. Using spanwise lift coefficients for each panel, XFOIL is used to determine both parasite and induced drag. Each section is then added together to determine the forces acting on the whole wing, and then it is combined with

fuselage and vertical tail drag. The Roskam/AVTIE Pan Air method also utilizes an XFOIL strip method to determine parasite drag effects of the wing. However, induced drag is determined by Pan Air. Total wing drag is determined by the addition of parasite drag from XFOIL and induced drag from Pan Air. Total aircraft configuration drag is determined by incorporating the total wing drag with fuselage and vertical tail drag provided by Roskam.

3.2 Pan Air Aerodynamic Analysis

The Pan Air model used in this study is a continuation from that used by Blair and Canfield [24]. Pan Air is used to analyze inviscid flow about three dimensional objects. The joined-wing model for this study was subdivided into individual panel elements as shown in Figure 19 and Table 5.

Table 5. AFRL Configuration Wing Strip Division

Forward Inside Wing		Aft Wing		Joint Section		Outboard Wing	
Panel	Strip Numbers	Panel	Strip Numbers	Panel	Strip Numbers	Panel	Strip Numbers
0	0 - 7	1	0 - 7	2	0 - 3	3	0 - 15

In total, the wing was divided into 28 spanwise strips. The span of each strip depended on the location on the wing. More strips were applied at the tip, in the hope to accurately capture downwash effects. The forward inside and aft wings utilized the same strip distribution, much more vague that the fine distribution at the tip. The joint section only consisted of four spanwise strips.

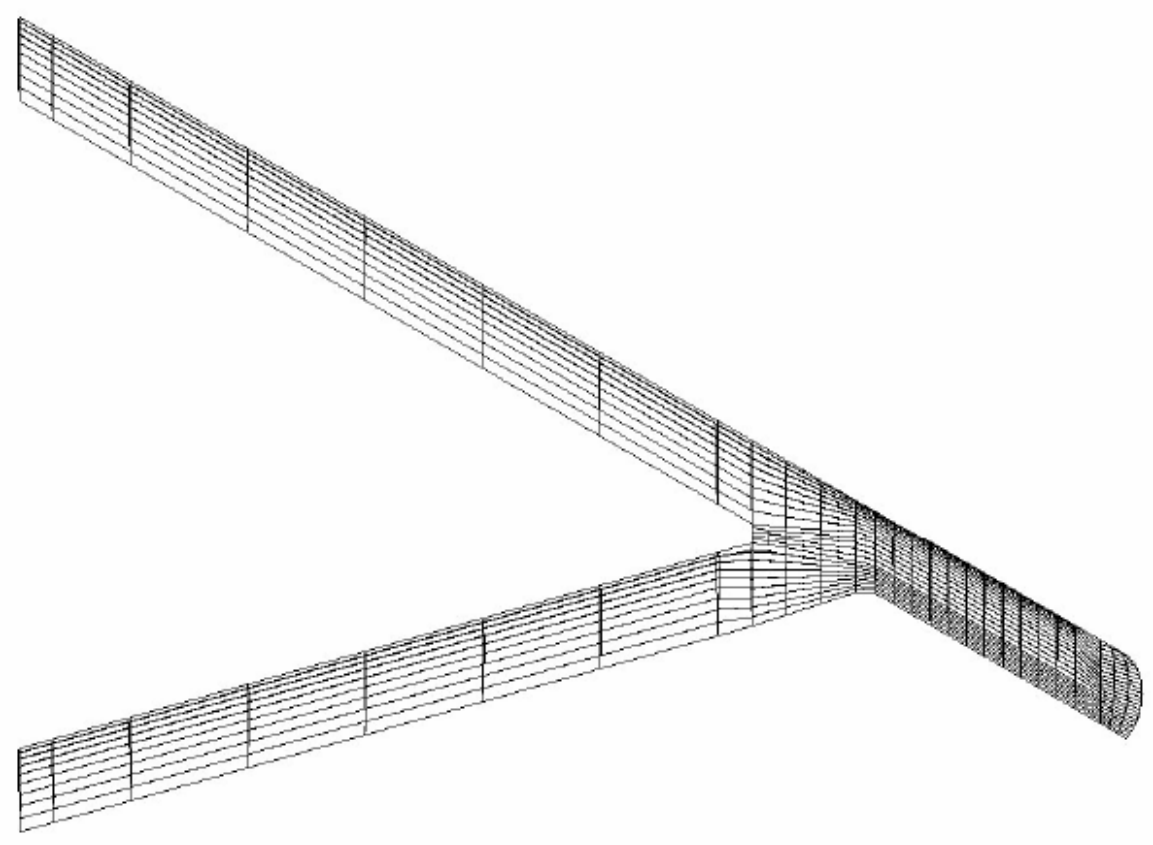

Figure 19. AVTIE Spanwise Strip Distribution

3.3 AVTIE Trim For Rigid Aerodynamic Loads

For the AFRL joined-wing configuration, aircraft angle-of-attack, aft wing twist, and fuel distribution control longitudinal trim. Note that aft wing twist only provides pitch trim control and does not effect any other axial translations. Additional control surfaces are used for roll and yaw control. The aft wing is rotated at the wing root intersection with the vertical tail and remains rigid at the wing joint with the main wing with a linear distribution between (Figure 20). An un-modeled actuator in the vertical tail controls the twist angle.

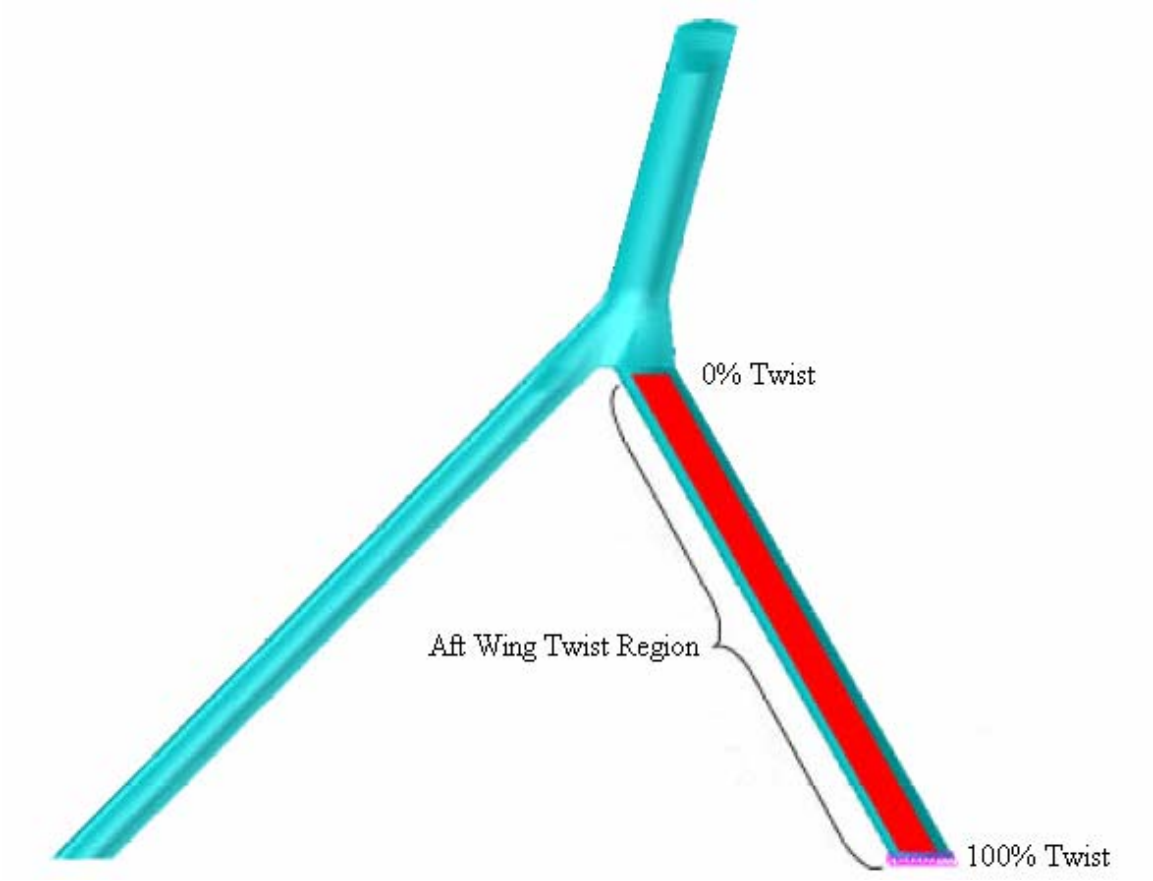

Figure 20. Linearly Tapered Aft Wing Twist Distribution

AVTIE uses a linear Taylor series approximation to compute a trimmed angle-of-attack (α) and aft wing root twist angle (δ) utilizing Equation (2), where lift is the load factor multiplied by the weight and the longitudinal moment of the aircraft is zero.

$$\begin{Bmatrix} C_L - C_{L_0} \\ C_M - C_{M_0} \end{Bmatrix} = \begin{bmatrix} \left(\frac{dC_L}{d\alpha}\right) & \left(\frac{dC_L}{d\delta}\right) \\ \left(\frac{dC_M}{d\alpha}\right) & \left(\frac{dC_M}{d\delta}\right) \end{bmatrix} \begin{Bmatrix} \alpha - \alpha_1 \\ \delta - \delta_0 \end{Bmatrix} \tag{2}$$

AVTIE first calls Pan Air to generate the aerodynamic coefficients and stability derivatives in Equation (2) using a finite difference procedure. After solving Equation (2) for the trimmed parameters, AVTIE then calls Pan Air to regenerate the pressure distributions at the trimmed conditions. The user must pay special attention to the aft wing root twist angle throughout the trimming process, as large angle-of-attack or twist angles will generate excessive drag and should be avoided if possible [22].

At each point within the mission profile of Table 4, Pan Air trims the model for steady wings level 1.0g flight. In order to trim properly, static stability requires that the center of gravity is forward of the aerodynamic center (the point where pitching moment remains constant), and proper pitch trim demands that the center of gravity is at the center of pressure. Using the location of the payload mass to adjust the center of gravity at the conclusion of the mission (point seven with zero fuel) aids the aircraft's ability to maintain a stable trim condition throughout the mission. This improves the aerodynamic performance at the trimmed condition by reducing the required angle-of-attack and twist angle. Equation (3) is used by AVTIE to calculate the shift in payload location to move the center of gravity to the aerodynamic center.

$$\left|X_{cg} - X_{ac}\right| \cdot \frac{\text{Total Mass}}{\text{Payload Mass}} = \Delta X_{cg} \tag{3}$$

Once the payload mass is shifted to an appropriate location, it is fixed at that location throughout the flight profile, and the location of the fuel can be used at the beginning of the mission to augment mass balancing of the aircraft. Adequate fuel management and distribution is utilized to force the center of gravity to lie within desired

locations throughout the mission profile when initial conditions no longer apply due to decreasing weight from fuel consumption.

3.4 The Roskam Method (R)

Roskam defines total drag as the sum of zero lift drag and drag due to lift. Drag due to lift is subdivided into induced drag and viscous drag due to lift terms. The induced drag (C_{Di}), also called trailing edge vortex drag, simply depends on the spanwise distribution of lift and is proportional to the square of the lift coefficient. This will be factored in later with other aerodynamic performance characteristics. Viscous drag due to lift results from the change in the boundary layer due to aircraft trim conditions, or when the airfoil's upper surface boundary layer thickness increases with increasing angle-of-attack (α). This in turn results in an increase in the so-called profile drag which itself is the sum of skin-friction drag and pressure drag [2], both of which are estimated by the Roskam drag buildup method. Therefore, according to the Roskam method, all factors of drag will be estimated with the exception of induced drag. For simplicity, this thesis will define all zero lift drag and viscous drag due to lift as parasite drag, and induced drag will be addressed as is. Throughout the Roskam method, lift was simply determined to equal the weight of the aircraft, simulating steady level 1.0g flight throughout the entire flight profile.

Roskam determines aircraft drag by breaking down the model into sections. The MATLAB code used for this method broke the AFRL model down into five components, the forward inside wing (FIW), the aft wing (AW), the forward outside wing (FOW, sometimes addressed as outboard wing), the vertical tail, and the fuselage. All parasite

drag acting on the model can be summed up in component form as shown in Equation (4).

$$C_{Dp_{A/C}} = C_{Dp_{FIW}} + C_{Dp_{FOW}} + C_{Dp_{AW}} + C_{Dp_{fuse}} + C_{Dp_{tail}} \quad (4)$$

The methods used to calculate the subsonic parasite drag effects on the forward inside wing, aft wing, and forward outside wing are exactly the same and are computed by Roskam using the relationship

$$C_{Dp_{wing}} = \left(R_{WF}\right)\left(R_{LS_{wing}}\right)\left(C_{F_{wing}}\right)\left\{1 + L'\left(t/c\right)_M + 100\left(t/c\right)_M{}^4\right\} S_{wet_{wing}} / S_{wing} \quad (5)$$

where R_{WF} is the wing-fuselage interference factor, R_{LS} is the lifting surface correction factor, C_{Fw} is the turbulent flat plate friction coefficient, L' is the airfoil thickness location parameter, t/c is the maximum thickness-to-chord ratio, and S_{wet} and S are the wetted area and area of the wings respectively. According to Roskam, this relationship is applicable to all wing and airfoil geometries.

$$f_{wing} = \left(R_{WF}\right)\left(R_{LS_{wing}}\right)\left(C_{F_{wing}}\right)\left\{1 + L'\left(t/c\right)_M + 100\left(t/c\right)_M{}^4\right\} S_{wet_{wing}} \quad (6)$$

In order to add each component of the aircraft to account for total aircraft parasite drag, each section will need to be translated into equivalent parasite areas, commonly give the abbreviation *f*. For each of the wing components, this is determined by

multiplying Equation (5) by the wing planform area, resulting in Equation (6). Each of these parameters, except *L'*, are found by using detailed charts within Roskam's text (Figure 21, Figure 22, and Figure 23).

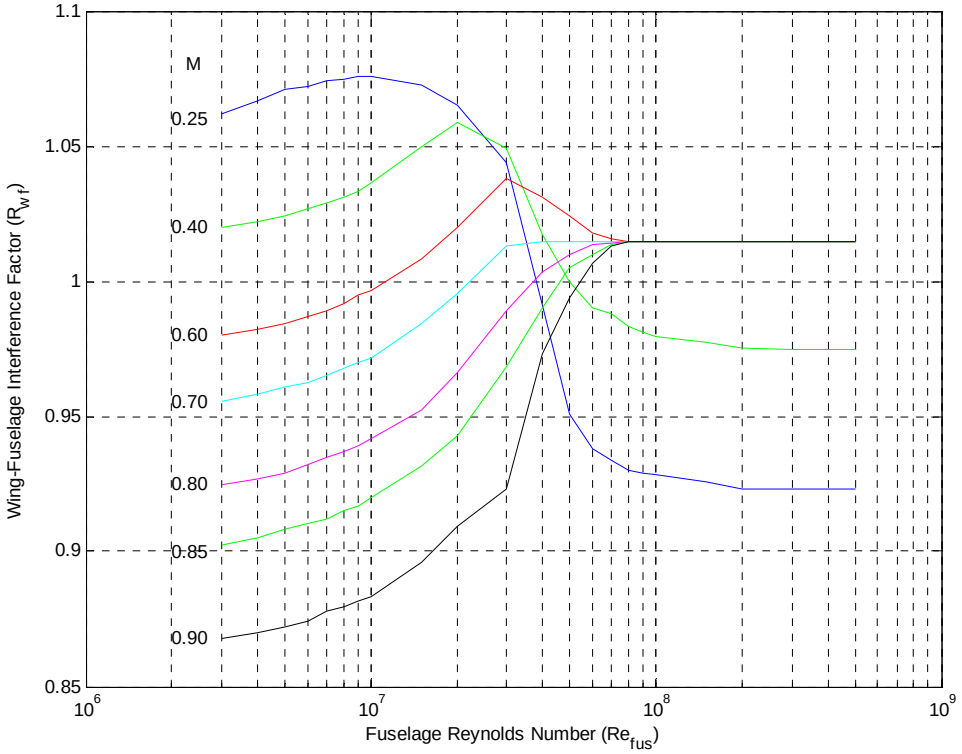

Figure 21. Wing-Fuselage Interference Factor

These charts were coded into MATLAB and fitted results were determined using linear interpolation methods. Each point of the flight profile resulted in different Reynolds numbers, due to varying Mach numbers throughout flight, but on average a Reynolds number of 3.8e06 occurred at each wing section. Although the joint wing section chord is larger than the other wing sections, all calculations were based on a constant Reynolds number throughout the wing span for each mission point.

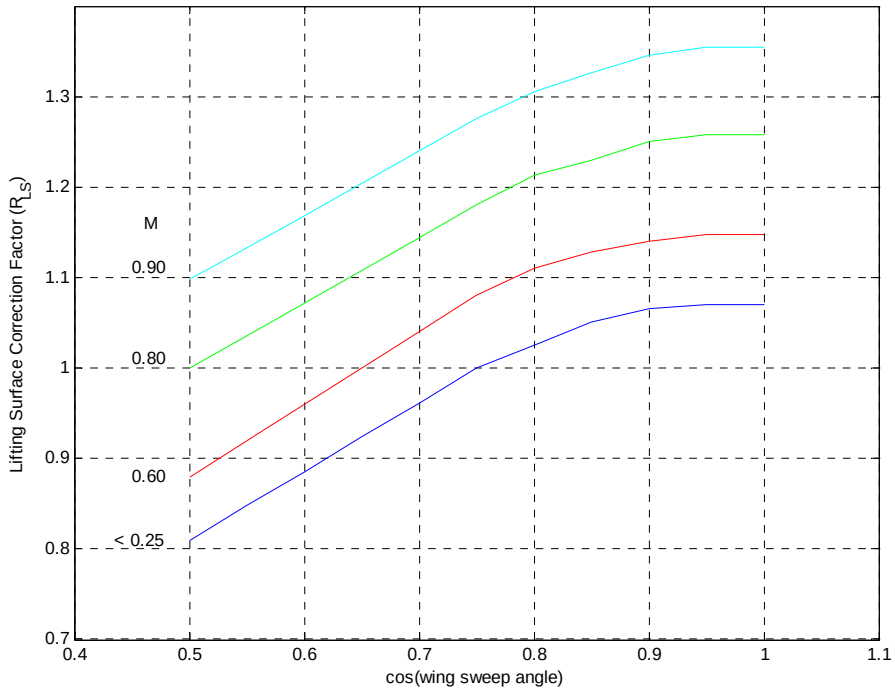

Figure 22. Lifting Surface Correction Factor

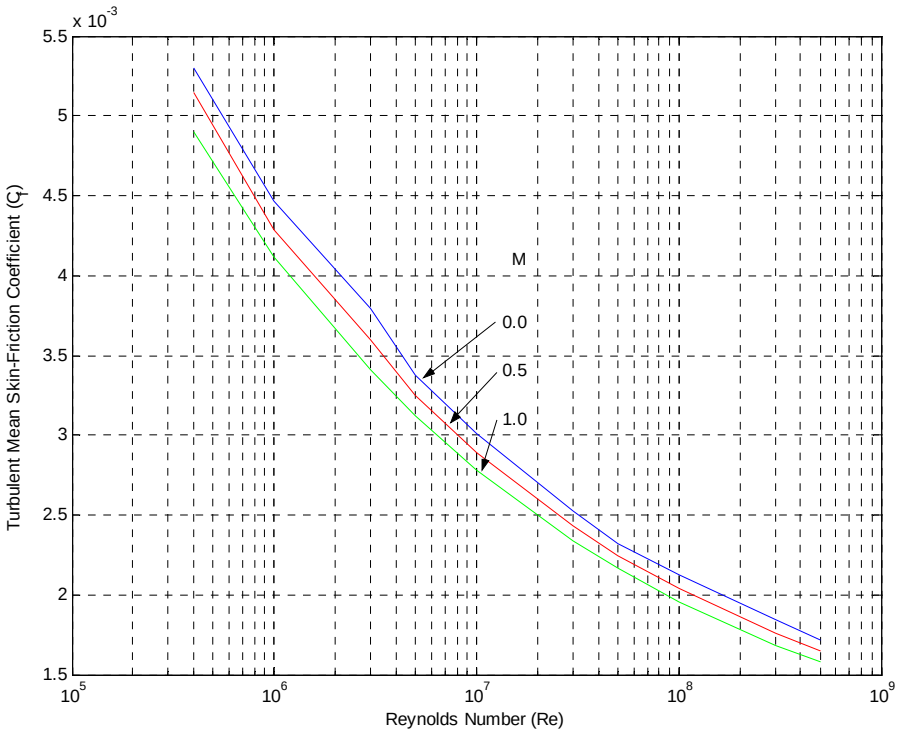

Figure 23. Turbulent Mean Skin-Friction Coefficient

In Equation (6), the airfoil thickness location parameter (L') is determined by the chord distance to the maximum *t/c* location. If the max *t/c* location is greater than or equal to 30% chord, L' is given a value of 1.2. If less than 30% chord, L' is set to 2.0. The LRN-1015 airfoil is only one of many candidate airfoils that may be used on the joined-wing sensor craft. In order to accurately estimate the drag on many possible airfoils, a simple average between these two values is used. The wetted area of the wings was estimated by Roskam with Equation (7)

$$S_{wet} = 2S_{net}\left\{1 + 0.25\left(t/c\right)_r \frac{1+\tau\lambda}{1+\lambda}\right\} \tag{7}$$

where λ is the taper ratio and τ represents the ratio between the *t/c* at the tip to the *t/c* at the root. For the joined-wing configuration, this term simply becomes unity, and the S_{net} was replaced with individual wing section areas (S_{FIW}, S_{FOW}, S_{AW}). The factor of two accounts for the wing on both sides of the aircraft, as FIW, FOW, and AW refer to just one side of the aircraft.

Roskam did not consider forward swept wing aircraft in the text. Since the joined-wing design has a forward swept aft wing, an assumption was made that a wing swept forward 30 degrees would have the same R_{LS} factor as one swept back 30 degrees. Parasite drag effects on the vertical tail were also estimated using Equation (4) in a similar fashion with each of the wing sections. The only difference is the wing-fuselage interference factor is preset to 1.0.

Roskam also divides fuselage drag into zero lift fuselage drag and fuselage drag due to lift. As previously mentioned, the fuselage is never accounted for in lift

calculations and all induced drag is assumed to act on the wings only. Therefore, all drag forces acting on the fuselage is parasite drag and is modeled by Roskam as

$$C_{Dp_{fuse}} = \left(R_{WF_{fuse}}\right)\left(C_{F_{fuse}}\right)\left\{1 + 60/\left(l_f / d_f\right)^3 + 0.0025\left(l_f / d_f\right) S_{wet_{fuse}} / S_{wing}\right\} \quad (8)$$

where l_f and d_f are the length and maximum diameter of the fuselage respectively. The wetted area of the fuselage was simply calculated using the equation for the surface area of a cylinder. Although this estimate will be higher than actual, it will allow for a small safety factor in fuel consumption. In order to add this component of parasite drag to that of the wing sections, it also has to be translated into an equivalent parasite area by multiplying Equation (8) by the wing planform area.

$$f_{fuse} = \left(R_{WF_{fuse}}\right)\left(C_{F_{fuse}}\right)\left\{1 + 60/\left(l_f / d_f\right)^3 + 0.0025\left(l_f / d_f\right) S_{wet_{fuse}}\right\} \quad (9)$$

At this point, equivalent parasite areas for each of the aircraft components have been determined. These equivalent parasite areas are additive and the parasite drag for the entire aircraft configuration is determined by simply dividing out the wing planform area as shown in Equation (10).

$$C_{Dp_{A/C}} = \frac{f_{FIW} + f_{FOW} + f_{AW} + f_{tail} + f_{fuse}}{S_{wing}} \quad (10)$$

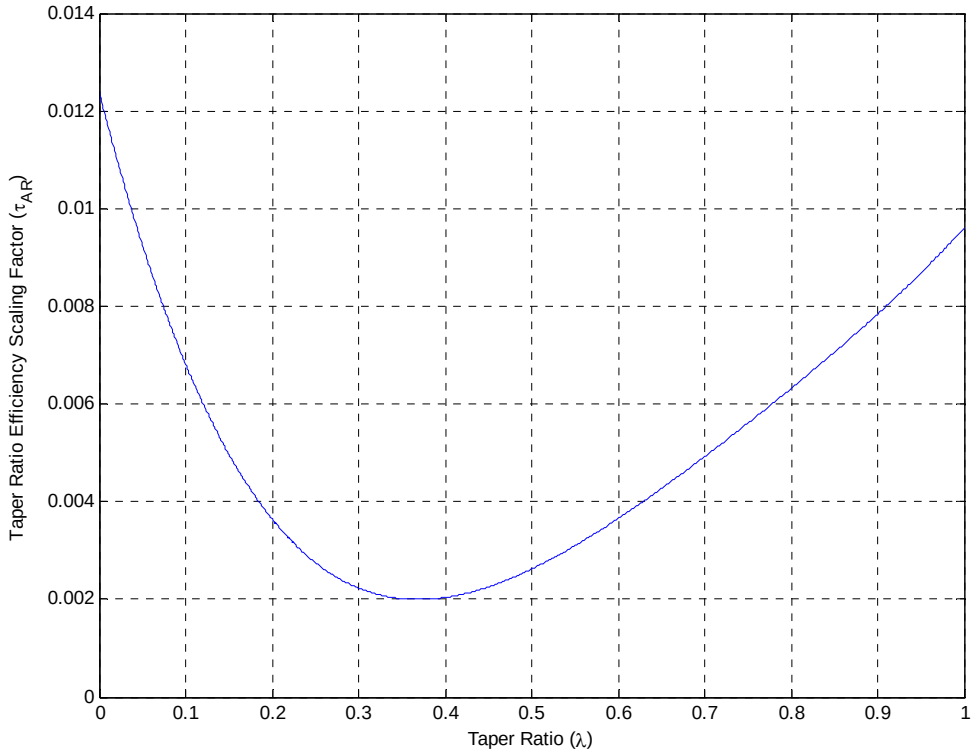

Figure 24. Taper Ratio Efficiency Calculation

Induced drag effects can be estimated using many methods. For the Roskam drag buildup method, the induced drag acting on the wing was calculated using an equation from Saarlas [31]

$$C_{Di} = \frac{C_L^2}{\pi AR} + \frac{C_L^2}{\pi AR}\tau \tag{11}$$

where AR is the aspect ratio of the aircraft and τ is a span efficiency scaling factor determined from Equation (12) using Figure 24. This factor is most notably recognized in the span efficiency factor relationship shown in Equation (13). This relationship for induced drag is based on an elliptical lift distribution for a single lifting surface, although

the joined-wing concept divides its lift force between two lifting surfaces, the forward and aft wings.

$$\tau = (AR)(\tau_{AR}) \tag{12}$$

$$e_{span} = \frac{1}{1+\tau} \tag{13}$$

Again, these equations have been formulated and validated throughout the years for conventional aircraft configurations. Applying these relationships to the radical joined-wing design may not produce accurate drag estimates. However, there are currently no formulations that relate lift coefficients to induced drag for unconventional wing planform configurations such as the AFRL joined-wing sensor craft.

3.5 The Roskam/AVTIE Strip Method (RAs)

Throughout the description of this method, refer to Figure 25 for airfoil nomenclature and Table 6 for the corresponding parameters. The Roskam/AVTIE strip method divides the wing structure into individual strips, as shown in Figure 19. However, AVTIE is only used to extract lift coefficient values from Pan Air for each section. The objective is to use spanwise lift distribution predicted with inviscid theory and extract an accurate drag assessment.

The goal of the Roskam/AVTIE strip method is to measure and calculate the lift and drag forces and represent them in the same coordinate frame. The freestream frame,

or the V_∞ frame, will be used as the primary frame to represent lift and drag on the airfoil; therefore, all forces must be represented and projected onto the L_∞ and D_∞ coordinate system.

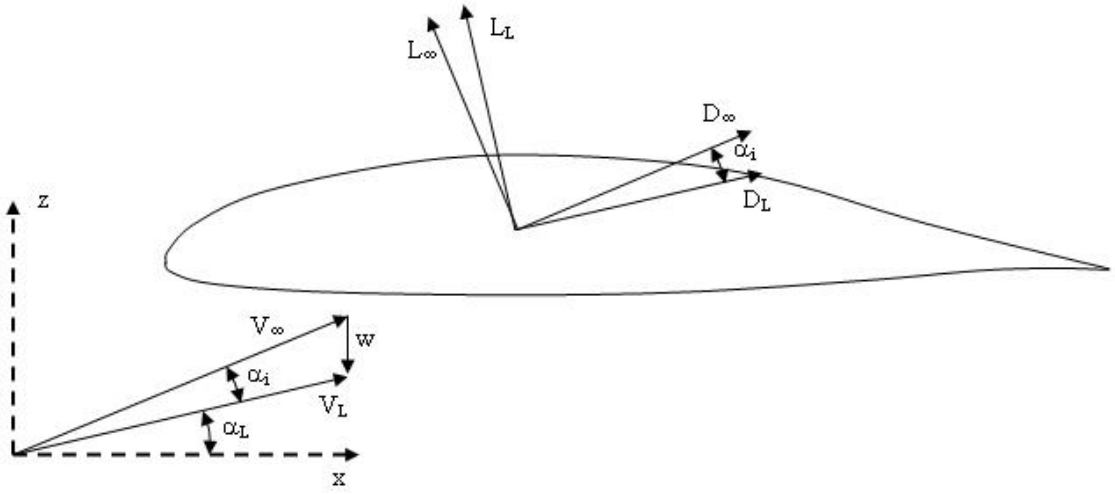

Figure 25. Roskam/AVTIE Strip Method Airfoil Nomenclature

Table 6. Roskam/AVTIE Strip Method Airfoil Definitions

Term	Definition
x, z	Coordinate frame of airfoil
V_∞	Velocity relative to freestream
V_L	Local velocity (V_∞ plus downwash component)
w	Local downwash component due to spanwise effects
α	Freestream angle of attack
α_L	Local angle of attack
α_i	Induced angle of attack ($= \alpha - \alpha_L$)
L_L	Local lift oriented with local velocity vector
D_L	Local drag oriented with local velocity vector
C_{LL}	Local lift coefficient oriented with local velocity vector
C_{DL}	Local drag coefficient oriented with local velocity vector
L_∞	Component of lift oriented with respect to freestream V_∞
D_∞	Component of drag oriented with respect to freestream V_∞

The first step in the strip method is to calculate the lift on each airfoil section. Since the local angle-of-attack (α_L), which is a function of induced downwash, is still an unknown parameter, assume an angle-of-attack relative to freestream (α) when integrating forces about the airfoil. This assumption implies the local lift coefficient (C_{LL}) is identical to the lift coefficient with respect to the freestream frame ($C_{L\infty}$).

The assumption that the local lift coefficient is equivalent to the lift coefficient in the freestream frame was validated using two dimensional drag polar generated by XFOIL for the LRN-1015 airfoil, see Appendix A, section A.4. At a Reynolds number of 1.0e+07 and an angle-of-attack of seven degrees, XFOIL predicts a $C_{L\infty}$ of 1.31790 and a $C_{D\infty}$ of 0.02396. These values are based on zero downwash effects, which in turn imply the local coordinate frame and the freestream coordinate frame are the same. If this same airfoil section, still with an angle-of-attack of seven degrees and Reynolds number 1.0e+07, is subjected to a downwash angle of five degrees, the local frame is rotated clock-wise. The corresponding lift coefficient is found by doing the calculation:

$$C_{LL} = C_{L\infty} \cos(-5°) - C_{D\infty} \sin(-5°) = 1.3128 + 0.0021 = 1.3149$$

This shows the rotated (correct) value of $C_{LL} = 1.3149$ is nearly identical to a $C_{L\infty}$ value of 1.3179 (0.22% error), sufficient for this research. Although other assumed induced angles-of-attack may increase the error, the results are negligible. Therefore, assuming $C_{LL} \approx C_{L\infty}$ for all angles-of-attack is an excellent approximation.

This closely approximated lift component (C_{LL}) is then used to look up the associated local drag coefficient (C_{DL}) and its corresponding local angle-of-attack (α_L)

from the two dimensional drag polar data, Appendix A, section A.4. Knowing the aircraft's trimmed angle-of-attack (α), including wing twist, and the angle-of-attack the airfoil actually experiences (α_L), the induced angle-of-attack can be determined from Equation (14). This induced angle is the amount the measured C_{LL} and C_{DL} for each individual panel must be rotated in order to represent all forces in the freestream frame.

$$\alpha_i = (\alpha - \alpha_L) \tag{14}$$

When rotating C_{LL} and C_{DL} back into equivalent $C_{L\infty}$ and $C_{D\infty}$ components, $C_{D\infty}$ absorbs a large component of lift from C_{LL}. This component of $C_{D\infty}$ is the elusive induced drag. The parasite drag of the section is the projection of C_{DL} back onto $C_{D\infty}$, which is slightly less in magnitude, and adding both induced and parasite drag forces results in the total drag force in the freestream frame for each individual spanwise strip.

This procedure is applied to each individual section of the wing structure in Figure 19, even to the four strip sections of the joint section consisting of complex airfoil geometry. At the joint section, the table lookup procedure with XFOIL is assuming an LRN-1015 airfoil, which is not the case. This will be a source of error with this approach, but the four strips of the wing joint section is just a small portion of the total drag on the aircraft and these small errors can assumed negligible.

Each panel is then summed together resulting in total lift and drag (parasite and induced) acting on the joined-wing. This method was determined utilizing MATLAB and relied solely on Pan Air lift coefficient values and the linear wing twist distribution from AVTIE in order to determine freestream angle-of-attack (α) with respect to the

airfoil's reference frame (x, z frame). These drag values were then combined with Roskam fuselage and tail drag estimates to predict total aircraft drag.

3.6 The Roskam/AVTIE Pan Air Method (RApa)

The Roskam/AVTIE Pan Air Method accounts for wing parasite drag using the same procedure as outlined in the Roskam/AVTIE strip method. However, induced drag is not determined individually by strips using two dimensional tabulated XFOIL data for the LRN-1015 airfoil as done previously. Instead, this method relies on Pan Air inviscid predictions about the joined-wing model. Since Pan Air determines inviscid forces about arbitrary three dimensional shapes, all of the calculated drag is in fact the induced drag.

At each point of the flight profile, AVTIE archives drag data that includes the Pan Air induced drag for the entire joined-wing structure. This value is a single value for the whole wing configuration and is not documented as individual strips along the wing as within the strip method. To estimate drag on the wing configuration, this value is summed with parasite drag results from the strip method for each panel. Total aircraft drag is found by combining wing drag from XFOIL and Pan Air with the fuselage and vertical tail drag estimates provided by Roskam.

3.7 Aerodynamic Performance Calculations

With both parasite and induced drag estimates from two different AVTIE methods and the Roskam method, other aerodynamic performance characteristics were computed using MATLAB. Similar to the Roskam method, the induced and parasite

drag components are not additive until all parasite drag effects have been accounted for. Aircraft parasite drag is determined by translating AVTIE parasite wing drag into an equivalent parasite area, and added to those for the fuselage and tail. The equivalent parasite area of the wings for each of the AVTIE methods is calculated by Equation (15)

$$f_{wings}\Big|_{meth} = \left(C_{D_P}\Big|_{meth}\right)(S) \tag{15}$$

where *meth* refers to a method used for wing drag estimation (RAs, RApa). This equivalent parasite area is now additive with the other parasite areas for the fuselage and vertical tail as demonstrated in Equation (10), where f_{wings} replaces the summation of f_{FIW}, f_{FOW}, and f_{AW}. This accounts for all parasite drag effects of the aircraft and is simply added to the induced drag inflicted on the wings to estimate total drag forces in the freestream frame (V_∞).

Fuel burn was determined using a specific Breguet range equation for propeller driven aircraft from Saarlas [31]

$$R = 375\left(\frac{\eta_p}{C}\right)\left(\frac{L}{D}\right)\ln\left(\frac{W_i}{W_{i+1}}\right) \tag{16}$$

where i represents a specific point within the flight profile, η_p represents a propeller efficiency factor (80% assumed for the AFRL configuration), R represents the range in miles, C represents specific fuel consumption in pounds per HP-hour (0.45 assumed throughout the flight profile), and W represents aircraft weight in pounds. A specific fuel

consumption of 0.45 is an estimate based on other HALE aircraft driven by a propeller. This equation was solved for W_{i+1} (Equation 17) and implemented into MATLAB to determine fuel burn throughout each segment of the flight profile.

$$W_{i+1} = \left(W_i\right) e^{\frac{-CDR}{375\eta_p L}} \tag{17}$$

The zero lift drag coefficient, or the parasite drag (C_{Do}), was found using Equation (18) from Saarlas [31]

$$C_D = C_{D_0} + k\left(C_L\right)^2 \tag{18}$$

where the spanwise induced drag constant k is defined

$$k = \frac{1}{\pi\left(\mathrm{AR}\right) e_{oswald}} \tag{19}$$

and the Oswald efficiency

$$e_{oswald} = \frac{1}{\pi\left(\mathrm{AR}\right)\mathrm{r} + 1 + \tau} \tag{20}$$

with τ representing the taper ratio efficiency factor determined in Equation (12) and r represents an efficiency scaling factor. An efficiency scaling factor of 0.010, a value

from an aircraft of similar size, was used for this study. Equation (18) produces a zero lift drag coefficient for each point of the flight profile, all very close in magnitude. To determine the overall zero lift drag coefficient, these values were averaged over each point of the profile.

IV. Results

4.1 Overview

This chapter will present and discuss the results from each of the three unique drag buildup methods. It will analyze the drag estimates from each method individually for a joined-wing not incorporating any aerodynamic twist. After close examination of each individual method, a brief overview will be conducted to compare each method. This chapter will conclude with an aerodynamic twist design for the AFRL configuration in an effort to optimize the wing for improved lift-to-drag ratios during cruise and loiter mission segments. Each method assumed an initial weight at point one of the flight profile to be 1,000 pounds less than that in Table 1 to account for fuel consumption from takeoff to 50,000 feet.

Since two different interfaces were used in order to determine aerodynamic performance (MATLAB and AVTIE), an iterative process was employed to converge on similar fuel consumption results from both programs. All results discussed and tabulated in the appendices refer to final converged solutions. The Roskam method is exempt from this iterative process since it is computed solely by MATLAB.

The flight profile within the AVTIE code was modeled slightly differently than that within the MATLAB code. The MATLAB code was based solely on a point-by-point method for each segment of the flight profile, based on Table 4. AVTIE was constructed relying more on segment information (distance between points, Mach number throughout segment, etc.) These segments were numbered 0 – 5 and were able to

be subdivided by fractions in order to measure aerodynamic performance at any location along the segment. To identify the location of the model within the profile, AVTIE uses a mission leg and fraction identifier, displayed as mission segment (0 – 5) and percentile completed (0 – 99 %) of the leg.

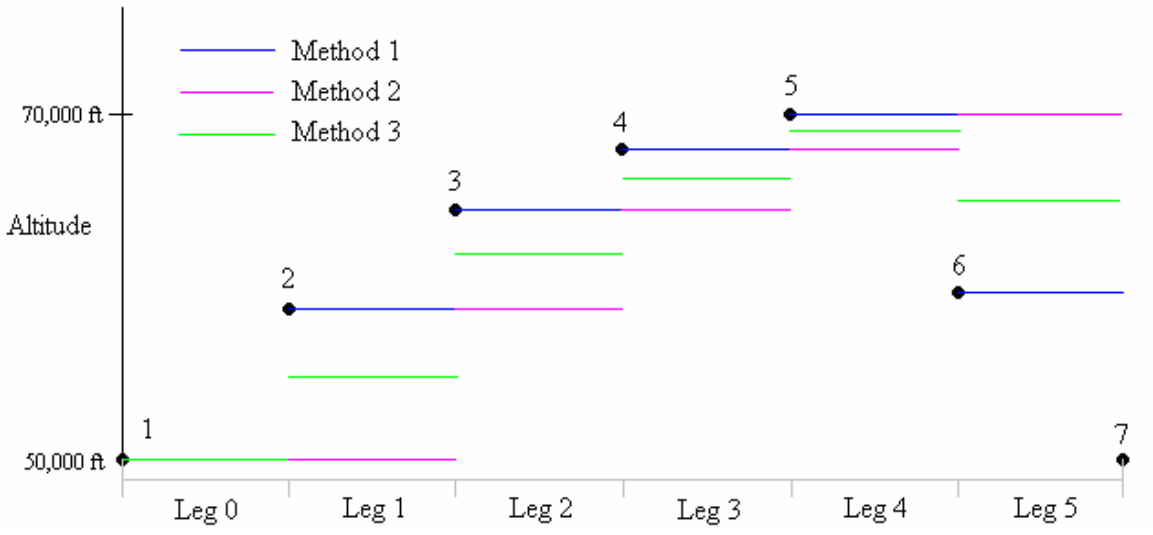

Figure 26. AVTIE Output Selection

In order to compare results with MATLAB's point-by-point method, three choices of AVTIE outputs are available (Figure 26). For aerodynamic data at the fourth point of the flight profile the user could choose to trim the aircraft at either mission leg 0, 99% complete (method 2, magenta line), or at mission leg 1, 0% complete (method 1, blue line). The weight of the model at each of these points is nearly identical, but difference in altitude and airspeed produces different results. The third choice (method 3, green line) would be a simple average between these two methods. The individual lines refer to where aerodynamic trim calculations for the corresponding leg were calculated and applied throughout the segment and are not to be confused with the actual altitude

throughout the leg. In order to eliminate redundant results, this chapter will only present data obtained using method 3, as it was a more creditable method to model aerodynamic performance between the high and low estimates from methods 1 and 2.

4.2 Roskam Method Results

The Roskam method was expected to produce fair results since it is based strictly on historical data from previously experimented aircraft configurations. But again, the results from this method need to be supplemented by others, since the joined-wing concept is radically different from any aircraft configurations Roskam intended to evaluate.

Equation (6) is used to estimate equivalent parasite areas for each of the wing sections (FIW, FOW, and AW). As shown in Figure 21 through Figure 23, all the parameters in this equation are a function Reynolds number, Mach number, and wing sweep, and will change throughout the flight profile. Therefore, equivalent parasite values will be determined at each point of the flight profile. Although the wing fuselage interference factor applies to just the forward inside wing and aft wing, it was also used for the forward outside wing to estimate disturbance factors at the wing joint. The results for the forward inside wing, forward outside wing, aft wing, and vertical tail are shown in Table 7 through Table 10, where the wing-fuselage interference factors, lifting surface correction factors, and skin friction coefficients were determined from Figure 21, Figure 22, and Figure 23 respectively.

Table 7. Forward Inside Wing Drag Correction Factors

Mission Point	R_{WF}	R_{LS}	C_F
1	1.0093	1.1209	0.0033
2	1.0162	1.1218	0.0034
3	1.0221	1.1228	0.0035
4	1.0343	1.1237	0.0036
5	1.0364	1.1247	0.0037
6	1.0225	1.1237	0.0035
7	1.0116	1.1228	0.0033

Table 8. Forward Outside Wing Drag Correction Factors

Mission Point	R_{WF}	R_{LS}	C_F
1	1.0093	1.1209	0.0033
2	1.0162	1.1218	0.0034
3	1.0221	1.1228	0.0035
4	1.0343	1.1237	0.0036
5	1.0364	1.1247	0.0037
6	1.0225	1.1237	0.0035
7	1.0116	1.1228	0.0033

Table 9. Aft Wing Drag Correction Factors

Mission Point	R_{WF}	R_{LS}	C_F
1	1.0093	1.1347	0.0033
2	1.0162	1.1349	0.0034
3	1.0221	1.1351	0.0035
4	1.0343	1.1354	0.0036
5	1.0364	1.1356	0.0037
6	1.0225	1.1354	0.0035
7	1.0116	1.1351	0.0033

Table 10. Vertical Tail Drag Correction Factors

Mission Point	R_{WF}	R_{LS}	C_F
1	1.0000	0.9280	0.0027
2	1.0000	0.9298	0.0029
3	1.0000	0.9316	0.0030
4	1.0000	0.9335	0.0031
5	1.0000	0.9353	0.0032
6	1.0000	0.9335	0.0029
7	1.0000	0.9316	0.0027

Equation (9) is used to determine the equivalent parasite area for the fuselage component of the aircraft. The wing fuselage interference factor (R_{WF}) is preset to unity since this equation represents fuselage drag only. Also, the fuselage does not include a lifting surface correction factor, since it is assumed all lift is produced by the wings only. Table 11 shows the drag correction factor results for the fuselage.

Table 11. Fuselage Drag Correction Factors

Mission Point	R_{WF}	C_F
1	1.0000	0.0022
2	1.0000	0.0023
3	1.0000	0.0023
4	1.0000	0.0024
5	1.0000	0.0024
6	1.0000	0.0023
7	1.0000	0.0022

An additional equivalent parasite area was added to those of the fuselage, tail, and wing sections to model body imperfections, rivets, and other sources of aircraft parasite drag not included in Roskam's drag buildup. This corrective drag term was estimated

from aircraft of similar size and wetted area. Using results in Table 7 through Table 11, and the assumed equivalent parasite area for body imperfections, Table 12 is generated to show the point-by-point equivalent parasite areas for each of the aircraft's components.

Table 12. Equivalent Parasite Area Breakdown

Point	f_{FIW}	f_{FOW}	f_{AW}	f_{TAIL}	f_{FUSE}	f_{IMPERF}	f_{TOTAL}
1	13.8810	4.2711	14.0518	5.5106	20.1501	0.2000	58.0645
2	14.7005	4.5232	14.8719	5.8670	20.8205	0.2000	60.9831
3	15.1157	4.6510	15.2823	6.0759	21.2351	0.2000	62.5599
4	15.8089	4.8643	15.9730	6.5000	21.8798	0.2000	65.2259
5	16.4492	5.0613	16.6094	6.6688	22.1538	0.2000	67.1425
6	15.1166	4.6513	15.2735	6.0726	21.2074	0.2000	62.5214
7	13.9101	4.2800	14.0634	5.5109	20.1103	0.2000	58.0748

As one would expect, the fuselage is responsible for the majority of the aircraft's parasite drag since it is the largest individual component of the aircraft. However, the total equivalent parasite area of the entire wing structure (FIW+AW+FOW) will surpasses that of the fuselage alone.

The total parasite drag of the aircraft is found by dividing out wing planform area from the equivalent parasite areas. Assuming lift equals weight and applying the induced drag relationship from Saarlas [31] in Equation (11), aerodynamic performance can be evaluated throughout the flight profile. Table 13 shows final drag coefficient estimates from Roskam, lift coefficients, and remaining fuel for each point of the flight profile.

The Roskam method initially consumed more fuel than previous AFRL joined-wing configurations allotted. Therefore, the fuel load was increased by 5,500 kg, which

ultimately increased drag, but not at the rate it increased range and time aloft. Final results show the Roskam method determined the aircraft was just barely able to accomplish its mission with just 11.60 pounds of fuel remaining at the conclusion of the mission.

With the extremely high aspect ratio wings of the joined-wing concept, lift-to-drag ratios in the 20's is expected, as 24 has been assumed throughout all previous drag estimations. The results from the Roskam approach show L/D ratios in the low 20s. This gives credibility to the Roskam approach as it was initially met with speculation to estimate the drag of a radical configuration such as the AFRL joined-wing.

Table 13. Roskam Method Drag Results

Point	C_L	C_{Dp}	C_{Di}	C_{Dtotal}	L/D	Fuel (lbs)
1	0.5592	0.0174	0.0076	0.0250	22.3411	64,317.66
2	0.6655	0.0183	0.0108	0.0291	22.8844	54,970.32
3	0.6864	0.0187	0.0115	0.0302	22.6969	46,582.62
4	0.6707	0.0195	0.0110	0.0305	21.9744	26,249.08
5	0.5590	0.0201	0.0076	0.0277	20.1474	10,549.07
6	0.3377	0.0187	0.0028	0.0215	15.6918	8,191.10
7	0.1718	0.0174	0.0007	0.0181	9.4789	11.60

The drag estimated by Roskam was predominately parasite drag, nearly double induced drag throughout the loiter segments of the profile. Again, these induced drag estimates are determined from Equation (11) which is a function of aspect ratio. This relationship is applicable to conventional aircraft configurations. The last point of the profile is by far the worst evaluation of the aircraft's performance. However, this is at the

conclusion of the flight profile and the values for mission point seven are not used in any fuel burn calculations for upcoming mission segments and can be neglected.

4.3 Roskam/AVTIE Strip Method Results

Extreme care had been invested in formulating the methodology behind this approach of drag assessment about an airfoil section. This approach divided the wing structure into individual sections. However, instead of relying on AVTIE to determine parasite drag, MATLAB code was developed to perform a table lookup and interpolation process from previously generated drag polar data for the LRN-1015 airfoil. The lift coefficient from AVTIE, calculated by Pan Air, and the freestream AOA were the only variables from AVTIE outputs utilized in MATLAB interpolations. Since all parameters are measured by individual strips, spanwise performance can be evaluated.

For HALE aircraft, the loitering segments of the flight profile are the most crucial to aircraft endurance. For the modified AFRL mission profile (Table 4) the fourth point of the profile is the mid-point of the loitering segments. Desired performance at this point generally implies desired overall performance throughout the other loitering segments. Therefore, spanwise analysis will be conducted only for the fourth point of the flight profile, which is at an altitude of 66,500 feet, Mach 0.58, and the aircraft has flown 21 hours of its mission. Spanwise aerodynamic performance is tabulated in Appendix B for each individual wing strip section for each point of the flight profile. For comparison between MALTAB and AVTIE, strip data from AVTIE for the fourth point of the flight profile is shown in Appendix C.

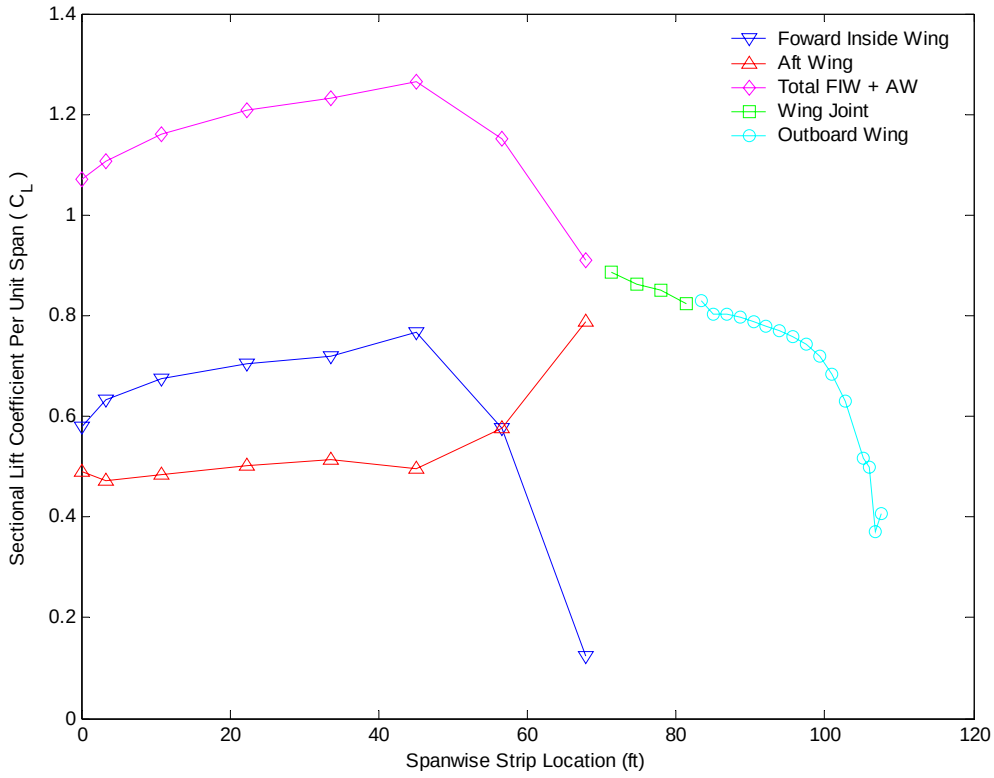

Figure 27. Roskam/AVTIE Strip Method Spanwise Lift Coefficient Distribution

Traditional aircraft design focuses on elliptical lift distribution in order to provide good aerodynamic performance. Figure 27 shows the Pan Air spanwise lift coefficient distribution supplemented by the spanwise lift force distribution of Figure 28. Although these lift distributions do not display a "perfect" elliptical distribution, favorable characteristics are shown with the inboard wing sections providing the majority of the lift, and lift decreasing spanwise from the fuselage. The first airfoil section actually lies on the fuselage centerline, within the fuselage model, and can be neglected. The second airfoil section represents the joint between the wing and fuselage, where many poor flowfield phenomena exist.

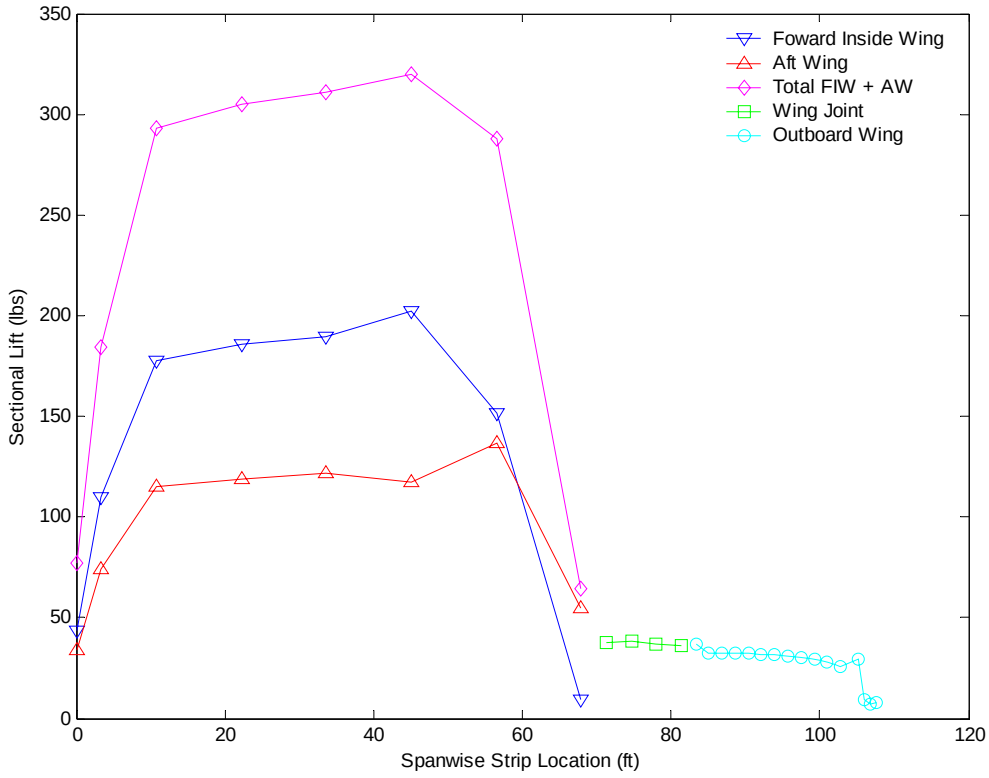

Figure 28. Roskam/AVTIE Strip Method Spanwise Lift Distribution

The AVTIE strip distribution of the joined-wing structure was divided into non-equal sectional areas. Since the forward inside and aft wings contained the largest spanwise distributions, as shown in Figure 19, the lift force was much greater at these sections than at the fine strips at the joint and outboard sections as shown in Figure 28. This makes comparing the lift distribution difficult. Therefore, all lift distribution results should be analyzed from Figure 27, where the lift coefficient is determined per unit span. This eliminates any inconsistencies at the joint section where the airfoil chord is doubled and tapered back to a normal chord of 2.5 meters.

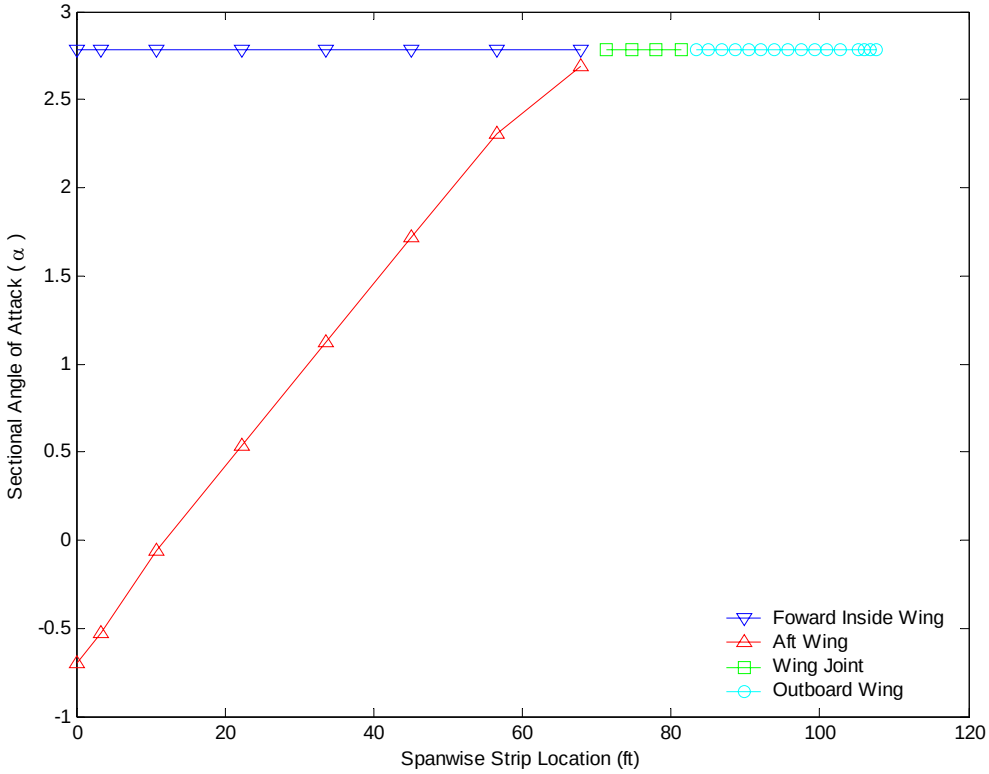

Figure 29. Roskam/AVTIE Strip Method Spanwise Freestream Angle-of-Attack

The freestream angle of attack (α) is defined as the AOA of each airfoil section with respect to freestream velocity, neglecting downwash effects. At the fourth point of the flight profile, AVTIE trimmed the aircraft at an AOA of 2.78 degrees (Figure 29). Since the current model incorporates no aerodynamic twist, every individual strip of the wing structure displays this freestream trimmed value except for the aft wing, which is being twisted down -3.53 degrees to -0.75 degrees freestream AOA in order to longitudinally trim the aircraft at the center of gravity. Within the MATLAB code, strip lift coefficient values are used to look up local AOA values for each panel as described in

the methodology section. Using C_L values from Figure 27, the local AOAs are generated in Figure 30.

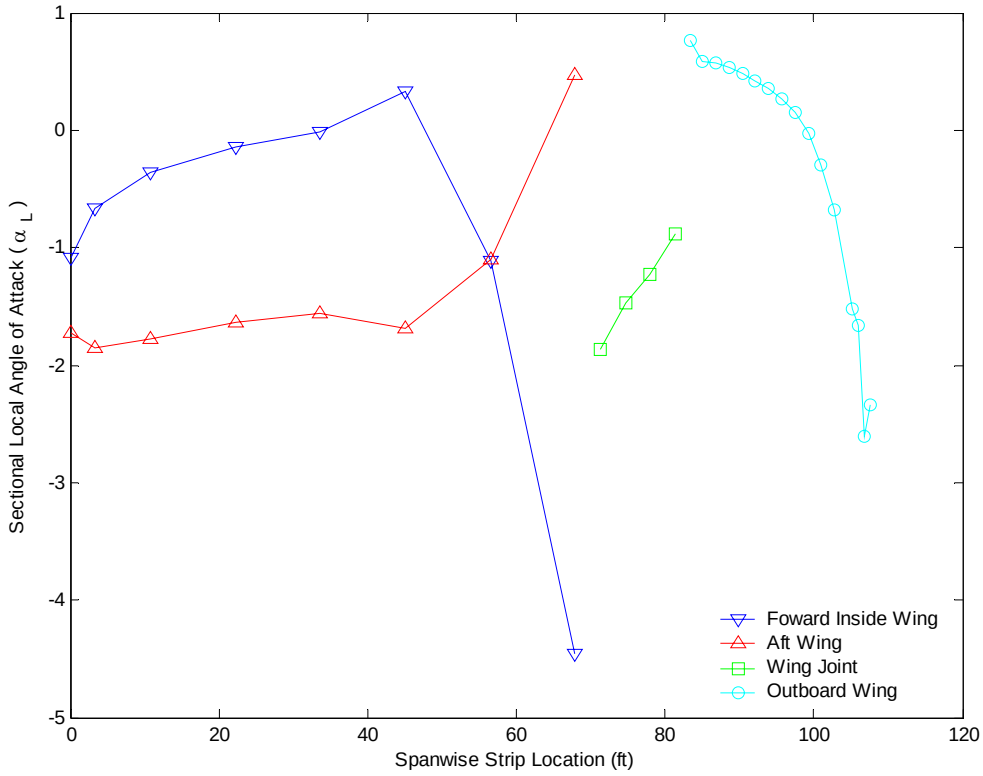

Figure 30. Roskam/AVTIE Strip Method Spanwise Local Angle-of-Attack

Local AOA is a three dimensional lift effect determined by two dimensional XFOIL data for the LRN-1015 airfoil. Measuring three dimensional effects with two dimensional data could lead to inconsistent results as shown in the local AOA distribution. Discontinuities are displayed at the merging of each of the wing sections with the joint, most likely due to a non-optimized wing joint model. The methodology of this process (Equation 14) requires the local AOA be less than the freestream AOA in order to create a positive induced AOA, resulting in positive induced drag. Each

spanwise section in Figure 30 has an α_L less than its corresponding α from Figure 29, resulting in all positive induced angles as shown in Figure 31.

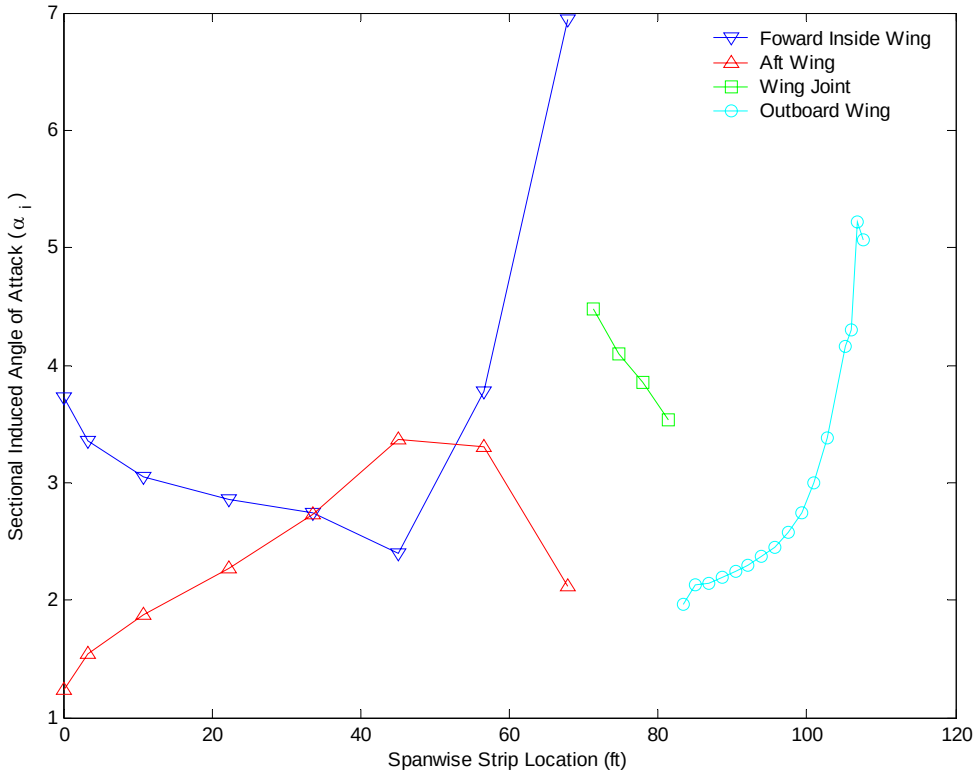

Figure 31. Roskam/AVTIE Strip Method Spanwise Induced Angle-of-Attack

The induced AOA distribution along the span of each of the wing sections shows expected characteristics. Most notable is the drastic increase of the induced AOA at the tip of the aircraft due to downwash effects. Again, similar to other results, the joint section creates complexity with the induced AOA. The merge between the forward inside wing and the joint section displays more induced AOA than at the tip, a result not expected in conventional configurations. This could be due to this methods inability to predict induced AOAs. Or it could be an accurate prediction due to very high

disturbances at this joint location as shown in Figure 18. To analyze further, Figure 32 is generated to display the spanwise induced drag distribution.

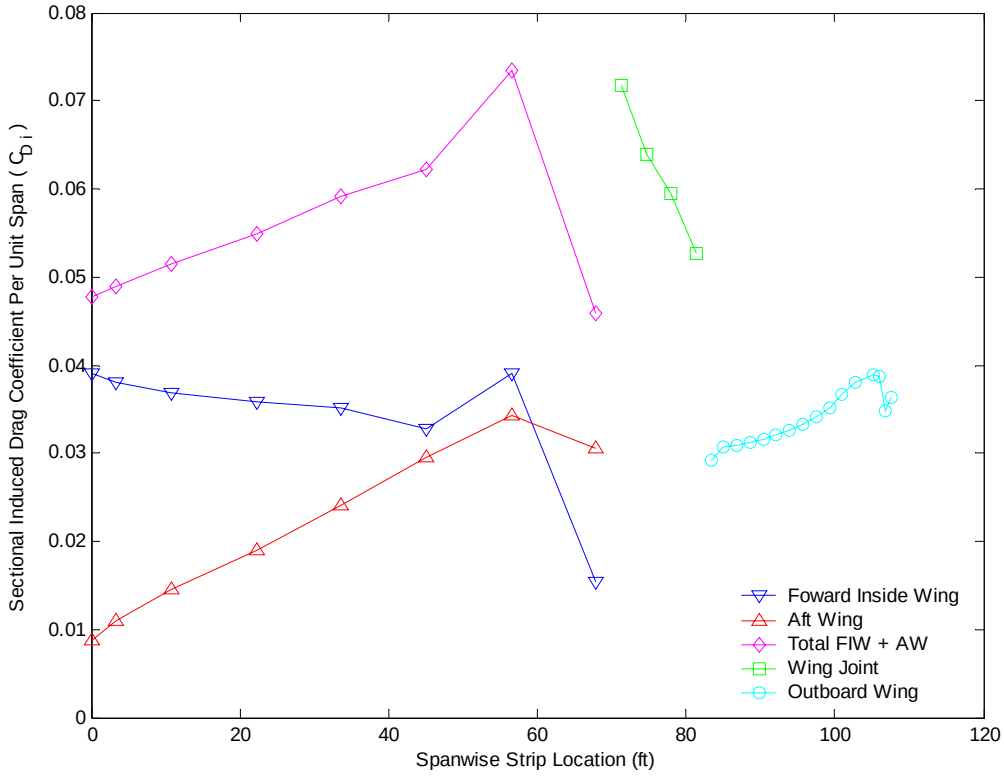

Figure 32. Roskam/AVTIE Strip Method Spanwise Induced Drag Distribution

Induced drag effects should increase approaching the tip of any lifting body. This method performed well in predicting this trend with the outboard wing. The aft wing also displayed increasing induced drag effects approaching the tip of the wing while the induced drag on the forward inside wing remained relatively constant. However, again many complications arise at the union between the forward inside and aft wings due to poor aerodynamic performance at these airfoil cross sections. Initial predictions assumed induced drag at the joint section would be relatively similar to the surrounding sections

and parasite drag would be much greater due to the complex airfoil shape throughout the wing joint section, as shown in Figure 33.

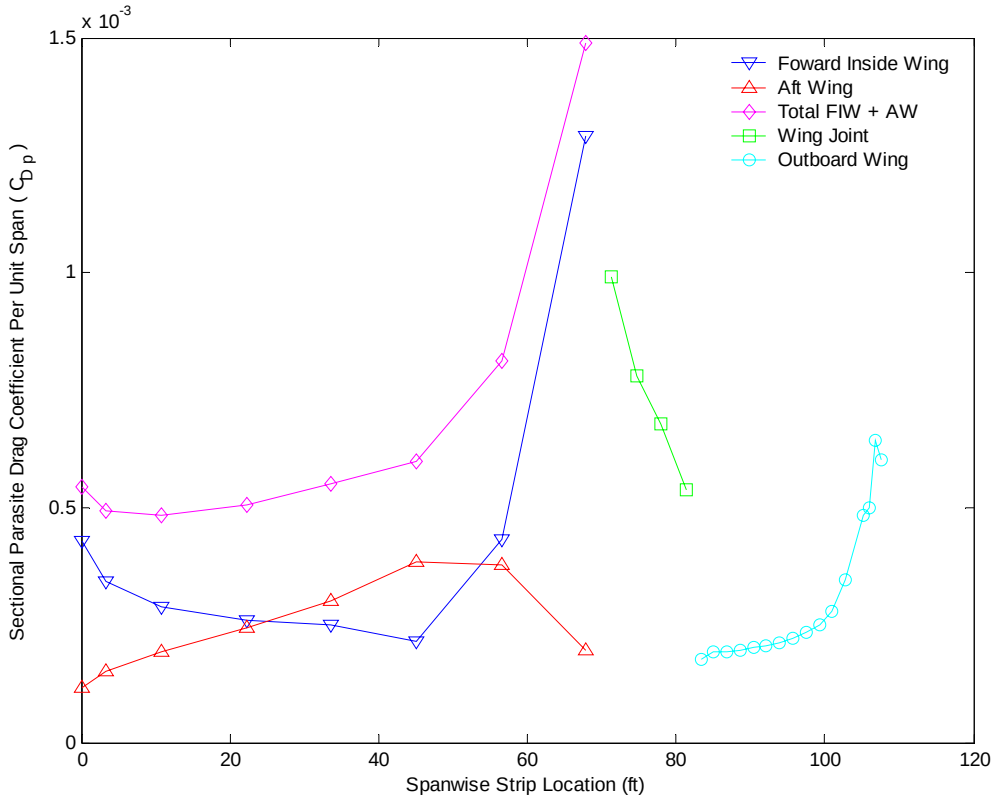

Figure 33. Roskam/AVTIE Strip Method Spanwise Parasite Drag Distribution

As expected, parasite drag increased at the wing joint due to complex airfoil geometry. This drag also decreases throughout the joint section as the airfoil geometry blended into its normal shape. The merging of the forward inside wing at the joint displays parasite drag effects nearly three times greater than all other sections. An increase of parasite drag near the tip is not expected as shown in Figure 32. All sections, with the exception of the merge of the forward inside wing and joint section, experience parasite drag effects within the range of 0.0050 and 0.0100, reasonable values assuming

two dimensional flow, low AOA trim conditions preventing boundary layer growth, and neglecting downwash effects.

Lift and drag predictions for each strip are totaled and factored according to each strip's corresponding wing area to determine induced and parasite drag for the entire wing configuration. The results for each mission point are shown in Table 14, in conjunction with the parasite drag transition to an equivalent parasite area for the wing.

Table 14. Roskam/AVTIE Strip Method Wing Drag Results

Mission Point	C_{Di}	C_{Dp}	f_{wing}
1	0.0292	0.0003198	1.0672
2	0.0315	0.0003243	1.0822
3	0.0373	0.0003330	1.1111
4	0.0306	0.0003184	1.0625
5	0.0231	0.0002989	0.9974
6	0.0170	0.0002789	0.9305
7	0.0096	0.0002384	0.7954

The wing results for the strip method are drastically different from those estimated by Roskam. Roskam predicted the equivalent parasite area for the wing to be around 35.0, while this method is estimating around one. Although, the induced drag predicted by this method is about three times greater than those estimated by the Roskam method. Just by observation, adding Roskam estimated fuselage and vertical tail drag to this method's equivalent parasite area will result in a much lower value for the total aircraft parasite drag than those produced by the Roskam method.

The Roskam/AVTIE strip method induced drag results from Table 14 are added to the configuration parasite drag to account for all drag effects on the model, assuming

the fuselage produces no lift and all induced drag acts on the wings alone. Using Pan Air generated lift coefficients, the calculated total drag on the aircraft from MATLAB, and the Breguet range equation, aerodynamic performance throughout the flight profile is determined and shown in Table 15.

Table 15. Roskam/AVTIE Strip Method Drag Results

Point	C_L	C_{Dp}	C_{Di}	C_{Dtotal}	L/D	Fuel (lbs)
1	0.5908	0.008070	0.0292	0.0373	15.8316	52,192.24
2	0.6161	0.008382	0.0315	0.0399	15.4545	40,903.36
3	0.6778	0.008578	0.0373	0.0458	14.7870	30,746.34
4	0.6034	0.008883	0.0306	0.0395	15.2898	7,660.70
5	0.5052	0.008997	0.0231	0.0320	15.7627	-6,700.30
6	0.4237	0.008514	0.0170	0.0255	16.6175	-8,468.45
7	0.3027	0.007977	0.0096	0.0176	17.1808	-12,991.02

Although aircraft parasite drag effects improved from the Roskam method, the induced drag dominated the lift-to-drag ratios, leading to the over-consumption of fuel. Fuel tank resizing was attempted, but the amount of fuel required overcame the aircraft's ability to trim at early flight profile points due to high required angles-of-attack pushing the airfoil beyond its stall condition.

The parasite drag values for the wing using this spanwise evaluation method are more accurate than those predicted by the Roskam method. Within the Roskam method, parasite drag for the wing was estimated utilizing Equation (6). This relationship is a general estimate for all airfoil geometries. The only parameters within this equation that define the geometry of the airfoil are L' and the maximum thickness-to-chord ratio. According to this relationship, any airfoil consisting of the same max t/c and its location

within the chord has the exact same parasite drag characteristics, which of course is obviously false. The Roskam/AVTIE strip method determined parasite drag for the wing based on XFOIL data specific to the LRN-1015 airfoil. Although an airfoil may have the same max t/c value and location, XFOIL will predict different parasite drag values for each unique airfoil section.

The induced drag values predicted using this method are large, and another method to model these effects was investigated. A third drag buildup method was developed and it combined parasite drag from the strip method with induced drag predictions from Pan Air.

4.4 Roskam/AVTIE Pan Air Method Results

With validated parasite drag estimates from the strip method, and reasonable induced drag results from Pan Air, the Roskam/AVTIE Pan Air method was accepted as the best approach to accurately model aircraft drag on the joined-wing model. Similar to the strip method, MATLAB code is used to interpolate two dimensional parasite drag effects from tabulated XFOIL data for the LRN-1015 airfoil by individual strips along the span of the wing to account for all parasite drag effects of the wing. However, unlike relying on XFOIL to predict induced drag results, this method used Pan Air archived induced drag for the entire aircraft configuration at each mission point. This value represents the induced drag of the entire vehicle and is not subdivided into panels as previously done within the Roskam/AVTIE strip method. The point-by-point results of the wing throughout the mission profile are shown in Table 16.

Table 16. Roskam/AVTIE Pan Air Method Wing Drag Results - Trial 1

Mission Point	C_{Di}	C_{Dp}	f_{wing}
1	0.0112	0.0004239	1.4145
2	0.0112	0.0004614	1.5396
3	0.0091	0.0005403	1.8029
4	0.0107	0.0006272	2.0929
5	0.0112	0.0008508	2.8389
6	0.0127	0.0007376	2.4613
7	0.0150	0.0007150	2.3858

As predicted, this method produced very favorable parasite and induced drag values for the wing structure. Applying these equivalent parasite areas to those of the vertical tail and fuselage from Roskam and incorporating the induced drag results from Pan Air in the flight profile results in Table 17.

Table 17. Roskam/AVTIE Pan Air Method Drag Results - Trail 1

Point	C_L	C_{Dp}	C_{Di}	C_{Dtotal}	L/D	Fuel (lbs)
1	0.4609	0.008156	0.0112	0.0194	23.7755	52,192.24
2	0.4884	0.008529	0.0112	0.0197	24.7298	44,363.13
3	0.6031	0.008821	0.0091	0.0179	33.6145	37,815.32
4	0.6294	0.009232	0.0107	0.0199	31.6550	26,152.76
5	0.5967	0.009706	0.0112	0.0209	28.5040	16,082.68
6	0.4837	0.009044	0.0127	0.0218	22.2212	14,437.09
7	0.3753	0.008489	0.0150	0.0234	16.0058	8,372.13

The acceptable wing parasite and induced drag results with this method resulted in lift-to-drag ratio up into the low 30's, as claimed by Wolkovich [11], during loitering flight, which are phenomenal, and nearly unrealistic for this type of configuration without

boundary layer control devices along the wing sections. Initial trimming of this configuration did not account for L/D ratios of this magnitude. Therefore, at every point of the flight profile the aircraft was trimmed at a lighter fuel load due to improved fuel consumption rates from high L/D ratios. To correctly model this flight profile, the AVTIE trim process was executed again, applying higher weight at each point of the flight profile to converge towards accurate trim condtions. This in turn increased the trim angle-of-attack, increasing drag, lowered L/D ratios, and resulted in increased fuel consumption.

Table 18. Roskam/AVTIE Pan Air Method Wing Drag Results - Trial 2

Mission Point	C_{Di}	C_{Dp}	f_{wing}
1	0.0127	0.0003000	1.0009
2	0.0129	0.0003105	1.0362
3	0.0111	0.0003256	1.0865
4	0.0108	0.0003252	1.0852
5	0.0169	0.0003733	1.2455
6	0.0194	0.0003643	1.2156
7	0.0173	0.0002940	0.9809

Table 19. Roskam/AVTIE Pan Air Method Drag Results - Trial 2

Point	C_L	C_{Dp}	C_{Di}	C_{Dtotal}	L/D	Fuel (lbs)
1	0.4768	0.008050	0.0127	0.0207	22.9906	52,192.24
2	0.5045	0.008368	0.0129	0.0212	23.7764	44,245.23
3	0.5471	0.008570	0.0111	0.0197	27.8252	35,031.32
4	0.5273	0.008890	0.0108	0.0197	26.8344	20,993.39
5	0.6174	0.009071	0.0169	0.0259	23.8172	9,593.56
6	0.6256	0.008600	0.0194	0.0280	22.3428	6,968.43
7	0.4169	0.008032	0.0173	0.0253	16.4637	969.23

The results for the correctly trimmed Pan Air method are shown in Table 18 and Table 19. The induced drag was roughly twice the parasite drag throughout the profile, a reasonable result. Lift-to-drag ratios consistently in the mid-20s throughout the flight profile result in favorable fuel consumption. The induced drag values from this method are similar to those from the Roskam method, which was determined using Equation (11), which was based on an elliptic lift distribution of conventional aircraft and was a function of aspect ratio. This relationship was initially not expected to accurately model induced drag values when implied to the radical joined-wing concept.

With an accurate method for predicting induced and parasite drag on the AFRL joined-wing model, wing twist was applied to the model in an attempt to optimize the aerodynamic performance of the wing.

4.5 Method Comparison Of Zero Lift Drag (C_{Do})

The zero lift drag, or overall aircraft parasite drag, is used to estimate the parasite, or viscous, drag on the entire vehicle. This parameter is measured at a trim condition at which the aircraft is producing zero lift. For aircraft incorporating cambered airfoils results in a required negative freestream angle-of-attack trim condition. For most aircraft configurations, the zero lift drag coefficient is actually higher than the minimum drag condition, usually located within the drag bucket.

For each point of the flight profile, MATLAB is used to estimate the zero lift drag according to Equation (18), supplemented by Equations (19) and (20). Since the zero lift drag coefficient of aircraft is usually represented as a single value, an average for each of

the seven zero lift drag coefficients is determined the overall C_{Do} of the aircraft. Table 20 shows the results for each of the three drag buildup methods used in this study.

Table 20. Zero Lift Drag Coefficients

Method	Abbreviation	C_{Do}
Roskam Method	R	0.0156
Roskam/AVTIE Strip Method	RAs	0.0237
Roskam/AVTIE Pan Air Method	RApa	0.0125

As expected, the Roskam/AVTIE strip method produced very high results, as a result of excessive induced drag predictions. The Roskam and Pan Air methods are very similar. The zero lift drag coefficient for the Boeing 747, an aircraft of similar wing span, is 0.0148 [31], but includes nacelles and a much larger fuselage. Neglecting these components results in a C_{Do} of approximately 0.0125, validating the Pan Air method.

4.6 Aerodynamic Twist

The purpose of aerodynamic twist is to tailor the lift distribution along the span as desired, which ultimately effects the C_{Di} distribution. Most configurations attempt to produce an elliptic lift distribution for optimal aerodynamic performance. The spanwise C_{Di} results were generated using the strip method, and was based solely on tabulated XFOIL data for the LRN-1015 airfoil. However, this method was found unable to accurately predict C_{Di}. The only reasonable C_{Di} results were found using the Pan Air method, which were determined for the whole wing structure and not by a spanwise strip

manner. Since this method did not archive spanwise induced drag by individual wing strips, tailoring the induced drag distribution of Figure 32 was not an option. However, Pan Air did output the spanwise lift coefficient distribution acting on each wing strip section. Therefore, an elliptic lift distribution was achieved by attempting to manually shape the lift curve shown in Figure 27. Manually tailoring the lift distribution towards an elliptic lift distribution will ultimately tailor the induced drag distribution.

The AVTIE environment allows for twist to be distributed at four locations along the joined-wing: the forward inside wing root at the fuselage (FIW_r), the forward inside wing tip at the joint section (FIW_t), the aft wing root at the fuselage (AW_r), and the outboard wing tip (FOW_t). The twist distribution between these stations followed a linear relationship. A trial-by-error process was determined as an acceptable approach to initial twist distribution studies. This process was executed and varied each station in one degree increments in order to establish a "twist trend" reduction of the induced drag from Pan Air. This method was applied to the fourth point of the flight profile at a fuel weight of 9,200 kg in an effort to optimize the aircraft for loitering performance.

The results for this process are shown in Table 21. Trials 2 and 3 indicate a reduction in induced drag if the forward inside wing root section is twisted up. Trials 4 and 5 indicate a reduction in induced drag if the forward inside wing tip is twisted down. Trials 6 and 7 show a decrease in induced drag when the outboard wing tip is also twisted down. These initial trends were combined in Trial 8. From this Trial spawned Trials 9 and 10, which continued enforcing the trends discovered in Trials 1 through 7. The aft wing was never twisted in this process, as the trimming process within AVTIE will twist this section to the required location required for zero longitudinal pitch.

Table 21. Trial-By-Error Twist Distribution

Trial Number	Twist Distribution (degrees)				Induced Drag (C_{Di})	Percentage Increase/Decrease
	FIW_r	FIW_t	AW_r	FOW_t		
1	0	0	0	0	0.00972	Base
2	1	0	0	0	0.00879	-9.57
3	-1	0	0	0	0.01093	12.45
4	0	1	0	0	0.01042	7.20
5	0	-1	0	0	0.00919	-5.45
6	0	0	0	1	0.01048	7.82
7	0	0	0	-1	0.00906	-6.79
8	1	-1	0	-1	0.00785	-19.24
9	2	-1	0	-2	0.00704	-27.57
10	3	-2	0	-3	0.00693	-28.70

Although induced drag continually decreased throughout this process, the parasite drag was never evaluated and must be considered in order to claim an optimized twist distribution. As the range of wing twist between the maximum and minimum stations increases, so does the freestream α to each wing section, resulting in boundary layer growth and increased parasite drag.

The lift coefficient distributions for Trials 1, 8, 9, and 10 are shown in Figure 34, Figure 35, Figure 36, and Figure 37 respectively. Elliptic distributions were determined by analyzing the total inside wing sections (the magenta line) and neglected the performance for the individual FIW and AW. In each of the lift distribution plots, initial intuition may suggest to rotate the FIW_t section up in order to "level out" the curve. However, the trial-by-error process concluded this section should be rotated down to reduce induced drag. The loss of lift at these sections is most likely due to the poor aerodynamic qualities of the wing joint section airfoil geometry, as shown in Figure 17.

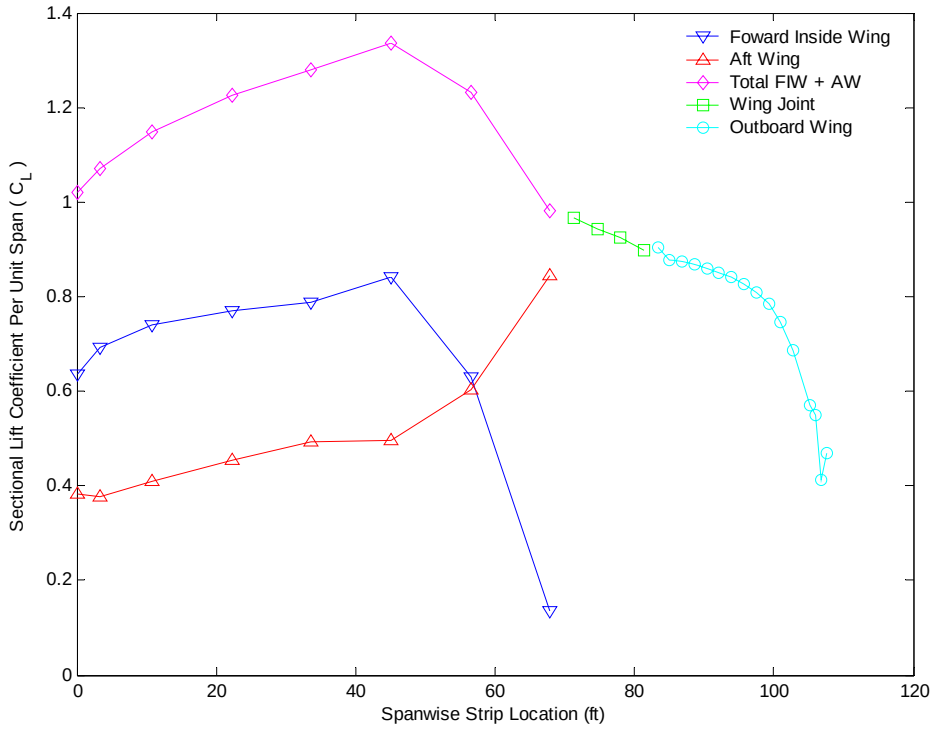

Figure 34. Trial 1 Twist Distribution (Zero Twist)

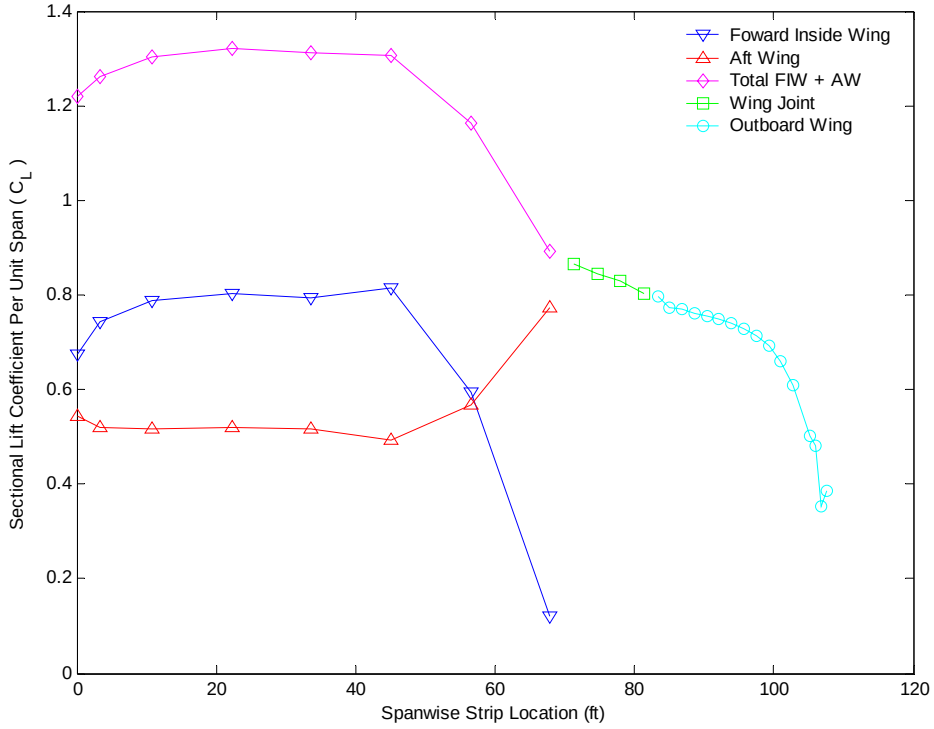

Figure 35. Trial 8 Twist Distribution

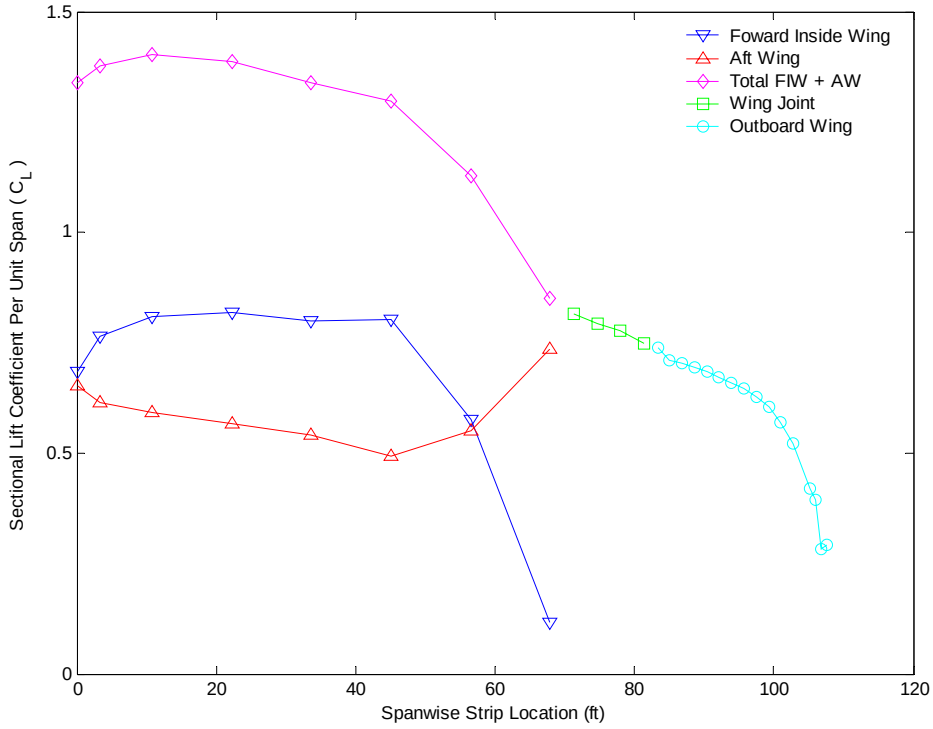

Figure 36. Trial 9 Twist Distribution

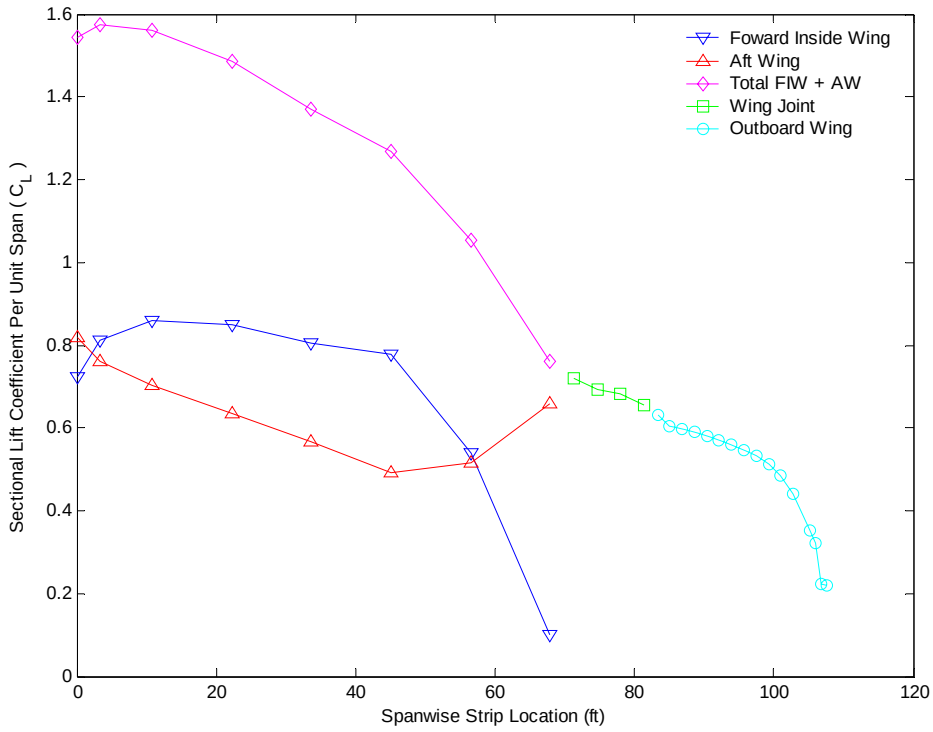

Figure 37. Trial 10 Twist Distribution

Each modified twist distribution displayed a more definite elliptic lift distribution with respect to the base configuration in Figure 34. Although each of these distributions are acceptable for the joined-wing, a full drag assessment for the entire aircraft is conducted in order to evaluate parasite drag forces and determine the optimal twist distribution. A full drag assessment of the whole aircraft configuration for each of the Trials is shown in Table 22.

Table 22. Twist Optimization Results

Trial Number	C_L	C_{Dp}	C_{Di}	C_{Dtotal}	L/D
1	0.6305	0.0150	0.0097	0.0247	25.4878
8	0.6328	0.0151	0.0078	0.0229	27.6139
9	0.6337	0.0151	0.0070	0.0222	28.6065
10	0.6366	0.0154	0.0069	0.0223	28.5290

Although induced drag decreased throughout each Trial, parasite drag effects began to overwhelm the benefits of lower C_{Di} values due to a wider range of sectional angle-of-attacks, increasing boundary layers. Based strictly on these results, the optimal twist distribution for the AFRL joined-wing sensor craft is 2° at the forward inside wing root, -1° at the forward inside wing tip at the joint, 0° of twist at the aft wing root, and -2° of twist at the outboard wing tip.

4.7 Induced Drag Relationship

The Roskam drag buildup method had the distinct advantage of producing reasonable results in a quick amount of time with respect to the Roskam/AVTIE Strip and

Pan Air methods. It was shown that although the joined-wing is radical design to the aviation industry, it does not deviate far enough for the spectrum of conventional aircraft Roskam analyzed when constructing the charts within his text. All parasite drag of the aircraft was estimated by Roskam, but the induced drag was determined by Equation (11). This relationship was based on a single lifting surface incorporating an elliptic lift distribution. A revised relationship between the lift and induced drag is formulated in Equation (21).

$$C_{Di} = \frac{2C_L^2}{\pi AR} \tag{21}$$

This relationship should be applied to the joined-wing concept in order to provide quick induced drag estimates. The relationship is applied to the whole wing, including both sides of the aircraft, and is not valid for individual wing strips. The lift coefficient is based on the lift of the entire vehicle, and the aspect ratio is calculated by Equation (22).

$$AR = \frac{b^2}{S} = \frac{\left\{2\left(b_{\text{FIW}} + b_{\text{Joint}} + b_{\text{FOW}}\right)\right\}^2}{S_{\text{entire wing}}} \tag{22}$$

In essence, Equation (22) states the aspect ratio applied to Equation (21) is equivalent to twice the span of the whole fore wing (to account for both sides of the aircraft) squared, divided by the wing planform area of the entire wing, including the aft wing. Table 23 compares the induced drag estimates from Equation (11), used with the

Roskam method, with those computed using the modified joined-wing induced drag relationship of Equation (21).

Table 23. Induced Drag Relationship Application

Point	C_L	C_{Di} (Pan Air)	C_{Di} (Eq. 11)	% Difference	C_{Di} (Eq. 21)	% Difference
1	0.4768	0.01270	0.00555	56.33	0.00970	23.60
2	0.5045	0.01290	0.00621	51.87	0.01086	15.79
3	0.5471	0.01110	0.00730	34.21	0.01277	15.09
4	0.5273	0.01080	0.00678	37.19	0.01187	9.88
5	0.6174	0.01690	0.00930	44.97	0.01627	3.73
6	0.6256	0.01940	0.00955	50.78	0.01670	13.90
7	0.4169	0.01730	0.00424	75.49	0.00742	57.12

The two induced drag relationships in Table 23 are compared to the induced drag predictions from Pan Air, essentially assuming Pan Air predicts correct induced drag. The original induced drag relationship in Equation (11) is incorrect by an average 51% throughout the flight profile. The modified relationship in Equation (21) is incorrect by an average of 19% throughout the flight profile. Most of this error is accounted for on the last point of the flight profile at the conclusion of the mission, which is not applied to any future mission segments, and can be neglected. If the induced drag estimate from mission point seven is neglected, this modified relationship yields a difference of 13% from the Pan Air predictions, much improved from the 51% difference obtained using Equation (11).

V. Conclusions and Recommendations

5.1 The Roskam Method

The Roskam method produced reasonable results based solely on data from other aircraft configurations. This method was the least labor intensive method to code and execute for varying configurations and flight profiles. This method required an increase of 5,500 kg of fuel (12,125 lbs) in order to accomplish its intended mission due to lift-to-drag ratios in the range of 20-23 instead of the previously assumed 24. Due to the potential fuel volume accommodated by the aircraft's extremely large wingspan, an additional 12,125 lbs of fuel is not a limitation.

The code generated in MATLAB was developed as independently as possible allowing many aircraft characteristics to be varied in order to analyze flight profile effects, for both conventional and un-conventional designs such as the joined-wing. With the Roskam method's proven ability to accurately model aircraft drag characteristics for conventional designs and the joined-wing configuration, the MATLAB code developed could also be applied to other radical designs to generate rough estimates of aircraft performance. But, other drag estimation methods are recommended to validate those from Roskam, as done within this research. However, unlike the AVTIE environment, the Roskam code in MATLAB cannot model any aeroelastic effects such as wing deflections. For high aspect ratio wing aircraft, aeroelastic effects will drastically affect the aerodynamic performance of the aircraft, and caution should be exercised when drag data is determined with the Roskam method.

5.2 The Roskam/AVTIE Strip Method

Contradictory to the Roskam method, this method estimated extremely large induced drag effects. This method estimated induced drag, a three dimensional effect, from two dimensional XFOIL data. Cleary, applying two dimensional results to a real world three dimensional lift distribution with downwash is not an acceptable approach to model aircraft drag. However, sectional parasite drag produced within this method was much more reasonable predicted by XFOIL. This method is recommended as an appropriate approach to model parasite drag of lifting bodies.

5.3 The Roskam/AVTIE Pan Air Method

This method combined the accurate inviscid (induced) drag predictions from Pan Air and the precise parasite drag estimates from the strip method extrapolated from XFOIL. Overall results showed L/D ratios in the upper to mid 20s throughout the flight regime, improved performance to the earlier assumption of 24. The correctly trimmed flight profile concluded with nearly 1,000 lbs of fuel in reserve. This could extend the range or loitering time of the aircraft. Or the tanks could be decreased in size, resulting in shallower trim angles with less induced and parasite drag, prolonging flight time. However, for an aircraft intended to fly for extended periods of time (greater than 24 hours), fuel to spare is a necessity and fuel tank resizing is not recommended.

This method relied on two esteemed aerodynamic tools, XFOIL and Pan Air, and extrapolated from these environments values they were intended to accurately predict, parasite drag and induced drag respectively. Overall, this method is determined to be the

best approach to model aircraft drag. This method can be easily applied to any geometric changes to the joined-wing baseline configuration.

5.4 AVTIE Recommendations

The AVTIE environment was found to be extremely labor intensive and time consuming to calculate results. The Roskam/AVTIE Pan Air method in MATLAB code could easily be translated to other software code in order to aid AVTIE in aerodynamic performance calculations. The lift coefficient and induced drag values from AVTIE, both produced by Pan Air, are the only dependant variables required to generate solutions with the MATLAB code. AVTIE is still a very powerful tool that is required for wing mesh generation, trim calculations, and structural analysis.

A recommendation is to re-write AVTIE into a software language more widely used. Adaptive Modeling Language (AML) is not intuitive and has a steep learning curve associated with it. Also, the de-bugging and error message generated available in AML is unhelpful. Thus, correcting errors in the object-oriented code becomes difficult in a very large program such as AVTIE. Combining the AVTIE environment with the Roskam/AVTIE Pan Air MATLAB code into a single design package will prove to be a powerful aid in future AFRL joined-wing design studies.

5.5 AFRL Model Recommendations and Future Studies

The process outlined in the Roskam/AVTIE Pan Air drag method proved to be an accurate assessment of both parasite and induced drag. Using this approach for drag

evaluation, a more thorough aerodynamic twist study should be conducted in order to converge on an optimal twist distribution. This twist should be applied to wing deflections throughout the flight regime of the aircraft. The Roskam/AVTIE Pan Air method is capable to model induced and parasite drag effects with any twist distribution and/or magnitude of wing deflection.

The complex geometry of the baseline wing joint section resulted in many unwanted aerodynamic disturbances, decreasing lift and increasing drag. A thorough study should be conducted in order to structurally and aerodynamically optimize the merging of the forward and aft wings. Once an optimal joint section is modeled, XFOIL should be applied to each modified airfoil section in order to more accurately predict the parasite drag about each of the joint section strips. Again, the Roskam/AVTIE Pan Air drag method presented in this study can be used to determine the aerodynamic gains from future joint section designs.

An experimental study should be conducted for the AFRL configuration in order to validate the drag results presented in this study. A CFD analysis for the AFRL baseline configuration will allow a comparison between both the parasite and induced drag results from the Roskam/AVTIE Pan Air method. Both a viscous and inviscid flowfield solution will provide enough information to validate the estimates presented with the Pan Air drag buildup method.

Appendix A. MATLAB Drag Evaluation Code

This Appendix displays all the MATLAB code developed to aid AVTIE in estimating drag characteristics. This code was pasted directly from MATLAB m-files, and can be copied back to an executable MATLAB form.

A.1 The Performance Code

```
%==========================================================================

%                    PERFORMANCE CODE

%==========================================================================

%      This code is the main component of the MATLAB calculations.  It
%   calls on many other supplimental MATLAB codes in order to generate
%   aerodynamic results for a given flight profile, twist distribution,
%   etc.

%==========================================================================

close all
clc
clear

%==========================================================================

% GENERAL DATA INPUT

   % General Constants
   R = 1716;                              % Gas Constant (ft-lbs/slug-R)
   gama = 1.4;                            % Air Constant
   g = 32.174;                            % Acceleration Due To Gravity (ft/s^2)

   % Conversion Factors
   cf_l = 3.280839895;                        % Converts (m) to (ft)
   cf_m = 2.204622622;                        % Converts (kg) to (lbs)
   cf_d = 0.621371192;                        % Converts (km) to (miles)
   cf_a = pi/180;                         % Converts (deg) to (rad)

   % Choice Of AVTIE Output Run       % 1 - 0 Twist
   run = 8;                      % 2 - +2,+2,0,-2 Degrees Of Twist (Old Code)
      %(FIB, AIB, JT, OB)          % 3 - +2,+2,0,-2 Degrees Of Twist (New Code)
                              % 4 - +16,-8,+14,+11 Degrees Of Twist
                              % 5 - +8,0,+6,+3 Degrees Of Twist
                              % 6 - +6,+2,+4,+1 Degrees Of Twist
                              % 7 - 0 Twist (Optimized RApa Method - 1st Try)
                              % 8 - 0 Twist (2nd Try - New Flt Prof)
```

```
   % Choice Of AVTIE Output Method     % 1 - Data Taken From Begining Legs Of Profile (0%)
   meth = 1;                          % 2 - Data Taken From Ending Legs Of Profile (99%)
                                      % 3 - Average Between 1 and 2

   % Choice Of Flight Profile
   flt_prof = 2;                      % 1 - Old
                                      % 2 - New

   % Recall Specific Flight Profile (Normal,Alt1,Alt2,Alt3,Alt4,Alt5)
   Joined_Wing_Flight_Profile_7PT
   npts = length(t);

%=========================================================================

% ESTIMATED AIRCRAFT DIMENSIONS/CHARACTERISTICS

   % Weight
   W_e = 10810*cf_m;                  % Empty Weight (lbs)
   W_p = 3550*cf_m;                   % Payload (lbs)
   W_f_base = 24674*cf_m;             % Base Amount Of Max Fuel Load (lbs)
   W_f_clm = 1000*cf_m;               % Assumed Fuel Consumed During Climb (lbs)
   W_ac = W_e+W_p;                    % Aircraft Weight Including Payload (lbs)

   % Fuselage
   L_fuselage = 30*cf_l;              % Fuselage Length (ft)
   D_fuselage = 6*cf_l;               % Fuselage Diameter (ft)
   LD_fuse = L_fuselage/D_fuselage;   % Fuselage Length to Diameter Ratio

   % Foward Inside Wing (FIW) -  Accounts For Just One Side of Aircraft
   fiw_b = 26*cf_l;                   % Foward Inside Wing Span (ft)
   fiw_c_r = 2.5*cf_l;                % Foward Inside Wing Root Chord (ft)
   fiw_c_t = 2.5*cf_l;                % Foward Inside Wing Tip Chord (ft)
   fiw_TR = fiw_c_t/fiw_c_r;          % Foward Inside Wing Taper Ratio
   fiw_GMC = 0.5*(fiw_c_t+fiw_c_r);   % Foward Inside Wing Geometric Mean Chord (ft)
   fiw_MAC = ((2*fiw_c_r)/3)*((1+fiw_TR+fiw_TR^2)/(1+fiw_TR)); % Foward Inside Wing Mean Aerodynamic
Chord (ft)
   fiw_S = fiw_GMC*fiw_b;             % Foward Inside Wing Wing Area (ft^2)
   fiw_AR = fiw_b^2/fiw_S;            % Foward Inside Wing Aspect Ratio
   fiw_swp = 30*cf_a;                 % Foward Inside Wing Sweep at T/Cmax (rad)
   fiw_t_c_max = 0.1519;              % Foward Inside Wing Max Thickness to Chord
   fiw_t_c_max_l = 0.40;              % Location of T/Cmax of Foward Inside Wing (% Chord)

   % Foward Outside Wing (FOW)  -  Accounts For Just One Side of Aircraft
   fow_b = 8*cf_l;                    % Foward Outside Wing Span (ft)
   fow_c_r = 2.5*cf_l;                % Foward Outside Wing Root Chord (ft)
   fow_c_t = 2.5*cf_l;                % Foward Outside Wing Tip Chord (ft)
   fow_TR = fow_c_t/fow_c_r;          % Foward Outside Wing Taper Ratio
   fow_GMC = 0.5*(fow_c_t+fow_c_r);   % Foward Outside Wing Geometric Mean Chord (ft)
   fow_MAC = ((2*fow_c_r)/3)*((1+fow_TR+fow_TR^2)/(1+fow_TR)); % Foward Outside Wing Mean Aerodynamic
Chord (ft)
   fow_S = fow_GMC*fow_b;             % Foward Outside Wing Wing Area (ft^2)
   fow_AR = fow_b^2/fow_S;            % Foward Outside Wing Aspect Ratio
   fow_swp = 30*cf_a;                 % Foward Outside Wing Sweep at T/Cmax (rad)
   fow_t_c_max = 0.1519;              % Foward Outside Wing Max Thickness to Chord
   fow_t_c_max_l = 0.40;              % Location of T/Cmax of Foward Outside Wing (% Chord)

   % Aft Wing (AW)  -  Accounts For Just One Side of Aircraft
   aw_b = 26*cf_l;                    % Aft Wing Span (ft)
   aw_c_r = 2.5*cf_l;                 % Aft Wing Root Chord (ft)
   aw_c_t = 2.5*cf_l;                 % Aft Wing Tip Chord (ft)
   aw_TR = aw_c_t/aw_c_r;             % Aft Wing Taper Ratio
   aw_GMC = 0.5*(aw_c_t+aw_c_r);      % Aft Wing Geometric Mean Chord (ft)
```

```
aw_MAC = ((2*aw_c_r)/3)*((1+aw_TR+aw_TR^2)/(1+aw_TR));      % Aft Wing Mean Aerodynamic Chord (ft)
aw_S = aw_GMC*aw_b;                                          % Aft Wing Wing Area (ft^2)
aw_AR = aw_b^2/aw_S;                                         % Aft Wing Aspect Ratio
aw_swp = 15*cf_a;                                            % Aft Wing Sweep at T/Cmax (rad)
aw_t_c_max = 0.1519;                                         % Aft Wing Max Thickness to Chord
aw_t_c_max_l = 0.40;                                         % Location of T/Cmax of Aft Wing (% Chord)

% Total Wing
b = 2*(fiw_b+fow_b);                                         % Wing Span (ft)
S_m = 310;                                                   % Wing Planform Area From AVTIE (m^2)
S = S_m*(cf_l)^2;                                            % Wing Planform Area (ft^2)
AR = b^2/S;                                                  % Aspect Ratio

% Vertical Tail
t_h = 10*cf_l;                                               % Tail Hieght (ft)
t_c_r = 10*cf_l;                                             % Tail Root Chord (ft)
t_c_t = 5*cf_l;                                              % Tail Tip Chord (ft)
t_TR = t_c_t/t_c_r;                                          % Tail Taper Ratio
t_GMC = 0.5*(t_c_t+t_c_r);                                   % Tail Geometric Mean Chord (ft)
t_MAC = ((2*t_c_r)/3)*((1+t_TR+t_TR^2)/(1+t_TR));            % Tail Mean Aerodynamic Chord (ft)
t_S = t_GMC*t_h;                                             % Tail Area (ft^2)
t_AR = t_h^2/t_S;                                            % Tail Aspect Ratio
t_swp = 55*cf_a;                                             % Tail Sweep at T/Cmax (rad)
t_t_c_max = 0.15;                                            % Tail Max Thickness to Chord
t_t_c_max_l = 0.25;                                          % Location of T/Cmax of Tail (% Chord)

% Propulsion
TSFC_1 = 0.450;                                              % Estimated TSFC For Climb And Cruise
TSFC_2 = 0.450;                                              % Estimated TSFC For Descent

% Propeller
np = 0.80;                                                   % Propeller Efficiency

%=========================================================================

% RETRIEVING ATMOSPHERIC CHARACTERISTICS

for i=1:npts;
   [tinf(i), pinf(i), rinf(i), muinf(i)]=Joined_Wing_Atmosphere(h(i));
end

tinf = tinf';
pinf = pinf';
rinf = rinf';
muinf = muinf';

%=========================================================================

% BUILDING UP THE FLIGHT PROFILE

V = M.*sqrt(gama*R.*tinf);
qinf = 0.5*rinf.*V.^2;
x(1) = 0;

for i = 1:npts-1
   dt(i) = (t(i+1)-t(i))*60;
   dh(i) = h(i+1)-h(i);
   aout(i) = (V(i+1)-V(i))/dt(i);
   Vout(i) = (V(i+1)+V(i))/2;
   Mout(i) = (M(i+1)+M(i))/2;
   tout(i) = (t(i+1)+t(i))/2;
   ds(i) = V(i)*dt(i) + 0.5*aout(i)*dt(i)^2;
```

```
    dx(i) = sqrt(ds(i)^2 - dh(i)^2);
    dsout(i) = (V(i)*dt(i))/2+(aout(i)*dt(i).^2)/8;
    x(i+1) = x(i) + dx(i);
    xout(i) = x(i)+(dsout(i)/ds(i))*dx(i);
    hout(i) = h(i)+(dsout(i)/ds(i))*dh(i);
    thetaout(i) = asin(dh(i)/ds(i));
    RCout(i) = dh(i)/dt(i);
    q = (1/2).*rinf.*V.^2;
end

% Converted Into Columns
x = x'/5280;
xout = xout'/5280;
aout = aout';
hout = hout';
Mout = Mout';
tout = tout';
Vout = Vout';
thetaout = (thetaout')*180/pi;
RCout = RCout'*60;
t_hrs = t/60;

% Displaying Values In Command Window
% On The Interval
disp(' ')
disp(' ')
disp('Atmospheric Coniditions Throughout Mission:')
disp('  t(min)    h(ft)      T(R)     p(psf)    rho(slug/ft^3)     Viscosity')
disp('-------------------------------------------------------------------------')
fprintf('%8.1f  %8.1f   %8.3f   %8.3f     %8.6f     %8.10f\n', [t,h,tinf,pinf,rinf,muinf]')

% On The Half Interval
disp(' ')
disp(' ')
disp('Mission Profile Details:')
disp('  t(min)     h(ft)    Range(miles)   Mach      V(fps)   a(ft/s^2)  theta(deg)  R/C(fpm)')
disp('-----------------------------------------------------------------------------------')
fprintf('%8.1f               %8.1f               %8.2f                  %8.4f              %8.3f              %8.5f              %8.4f
%8.2f\n',[tout,hout,xout,Mout,Vout,aout,thetaout,RCout]')

%=========================================================================

% ROSKAM DRAG BUILDUP

   % Recall Roskam Drag Buildup Data
   Roskam_Drag_Buildup

   % Reference Values
   for j = 1:npts
      nu(j) = muinf(j)/rinf(j);
      RNL(j) = V(j)/nu(j);
   end
   nu = nu';
   RNL = RNL';

   % Wing/Fuselage Interference Factor
   RN_fuse = RNL*L_fuselage;
   for wf = 1:npts;
      wfMach(wf) = fix(20*M(wf)-4);
      wfRN(wf) = fix(4.339602918*log(RN_fuse(wf))-63.2680625528);
      if M(wf) > 0.25;
         wfMach_l(wf) = (wfMach(wf)+4)/20;
```

```
        wfMach_h(wf) = (wfMach(wf)+5)/20;
        wfRN_l(wf) = exp((wfRN(wf)+63.2680625528)/4.339602918);
        wfRN_h(wf) = exp((wfRN(wf)+64.2680625528)/4.339602918);
        R_wf_l(wf) = ((M(wf)-wfMach_l(wf))/(wfMach_h(wf)-wfMach_l(wf)))*(R_wf(wfRN(wf),wfMach(wf)+1)-
R_wf(wfRN(wf),wfMach(wf)))+R_wf(wfRN(wf),wfMach(wf));
        R_wf_h(wf) = ((M(wf)-wfMach_l(wf))/(wfMach_h(wf)-wfMach_l(wf)))*(R_wf(wfRN(wf)+1,wfMach(wf)+1)-
R_wf(wfRN(wf)+1,wfMach(wf)))+R_wf(wfRN(wf)+1,wfMach(wf));
        R_wf_s(wf) = ((RN_fuse(wf)-wfRN_l(wf))/(wfRN_h(wf)-wfRN_l(wf)))*(R_wf_h(wf)-
R_wf_l(wf))+R_wf_l(wf);
      else
        R_wf_s(wf) = 1.10;
      end
   end
   R_wf_s = R_wf_s';



   % FOWARD INSIDE WING (FIW)

   % FIW Lifting Surface Correction Factor (R_LS)
   for j = 1:npts;
      jMach(j) = fix(20*M(j)-4);
      s_swp = cos(fiw_swp);
      if M(j) > 0.25;
        R_LS_l(j) =
R_LS(jMach(j),1)*s_swp^6+R_LS(jMach(j),2)*s_swp^5+R_LS(jMach(j),3)*s_swp^4+R_LS(jMach(j),4)*s_swp^3+R
_LS(jMach(j),5)*s_swp^2+R_LS(jMach(j),6)*s_swp+R_LS(jMach(j),7);
        R_LS_h(j) =
R_LS(jMach(j)+1,1)*s_swp^6+R_LS(jMach(j)+1,2)*s_swp^5+R_LS(jMach(j)+1,3)*s_swp^4+R_LS(jMach(j)+1,4)*s
_swp^3+R_LS(jMach(j)+1,5)*s_swp^2+R_LS(jMach(j)+1,6)*s_swp+R_LS(jMach(j)+1,7);
        Mach_lw(j) = (jMach(j)+4)/20;
        Mach_hi(j) = (jMach(j)+5)/20;
        R_LS_s(j) = ((M(j)-Mach_lw(j))/(Mach_hi(j)-Mach_lw(j)))*(R_LS_h(j)-R_LS_l(j))+R_LS_l(j);
      else
        R_LS_s(j) =
R_LS(1,1)*s_swp^6+R_LS(1,2)*s_swp^5+R_LS(1,3)*s_swp^4+R_LS(1,4)*s_swp^3+R_LS(1,5)*s_swp^2+R_LS(1,6
)*s_swp+R_LS(1,7);
      end
   end
   R_LS_fiw = R_LS_s';

   % FIW Turbulent Mean Skin-Friction Coefficient (C_f)
   RN_s = RNL*fiw_MAC;
   for k = 1:npts;
      kMach(k) = fix(2*M(k)+1);
      kRN(k) = fix(1.290086*log(RN_s(k))-16.958276);
      kMach_l(k) = (kMach(k)-1)/2;
      kMach_h(k) = (kMach(k))/2;
      kRN_l(k) = exp((kRN(k)+16.958276)/1.290086);
      kRN_h(k) = exp((kRN(k)+17.958276)/1.290086);
      C_f_l(k) = ((M(k)-kMach_l(k))/(kMach_h(k)-kMach_l(k)))*(C_f(kMach(k),kRN(k))-
C_f(kMach(k)+1,kRN(k)))+C_f(kMach(k),kRN(k));
      C_f_h(k) = ((M(k)-kMach_l(k))/(kMach_h(k)-kMach_l(k)))*(C_f(kMach(k),kRN(k)+1)-
C_f(kMach(k)+1,kRN(k)+1))+C_f(kMach(k),kRN(k)+1);
      C_f_s(k) = ((RN_s(k)-kRN_l(k))/(kRN_h(k)-kRN_l(k)))*(C_f_h(k)-C_f_l(k))+C_f_l(k);
   end
   C_f_fiw = C_f_s';

   % FIW Airfoil Thickness Location Parameter (L')
   t_c = fiw_t_c_max;
   L_p_l = 1+1.2*t_c+100*t_c^4;
   L_p_h = 1+2*t_c+100*t_c^4;
```

```
K_fiw = (L_p_l+L_p_h)/2;

% FIW Wetted Area Calculation
fiw_S_wet = (2*(2*fiw_S))*(1+0.25*(fiw_t_c_max));

% FIW Equivalent Parasite Area
f_fiw = (R_wf_s).*(R_LS_fiw).*(C_f_fiw)*(K_fiw)*fiw_S_wet;



% FOWARD OUTSIDE WING (FOW)

% FOW Lifting Surface Correction Factor (R_LS)
for j = 1:npts;
   jMach(j) = fix(20*M(j)-4);
   s_swp = cos(fow_swp);
   if M(j) > 0.25;
      R_LS_l(j) = R_LS(jMach(j),1)*s_swp^6+R_LS(jMach(j),2)*s_swp^5+R_LS(jMach(j),3)*s_swp^4+R_LS(jMach(j),4)*s_swp^3+R_LS(jMach(j),5)*s_swp^2+R_LS(jMach(j),6)*s_swp+R_LS(jMach(j),7);
      R_LS_h(j) = R_LS(jMach(j)+1,1)*s_swp^6+R_LS(jMach(j)+1,2)*s_swp^5+R_LS(jMach(j)+1,3)*s_swp^4+R_LS(jMach(j)+1,4)*s_swp^3+R_LS(jMach(j)+1,5)*s_swp^2+R_LS(jMach(j)+1,6)*s_swp+R_LS(jMach(j)+1,7);
      Mach_lw(j) = (jMach(j)+4)/20;
      Mach_hi(j) = (jMach(j)+5)/20;
      R_LS_s(j) = ((M(j)-Mach_lw(j))/(Mach_hi(j)-Mach_lw(j)))*(R_LS_h(j)-R_LS_l(j))+R_LS_l(j);
   else
      R_LS_s(j) = R_LS(1,1)*s_swp^6+R_LS(1,2)*s_swp^5+R_LS(1,3)*s_swp^4+R_LS(1,4)*s_swp^3+R_LS(1,5)*s_swp^2+R_LS(1,6)*s_swp+R_LS(1,7);
   end
end
R_LS_fow = R_LS_s';

% FOW Turbulent Mean Skin-Friction Coefficient (C_f)
RN_s = RNL*fow_MAC;
for k = 1:npts;
   kMach(k) = fix(2*M(k)+1);
   kRN(k) = fix(1.290086*log(RN_s(k))-16.958276);
   kMach_l(k) = (kMach(k)-1)/2;
   kMach_h(k) = (kMach(k))/2;
   kRN_l(k) = exp((kRN(k)+16.958276)/1.290086);
   kRN_h(k) = exp((kRN(k)+17.958276)/1.290086);
   C_f_l(k) = ((M(k)-kMach_l(k))/(kMach_h(k)-kMach_l(k)))*(C_f(kMach(k),kRN(k))-C_f(kMach(k)+1,kRN(k)))+C_f(kMach(k),kRN(k));
   C_f_h(k) = ((M(k)-kMach_l(k))/(kMach_h(k)-kMach_l(k)))*(C_f(kMach(k),kRN(k)+1)-C_f(kMach(k)+1,kRN(k)+1))+C_f(kMach(k),kRN(k)+1);
   C_f_s(k) = ((RN_s(k)-kRN_l(k))/(kRN_h(k)-kRN_l(k)))*(C_f_h(k)-C_f_l(k))+C_f_l(k);
end
C_f_fow = C_f_s';

% FOW Airfoil Thickness Location Parameter (L')
t_c = fow_t_c_max;
L_p_l = 1+1.2*t_c+100*t_c^4;
L_p_h = 1+2*t_c+100*t_c^4;
K_fow = (L_p_l+L_p_h)/2;

% FOW Wetted Area Calculation
fow_S_wet = (2*(2*fow_S))*(1+0.25*(fow_t_c_max));

% FOW Equivalent Parasite Area
f_fow = (R_wf_s).*(R_LS_fow).*(C_f_fow)*(K_fow)*fow_S_wet;
```

```
% AFT WING (AW)

% AW Lifting Surface Correction Factor (R_LS)
for j = 1:npts;
   jMach(j) = fix(20*M(j)-4);
   s_swp = cos(aw_swp);
   if M(j) > 0.25;
      R_LS_l(j) = R_LS(jMach(j),1)*s_swp^6+R_LS(jMach(j),2)*s_swp^5+R_LS(jMach(j),3)*s_swp^4+R_LS(jMach(j),4)*s_swp^3+R_LS(jMach(j),5)*s_swp^2+R_LS(jMach(j),6)*s_swp+R_LS(jMach(j),7);
      R_LS_h(j) = R_LS(jMach(j)+1,1)*s_swp^6+R_LS(jMach(j)+1,2)*s_swp^5+R_LS(jMach(j)+1,3)*s_swp^4+R_LS(jMach(j)+1,4)*s_swp^3+R_LS(jMach(j)+1,5)*s_swp^2+R_LS(jMach(j)+1,6)*s_swp+R_LS(jMach(j)+1,7);
      Mach_lw(j) = (jMach(j)+4)/20;
      Mach_hi(j) = (jMach(j)+5)/20;
      R_LS_s(j) = ((M(j)-Mach_lw(j))/(Mach_hi(j)-Mach_lw(j)))*(R_LS_h(j)-R_LS_l(j))+R_LS_l(j);
   else
      R_LS_s(j) = R_LS(1,1)*s_swp^6+R_LS(1,2)*s_swp^5+R_LS(1,3)*s_swp^4+R_LS(1,4)*s_swp^3+R_LS(1,5)*s_swp^2+R_LS(1,6)*s_swp+R_LS(1,7);
   end
end
R_LS_aw = R_LS_s';

% AW Turbulent Mean Skin-Friction Coefficient (C_f)
RN_s = RNL*aw_MAC;
for k = 1:npts;
   kMach(k) = fix(2*M(k)+1);
   kRN(k) = fix(1.290086*log(RN_s(k))-16.958276);
   kMach_l(k) = (kMach(k)-1)/2;
   kMach_h(k) = (kMach(k))/2;
   kRN_l(k) = exp((kRN(k)+16.958276)/1.290086);
   kRN_h(k) = exp((kRN(k)+17.958276)/1.290086);
   C_f_l(k) = ((M(k)-kMach_l(k))/(kMach_h(k)-kMach_l(k)))*(C_f(kMach(k),kRN(k))-C_f(kMach(k)+1,kRN(k)))+C_f(kMach(k),kRN(k));
   C_f_h(k) = ((M(k)-kMach_l(k))/(kMach_h(k)-kMach_l(k)))*(C_f(kMach(k),kRN(k)+1)-C_f(kMach(k)+1,kRN(k)+1))+C_f(kMach(k),kRN(k)+1);
   C_f_s(k) = ((RN_s(k)-kRN_l(k))/(kRN_h(k)-kRN_l(k)))*(C_f_h(k)-C_f_l(k))+C_f_l(k);
end
C_f_aw = C_f_s';

% AW Airfoil Thickness Location Parameter (L')
t_c = aw_t_c_max;
L_p_l = 1+1.2*t_c+100*t_c^4;
L_p_h = 1+2*t_c+100*t_c^4;
K_aw = (L_p_l+L_p_h)/2;

% AW Wetted Area Calculation
aw_S_wet = (2*(2*aw_S))*(1+0.25*(aw_t_c_max));

% AW Equivalent Parasite Area
f_aw = (R_wf_s).*(R_LS_aw).*(C_f_aw)*(K_aw)*aw_S_wet;



% TAIL

% Tail Lifting Surface Correction Factor (R_LS)
for j = 1:npts;
```

```
      jMach(j) = fix(20*M(j)-4);
      s_swp = cos(t_swp);
      if M(j) > 0.25;
         R_LS_l(j) =
R_LS(jMach(j),1)*s_swp^6+R_LS(jMach(j),2)*s_swp^5+R_LS(jMach(j),3)*s_swp^4+R_LS(jMach(j),4)*s_swp^3+R
_LS(jMach(j),5)*s_swp^2+R_LS(jMach(j),6)*s_swp+R_LS(jMach(j),7);
         R_LS_h(j) =
R_LS(jMach(j)+1,1)*s_swp^6+R_LS(jMach(j)+1,2)*s_swp^5+R_LS(jMach(j)+1,3)*s_swp^4+R_LS(jMach(j)+1,4)*s
_swp^3+R_LS(jMach(j)+1,5)*s_swp^2+R_LS(jMach(j)+1,6)*s_swp+R_LS(jMach(j)+1,7);
         Mach_lw(j) = (jMach(j)+4)/20;
         Mach_hi(j) = (jMach(j)+5)/20;
         R_LS_s(j) = ((M(j)-Mach_lw(j))/(Mach_hi(j)-Mach_lw(j)))*(R_LS_h(j)-R_LS_l(j))+R_LS_l(j);
      else
         R_LS_s(j) =
R_LS(1,1)*s_swp^6+R_LS(1,2)*s_swp^5+R_LS(1,3)*s_swp^4+R_LS(1,4)*s_swp^3+R_LS(1,5)*s_swp^2+R_LS(1,6
)*s_swp+R_LS(1,7);
      end
   end
   R_LS_t = R_LS_s';

   % Tail Turbulent Mean Skin-Friction Coefficient (C_f)
   RN_s = RNL*t_MAC;
   for k = 1:npts;
      kMach(k) = fix(2*M(k)+1);
      kRN(k) = fix(1.290086*log(RN_s(k))-16.958276);
      kMach_l(k) = (kMach(k)-1)/2;
      kMach_h(k) = (kMach(k))/2;
      kRN_l(k) = exp((kRN(k)+16.958276)/1.290086);
      kRN_h(k) = exp((kRN(k)+17.958276)/1.290086);
      C_f_l(k) = ((M(k)-kMach_l(k))/(kMach_h(k)-kMach_l(k)))*(C_f(kMach(k),kRN(k))-
C_f(kMach(k)+1,kRN(k)))+C_f(kMach(k),kRN(k));
      C_f_h(k) = ((M(k)-kMach_l(k))/(kMach_h(k)-kMach_l(k)))*(C_f(kMach(k),kRN(k)+1)-
C_f(kMach(k)+1,kRN(k)+1))+C_f(kMach(k),kRN(k)+1);
      C_f_s(k) = ((RN_s(k)-kRN_l(k))/(kRN_h(k)-kRN_l(k)))*(C_f_h(k)-C_f_l(k))+C_f_l(k);
   end
   C_f_t = C_f_s';

   % Tail Airfoil Thickness Location Parameter (L')
   t_c = t_t_c_max;
   L_p_l = 1+1.2*t_c+100*t_c^4;
   L_p_h = 1+2*t_c+100*t_c^4;
   K_t = (L_p_l+L_p_h)/2;

   % Tail Wetted Area Calculation
   t_S_wet = (2*t_S)*(1+0.25*(t_t_c_max));

   % Tail Equivalent Parasite Area
   f_t = (R_wf_s).*(R_LS_t).*(C_f_t)*(K_t)*t_S_wet;



   % FUSELAGE

   % Fuselage Turbulent Mean Skin-Friction Coefficient (C_f)
   RN_s = RNL*L_fuselage;
   for k = 1:npts;
      kMach(k) = fix(2*M(k)+1);
      kRN(k) = fix(1.290086*log(RN_s(k))-16.958276);
      kMach_l(k) = (kMach(k)-1)/2;
      kMach_h(k) = (kMach(k))/2;
      kRN_l(k) = exp((kRN(k)+16.958276)/1.290086);
      kRN_h(k) = exp((kRN(k)+17.958276)/1.290086);
```

```
      C_f_l(k)                    =                    ((M(k)-kMach_l(k))/(kMach_h(k)-kMach_l(k)))*(C_f(kMach(k),kRN(k))-
C_f(kMach(k)+1,kRN(k)))+C_f(kMach(k),kRN(k));
      C_f_h(k)                    =                    ((M(k)-kMach_l(k))/(kMach_h(k)-kMach_l(k)))*(C_f(kMach(k),kRN(k)+1)-
C_f(kMach(k)+1,kRN(k)+1))+C_f(kMach(k),kRN(k)+1);
      C_f_s(k) = ((RN_s(k)-kRN_l(k))/(kRN_h(k)-kRN_l(k)))*(C_f_h(k)-C_f_l(k))+C_f_l(k);
   end
   C_f_fuse = C_f_s';

   % Fuselage Wetted Area Calculation
   fuse_wet = L_fuselage*D_fuselage*pi;

   % Fuselage Equivalent Parasite Area
   f_fuse = C_f_fuse*(1+60/LD_fuse^3+0.0025*LD_fuse)*fuse_wet;



   % HINGES

   % Hinges Equivalent Parasite Area
   f_hinges = 0.20;


   % TOTAL ROSKAM AIRCRAFT DRAG CALCULATION

   % Parasite Drag of Wings Only
   CD_v_wings_R = (f_fiw+f_fow+f_aw)/S;

   % Summation of Equivalent Parasite Areas
   f_ac_R = f_fiw+f_fow+f_aw+f_t+f_fuse+f_hinges;

   % Aircraft Pressure Drag Coefficient
   CD_v_ac_R = f_ac_R/S;

%=========================================================================

% AIRCRAFT EFFICIENCIES

   % Efficiency Calculations
   TR = fow_c_t/fiw_c_r;
   tou_AR = 0.0457*TR^4-0.1367*TR^3+0.1586*TR^2-0.0704*TR+0.0124;
   tou = AR*tou_AR;
   e_span = 1/(1+tou);                          % Span Efficiency
   r = 0.010;                                   % Correction Factor
   e_osw = 1/(pi*AR*r+1+tou);                   % Oswald Efficiency
   k = 1/(pi*AR*e_osw);

%=========================================================================

% THRUST AVAILABLE AND TSFC INTERPOLATION THROUGHOUT CRUISE

   % Thrust Available and TSFC
   TSFC_climb_cruise = TSFC_1*ones(cr_e,1);
   TSFC_descent = TSFC_2*ones(npts-cr_e,1);
   TSFC = [TSFC_climb_cruise;TSFC_descent];
   Ta = 1000*ones(npts,1);

%=========================================================================

% WEIGHT, LIFT, AND DRAG CALCULATIONS FOR ROSKAM

   % Flight Profile Endpoints
   aout(npts) = 0;
```

```
thetaout(npts) = 0;
RCout(npts) = 0;
tout(npts) = t(npts);
hout(npts) = h(npts);
theta = thetaout*(pi/180);

% Roskam Fuel Assumptions
W_f_R_extra = 5500*cf_m;                    % Coded in (kg), coverted to (lbs)

% Roskam Intial Weight Settings
W_f_R(1) = W_f_base+W_f_R_extra-W_f_clm;
W_R(1) = W_ac+W_f_R(1);

% Fuel Burn Calculation
for n = 1:npts-1;
   CL_R(n) = (2*(W_R(n)/S))/(gama*pinf(n)*M(n)^2);
   CD_i_R(n) = (CL_R(n)^2/(pi*AR))+(CL_R(n)^2/(pi*AR))*tou;
   CD_t_R(n) = CD_v_ac_R(n)+CD_i_R(n);
   Treq_R(n) = W_R(n)*(aout(n)/g)+CD_t_R(n)*q(n)*S+W_R(n)*sin(theta(n));
   W_R(n+1) = W_R(n)*(1/exp((TSFC(n)*CD_t_R(n)*(x(n+1)-x(n)))/(375*np*CL_R(n))));
   W_f_R(n+1) = W_f_R(1)-(W_R(1)-W_R(n+1));
end

% Aerodynamic Forces At Endpoints
CL_R(npts) = (2*(W_R(npts)/S))/(gama*pinf(npts)*M(npts)^2);
CD_i_R(npts) = (CL_R(npts)^2/(pi*AR))+(CL_R(npts)^2/(pi*AR))*tou;
CD_t_R(npts) = CD_v_ac_R(npts)+CD_i_R(npts);
Treq_R(npts) = W_R(npts)*(aout(npts)/g)+CD_t_R(npts)*q(npts)*S+W_R(npts)*sin(theta(npts));

% Redistribution Into Columns
W_R = W_R';
W_f_R = W_f_R';
CL_R = CL_R';
CD_i_R = CD_i_R';
CD_t_R = CD_t_R';
Treq_R = Treq_R';

% Profile Drag
CD0_array_R = CD_t_R-k*CL_R.^2;
CD0_R = mean(CD0_array_R(1:npts));

%=========================================================================

% AERODYNAMIC RESULTS FOR ROSKAM

% Aerodynamic Characteristics
Em_R = 1/(2*sqrt(k*CD0_R));
L_D_R = CL_R./CD_t_R;
W_S_R = W_R/S;

% Fuel/Time Requirements
W_F_req_R = W_R(1)-W_R(npts);
V_F_req_R = W_F_req_R/6;                % Volume of Fuel Required (Gal)
TOS_R = (t(cr_e)-t(cr_b))/60;

% Specific Excess Power
Ps_R = Ta-Treq_R;

% Display Aerodynamic Results
disp(' ')
disp(' ')
disp('Aerodynamic Results For Roskam Method (English):')
```

```
   disp('  t(min)   h(ft)   Range(miles)   Weight(lbs)   Fuel(lbs)   Mach  Velocity(fps) CL      CD      L/D')
   disp('-------------------------------------------------------------------------------------------------------')
   fprintf('%8.1f        %8.1f          %8.2f              %8.2f              %8.2f      %8.3f      %8.3f   %8.3f   %8.4f
%8.2f\n',[t,h,x,W_R,W_f_R,M,V,CL_R,CD_t_R,L_D_R]')

   disp(' ')
   disp(' ')
   disp('Aerodynamic Results For Roskam Method (Metric):')
   disp('  t(min)    h(m)     Range(km)    Weight(kg)   Fuel(kg)    Mach  Velocity(mps) CL      CD      L/D')
   disp('-------------------------------------------------------------------------------------------------------')
   fprintf('%8.1f        %8.1f          %8.2f              %8.2f              %8.2f      %8.3f      %8.3f   %8.3f   %8.4f
%8.2f\n',[t,h/cf_l,x/cf_d,W_R/cf_m,W_f_R/cf_m,M,V/cf_l,CL_R,CD_t_R,L_D_R]')

%=========================================================================

%  RAs AND RApa DRAG BUILDUP

   Joined_Wing_Drag_Polars
   Joined_Wing_AVTIE_Outputs
   Joined_Wing_AVTIE_Data

   % Equivalent Parasite Area Without Wings (from Roskam)
   f_without_wings = f_t+f_fuse+f_hinges;



   % ROSKAM/AVTIE STRIP METHOD (RAs)

   % Equivalent Parasite Area Of Wings
   f_wings_RAs = CDinf_v_total_s*S;

   % Total Aircraft Drag Pressure Coefficient
   f_ac_RAs = f_without_wings+f_wings_RAs;
   CD_v_ac_RAs = f_ac_RAs/S;

   % Total Aircraft Drag Coefficient Including Induced Drag
   CD_ac_RAs = CD_v_ac_RAs+CDinf_i_total_s;



   % ROSKAM/AVTIE PAN-AIR METHOD (RApa)

   % Equivalent Parasite Area Of Wings
   f_wings_RApa = CDinf_v_total_s*S;

   % Total Aircraft Drag Pressure Coefficient
   f_ac_RApa = f_without_wings+f_wings_RApa;
   CD_v_ac_RApa = f_ac_RApa/S;

   % Total Aircraft Drag Coefficient Including Induced Drag
   CD_ac_RApa = CD_v_ac_RApa+CD_i_pa;



   % TWIST DISTRIBUTION 1 (0, 0, 0, 0)
   f_wings_twst1 = CDvtwst_tot_1*S;
   f_ac_twst1 = f_without_wings(4)+f_wings_twst1;
   CD_v_ac_twst1 = f_ac_twst1/S;
   CD_ac_twst1 = CD_v_ac_twst1+CDi_twst_1;
   L_D_twst1 = CLtwst_tot_1/CD_ac_twst1;

   % TWIST DISTRIBUTION 2 (1, -1, 0, -1)
```

```
f_wings_twst2 = CDvtwst_tot_2*S;
f_ac_twst2 = f_without_wings(4)+f_wings_twst2;
CD_v_ac_twst2 = f_ac_twst2/S;
CD_ac_twst2 = CD_v_ac_twst2+CDi_twst_2;
L_D_twst2 = CLtwst_tot_2/CD_ac_twst2;

% TWIST DISTRIBUTION 3 (2, -1, 0, -2)
f_wings_twst3 = CDvtwst_tot_3*S;
f_ac_twst3 = f_without_wings(4)+f_wings_twst3;
CD_v_ac_twst3 = f_ac_twst3/S;
CD_ac_twst3 = CD_v_ac_twst3+CDi_twst_3;
L_D_twst3 = CLtwst_tot_3/CD_ac_twst3;

% TWIST DISTRIBUTION 4 (3, -2, 0, -3)
f_wings_twst4 = CDvtwst_tot_4*S*1.05;
f_ac_twst4 = f_without_wings(4)+f_wings_twst4;
CD_v_ac_twst4 = f_ac_twst4/S;
CD_ac_twst4 = CD_v_ac_twst4+CDi_twst_4;
L_D_twst4 = CLtwst_tot_4/CD_ac_twst4;

% TWIST COMPARISON
swt = [1;8;9;10];
CL_twstd = [CLtwst_tot_1,CLtwst_tot_2,CLtwst_tot_3,CLtwst_tot_4]';
CD_v_twstd = [CD_v_ac_twst1,CD_v_ac_twst2,CD_v_ac_twst3,CD_v_ac_twst4]';
CD_i_twstd = [CDi_twst_1,CDi_twst_2,CDi_twst_3,CDi_twst_4]';
CD_twstd = [CD_ac_twst1,CD_ac_twst2,CD_ac_twst3,CD_ac_twst4]';
L_D_twstd = [L_D_twst1,L_D_twst2,L_D_twst3,L_D_twst4]';



% DISPLAY WING LIFT/DRAG DISTRIBUTIONS
% Strip Locations
   spn_iw = strips(1:8)*cf_l;
   spn_jt = strips(17:20)*cf_l;
   spn_ow = strips(21:36)*cf_l;
% Span Correction Factors
   base_chord = 2.5;
   ib = 2.5/base_chord;
   ob = 2.5/base_chord;
   jt = [4.7,4.1,3.8,3.4]/base_chord;

% Display Lift Coefficient Distribution From Strip Method (PT 4)
CL_fiw = ib*CL_4_strip(1:8);
CL_aw = ib*CL_4_strip(9:16);
CL_iw = CL_fiw+CL_aw;
for i = 1:4;
   CL_joint(i) = CL_4_strip(i+16)*jt(i);
end
CL_fow = ob*CL_4_strip(21:36);
plot(spn_iw,CL_fiw,'bv-'),hold on,plot(spn_iw,CL_aw,'r^-'),hold on,plot(spn_iw,CL_iw,'md-'),hold on
plot(spn_jt,CL_joint,'gs-'), hold on,plot(spn_ow,CL_fow,'co-')
xlabel('Spanwise Strip Location (ft)'),ylabel('Sectional Lift Coefficient Per Unit Span ( C _L )')
legend('Foward Inside Wing','Aft Wing','Total FIW + AW','Wing Joint','Outboard Wing'),legend boxoff,figure

% Display Lift Distribution From Strip Method (PT 4)
L_fiw = CL_4_strip(1:8)*qinf(4).*A_4_strip(1:8);
L_aw = CL_4_strip(9:16)*qinf(4).*A_4_strip(9:16);
L_iw = L_fiw+L_aw;
L_joint = CL_4_strip(17:20)*qinf(4).*A_4_strip(17:20);
L_fow = CL_4_strip(21:36)*qinf(4).*A_4_strip(21:36);
plot(spn_iw,L_fiw,'bv-'),hold on,plot(spn_iw,L_aw,'r^-'),hold on,plot(spn_iw,L_iw,'md-'),hold on
plot(spn_jt,L_joint,'gs-'), hold on,plot(spn_ow,L_fow,'co-'),hold on
```

```
xlabel('Spanwise Strip Location (ft)'),ylabel('Sectional Lift (lbs)')
legend('Foward Inside Wing','Aft Wing','Total FIW + AW','Wing Joint','Outboard Wing'),legend boxoff,figure

% Display Strip Viscous Drag Distribution From Stirp Method (PT 4)
CD_v_fiw = ib*CDinf_v_4_strip(1:8);
CD_v_aw = ib*CDinf_v_4_strip(9:16);
CD_v_iw = CD_v_fiw+CD_v_aw;
for i = 1:4;
   CD_v_joint(i) = CDinf_v_4_strip(i+16)*jt(i);
end
CD_v_fow = ob*CDinf_v_4_strip(21:36);
plot(spn_iw,CD_v_fiw,'bv-'),hold on,plot(spn_iw,CD_v_aw,'r^-'),hold on,plot(spn_iw,CD_v_iw,'md-'),hold on
plot(spn_jt,CD_v_joint,'gs-'), hold on,plot(spn_ow,CD_v_fow,'co-')
xlabel('Spanwise Strip Location (ft)'),ylabel('Sectional Parasite Drag Coefficient Per Unit Span ( C_D_p )')
legend('Foward Inside Wing','Aft Wing','Total FIW + AW','Wing Joint','Outboard Wing'),legend boxoff,figure

% Display Strip Induced Drag Distribution From Strip Method (PT 4)
CD_i_fiw = ib*CDinf_i_4_strip(1:8);
CD_i_aw = ib*CDinf_i_4_strip(9:16);
CD_i_iw = CD_i_fiw+CD_i_aw;
for i = 1:4;
   CD_i_joint(i) = CDinf_i_4_strip(i+16)*jt(i);
end
CD_i_fow = ob*CDinf_i_4_strip(21:36);
plot(spn_iw,CD_i_fiw,'bv-'),hold on,plot(spn_iw,CD_i_aw,'r^-'),hold on,plot(spn_iw,CD_i_iw,'md-'),hold on
plot(spn_jt,CD_i_joint,'gs-'), hold on,plot(spn_ow,CD_i_fow,'co-')
xlabel('Spanwise Strip Location (ft)'),ylabel('Sectional Induced Drag Coefficient Per Unit Span ( C_D_i )')
legend('Foward Inside Wing','Aft Wing','Total FIW + AW','Wing Joint','Outboard Wing',4),legend boxoff,figure

% Display All AOA Distributions From Strip Method (PT 4)
% AOA
   AOA_fiw = alpha_A_4(1:8);
   AOA_aw = alpha_A_4(9:16);
   AOA_joint = alpha_A_4(17:20);
   AOA_fow = alpha_A_4(21:36);
   plot(spn_iw,AOA_fiw,'bv-'),hold on,plot(spn_iw,AOA_aw,'r^-'),hold on
   plot(spn_jt,AOA_joint,'gs-'), hold on,plot(spn_ow,AOA_fow,'co-')
   xlabel('Spanwise Strip Location (ft)'),ylabel('Sectional Angle of Attack ( \alpha )')
   legend('Foward Inside Wing','Aft Wing','Wing Joint','Outboard Wing',4),legend boxoff,figure
% AOA Local
   AOAL_fiw = aoaL_4_strip(1:8);
   AOAL_aw = aoaL_4_strip(9:16);
   AOAL_joint = aoaL_4_strip(17:20);
   AOAL_fow = aoaL_4_strip(21:36);
   plot(spn_iw,AOAL_fiw,'bv-'),hold on,plot(spn_iw,AOAL_aw,'r^-'),hold on
   plot(spn_jt,AOAL_joint,'gs-'), hold on,plot(spn_ow,AOAL_fow,'co-')
   xlabel('Spanwise Strip Location (ft)'),ylabel('Sectional Local Angle of Attack ( \alpha _L )')
   legend('Foward Inside Wing','Aft Wing','Wing Joint','Outboard Wing',4),legend boxoff,figure
% AOA Induced
   AOAi_fiw = AOAi_1_strip(1:8);
   AOAi_aw = AOAi_1_strip(9:16);
   AOAi_joint = AOAi_1_strip(17:20);
   AOAi_fow = AOAi_1_strip(21:36);
   plot(spn_iw,AOAi_fiw,'bv-'),hold on,plot(spn_iw,AOAi_aw,'r^-'),hold on
   plot(spn_jt,AOAi_joint,'gs-'), hold on,plot(spn_ow,AOAi_fow,'co-')
   xlabel('Spanwise Strip Location (ft)'),ylabel('Sectional Induced Angle of Attack ( \alpha _i )')
   legend('Foward Inside Wing','Aft Wing','Wing Joint','Outboard Wing'),legend boxoff,figure

% Display Lift Coefficient Distributions For Twisted Runs
% Twist Run 1 (0, 0, 0, 0)
   CL_fiw_tr1 = ib*CL_twst_1(1:8);
   CL_aw_tr1 = ib*CL_twst_1(9:16);
```

```
    CL_iw_tr1 = CL_fiw_tr1+CL_aw_tr1;
    for i = 1:4;
       CL_joint_tr1(i) = CL_twst_1(i+16)*jt(i);
    end
    CL_fow_tr1 = ob*CL_twst_1(21:36);
    plot(spn_iw,CL_fiw_tr1,'bv-'),hold on,plot(spn_iw,CL_aw_tr1,'r^-'),hold on,plot(spn_iw,CL_iw_tr1,'md-'),hold on
    plot(spn_jt,CL_joint_tr1,'gs-'), hold on,plot(spn_ow,CL_fow_tr1,'co-')
    xlabel('Spanwise Strip Location (ft)'),ylabel('Sectional Lift Coefficient Per Unit Span ( C _L )')
    legend('Foward Inside Wing','Aft Wing','Total FIW + AW','Wing Joint','Outboard Wing'),legend boxoff,figure
% Twist Run 2 (1, -1, 0, -1)
    CL_fiw_tr2 = ib*CL_twst_2(1:8);
    CL_aw_tr2 = ib*CL_twst_2(9:16);
    CL_iw_tr2 = CL_fiw_tr2+CL_aw_tr2;
    for i = 1:4;
       CL_joint_tr2(i) = CL_twst_2(i+16)*jt(i);
    end
    CL_fow_tr2 = ob*CL_twst_2(21:36);
    plot(spn_iw,CL_fiw_tr2,'bv-'),hold on,plot(spn_iw,CL_aw_tr2,'r^-'),hold on,plot(spn_iw,CL_iw_tr2,'md-'),hold on
    plot(spn_jt,CL_joint_tr2,'gs-'), hold on,plot(spn_ow,CL_fow_tr2,'co-')
    xlabel('Spanwise Strip Location (ft)'),ylabel('Sectional Lift Coefficient Per Unit Span ( C _L )')
    legend('Foward Inside Wing','Aft Wing','Total FIW + AW','Wing Joint','Outboard Wing'),legend boxoff,figure
 % Twist Run 3 (2, 0, -1, -2)
    CL_fiw_tr3 = ib*CL_twst_3(1:8);
    CL_aw_tr3 = ib*CL_twst_3(9:16);
    CL_iw_tr3 = CL_fiw_tr3+CL_aw_tr3;
    for i = 1:4;
       CL_joint_tr3(i) = CL_twst_3(i+16)*jt(i);
    end
    CL_fow_tr3 = ob*CL_twst_3(21:36);
    plot(spn_iw,CL_fiw_tr3,'bv-'),hold on,plot(spn_iw,CL_aw_tr3,'r^-'),hold on,plot(spn_iw,CL_iw_tr3,'md-'),hold on
    plot(spn_jt,CL_joint_tr3,'gs-'), hold on,plot(spn_ow,CL_fow_tr3,'co-')
    xlabel('Spanwise Strip Location (ft)'),ylabel('Sectional Lift Coefficient Per Unit Span ( C _L )')
    legend('Foward Inside Wing','Aft Wing','Total FIW + AW','Wing Joint','Outboard Wing'),legend boxoff,figure
% Twist Run 4 (3, -2, 0, -3)
    CL_fiw_tr4 = ib*CL_twst_4(1:8);
    CL_aw_tr4 = ib*CL_twst_4(9:16);
    CL_iw_tr4 = CL_fiw_tr4+CL_aw_tr4;
    for i = 1:4;
       CL_joint_tr4(i) = CL_twst_4(i+16)*jt(i);
    end
    CL_fow_tr4 = ob*CL_twst_4(21:36);
    plot(spn_iw,CL_fiw_tr4,'bv-'),hold on,plot(spn_iw,CL_aw_tr4,'r^-'),hold on,plot(spn_iw,CL_iw_tr4,'md-'),hold on
    plot(spn_jt,CL_joint_tr4,'gs-'), hold on,plot(spn_ow,CL_fow_tr4,'co-')
    xlabel('Spanwise Strip Location (ft)'),ylabel('Sectional Lift Coefficient Per Unit Span ( C _L )')
    legend('Foward Inside Wing','Aft Wing','Total FIW + AW','Wing Joint','Outboard Wing'),legend boxoff,figure

%=========================================================================

% WEIGHT, LIFT, AND DRAG CALCULATIONS FOR RAs AND RApa METHODS

  % AVTIE Method Fuel Assumptions
    W_f_A_extra = 0*cf_m;                    % Coded in (kg), coverted to (lbs)
  % AVTIE Method Initial Weight Settings
    W_f_A(1) = W_f_base+W_f_A_extra-W_f_clm;
    W_A(1) = W_ac+W_f_A(1);

  % Roskam/AVTIE Strip Method Fuel Assumptions
    W_f_RAs_extra = 0*cf_m;                    % Coded in (kg), coverted to (lbs)
  % Roskam/AVTIE Strip Method Initial Weight Settings
    W_f_RAs(1) = W_f_base+W_f_RAs_extra-W_f_clm;
    W_RAs(1) = W_ac+W_f_RAs(1);
```

```
% Roskam/AVTIE Pan Air Method Fuel Assumptions
   W_f_RApa_extra = 0*cf_m;                    % Coded in (kg), coverted to (lbs)
% Roskam/AVTIE Pan Air Method Initial Weight Settings
   W_f_RApa(1) = W_f_base+W_f_RApa_extra-W_f_clm;
   W_RApa(1) = W_ac+W_f_RApa(1);

% Fuel Burn Calculation
for n = 1:npts-1;
   % AVTIE Method (Wing Only) (A)
      CL_A(n) = CL_A(n);
      CL_A_ch(n) = (2*(W_A(n)/S))/(gama*pinf(n)*M(n)^2);
      W_f_A(n+1) = fuel_A(n+1)*cf_m;
      W_A(n+1) = W_ac+W_f_A(n+1);
      Treq_A(n) = W_A(n)*(aout(n)/g)+CD_t_A(n)*q(n)*S+W_A(n)*sin(theta(n));
   % Roskam/AVTIE Strip Method (RAs)
      CL_RAs(n) = CLinf_total_s(n);
      CL_RAs_ch(n) = (2*(W_RAs(n)/S))/(gama*pinf(n)*M(n)^2);
      W_RAs(n+1) = W_RAs(n)*(1/exp((TSFC(n)*CD_ac_RAs(n)*(x(n+1)-x(n)))/(375*np*CL_RAs(n))));
      W_f_RAs(n+1) = W_f_RAs(1)-(W_RAs(1)-W_RAs(n+1));
      Treq_RAs(n) = W_RAs(n)*(aout(n)/g)+CD_ac_RAs(n)*q(n)*S+W_RAs(n)*sin(theta(n));
   % Roskam/AVTIE Pan Air Method (RApa)
      CL_RApa(n) = CLinf_total_s(n);
      CL_RApa_ch(n) = (2*(W_RApa(n)/S))/(gama*pinf(n)*M(n)^2);
      W_RApa(n+1) = W_RApa(n)*(1/exp((TSFC(n)*CD_ac_RApa(n)*(x(n+1)-x(n)))/(375*np*CL_RApa(n))));
      W_f_RApa(n+1) = W_f_RApa(1)-(W_RApa(1)-W_RApa(n+1));
      Treq_RApa(n) = W_RApa(n)*(aout(n)/g)+CD_ac_RApa(n)*q(n)*S+W_RApa(n)*sin(theta(n));
end

% Aerodynamic Forces At Endpoints
   % AVTIE Method (Wing Only) (A)
      CL_A(npts) = CL_A(npts);
      CL_A_ch(npts) = (2*(W_A(npts)/S))/(gama*pinf(npts)*M(npts)^2);
      Treq_A(npts) = W_A(npts)*(aout(npts)/g)+CD_t_A(n)*q(npts)*S+W_A(npts)*sin(theta(npts));
   % Roskam/AVTIE Strip Method (RAs)
      CL_RAs(npts) = CLinf_total_s(npts);
      CL_RAs_ch(npts) = (2*(W_RAs(npts)/S))/(gama*pinf(npts)*M(npts)^2);
      Treq_RAs(npts) = W_RAs(npts)*(aout(npts)/g)+CD_ac_RAs(npts)*q(npts)*S+W_RAs(npts)*sin(theta(npts));
   % Roskam/AVTIE Pan Air Method (RApa)
      CL_RApa(npts) = CLinf_total_s(npts);
      CL_RApa_ch(npts) = (2*(W_RApa(npts)/S))/(gama*pinf(npts)*M(npts)^2);
      Treq_RApa(npts)                                                                                    =
W_RApa(npts)*(aout(npts)/g)+CD_ac_RApa(npts)*q(npts)*S+W_RApa(npts)*sin(theta(npts));

% Redistribution Into Columns
   % AVTIE Method (Wing Only) (A)
      CL_A_ch = CL_A_ch';
      Treq_A = Treq_A';
      W_A = W_A';
      W_f_A = W_f_A';
   % Roskam/AVTIE Strip Method (RAs)
      CL_RAs = CL_RAs';
      CL_RAs_ch = CL_RAs_ch';
      Treq_RAs = Treq_RAs';
      W_RAs = W_RAs';
      W_f_RAs = W_f_RAs';
   % Roskam/AVTIE Pan Air Method (RApa)
      CL_RApa = CL_RApa';
      CL_RApa_ch = CL_RApa_ch';
      Treq_RApa = Treq_RApa';
      W_RApa = W_RApa';
      W_f_RApa = W_f_RApa';
```

```
   % Profile Drag
      % AVTIE Method (Wing Only) (A)
         CD0_array_A = CD_t_A-k*CL_A.^2;
         CD0_A = mean(CD0_array_A);
      % Roskam/AVTIE Strip Method (RAs)
         CD0_array_RAs = CD_ac_RAs-k*CL_RAs.^2;
         CD0_RAs = mean(CD0_array_RAs);
      % Roskam/AVTIE Pan Air Method (RApa)
         CD0_array_RApa = CD_ac_RApa-k*CL_RApa.^2;
         CD0_RApa = mean(CD0_array_RApa);

%=========================================================================

% AERODYNAMIC RESULTS FOR RAs AND RApa METHODS

   % Aerodynamic Characteristics
      % AVTIE Method (Wing Only) (A)
         Em_A = 1/(2*sqrt(k*CD0_A));
         L_D_A = CL_A./CD_t_A;
         W_S_A = W_A/S;
      % Roskam/AVTIE Strip Method (RAs)
         Em_RAs = 1/(2*sqrt(k*CD0_RAs));
         L_D_RAs = CL_RAs./CD_ac_RAs;
         W_S_RAs = W_RAs/S;
      % Roskam/AVTIE Pan Air Method (RApa)
         Em_RApa = 1/(2*sqrt(k*CD0_RApa));
         L_D_RApa = CL_RApa./CD_ac_RApa;
         W_S_RApa = W_RApa/S;

   % Fuel/Time Requirements
      % AVTIE Method (Wing Only) (A)
         W_F_req_A = W_A(1)-W_A(npts);
         V_F_req_A = W_F_req_A/6;                  % Volume of Fuel Required (gal)
         TOS_A = (t(cr_e)-t(cr_b))/60;
      % Roskam/AVTIE Strip Method (RAs)
         W_F_req_RAs = W_RAs(1)-W_RAs(npts);
         V_F_req_RAs = W_F_req_RAs/6;              % Volume of Fuel Required (gal)
         TOS_RAs = (t(cr_e)-t(cr_b))/60;
      % Roskam/AVTIE Pan Air Method (RApa)
         W_F_req_RApa = W_RApa(1)-W_RApa(npts);
         V_F_req_RApa = W_F_req_RApa/6;            % Volume of Fuel Required (gal)
         TOS_RApa = (t(cr_e)-t(cr_b))/60;

   % Specific Excess Power
      % AVTIE Method (Wing Only) (A)
         Ps_A = Ta-Treq_A;
      % Roskam/AVTIE Strip Method (RAs)
         Ps_RAs = Ta-Treq_RAs;
      % Roskam/AVTIE Pan Air Method (RApa)
         Ps_RApa = Ta-Treq_RApa;

   % Display Aerodynamic Results
   disp(' ')
   disp(' ')
   disp('Aerodynamic Results For AVTIE Method (Metric):')
   disp('   t(min)    h(m)     Range(km)   Weight(kg)  Fuel(kg)    Mach  Velocity(mps) CL      CD      L/D')
   disp('-------------------------------------------------------------------------------------------------------')
   fprintf('%8.1f         %8.1f            %8.2f                %8.2f             %8.2f      %8.3f      %8.3f   %8.3f   %8.4f
%8.2f\n',[t,h/cf_l,x/cf_d,W_A/cf_m,W_f_A/cf_m,M,V/cf_l,CL_A,CD_t_A,L_D_A]')

   disp(' ')
   disp(' ')
```

```
   disp('Aerodynamic Results For Roskam/AVTIE Strip Method (Metric):')
   disp('   t(min)    h(m)     Range(km)   Weight(kg)  Fuel(kg)    Mach  Velocity(mps) CL      CD      L/D')
   disp('-------------------------------------------------------------------------------------------------')
   fprintf('%8.1f        %8.1f        %8.2f              %8.2f            %8.2f     %8.3f     %8.3f   %8.3f   %8.4f
%8.2f\n',[t,h/cf_l,x/cf_d,W_RAs/cf_m,W_f_RAs/cf_m,M,V/cf_l,CL_RAs,CD_ac_RAs,L_D_RAs]')

   disp(' ')
   disp(' ')
   disp('Aerodynamic Results For Roskam/AVTIE Pan Air Method (Metric):')
   disp('   t(min)    h(m)     Range(km)   Weight(kg)  Fuel(kg)    Mach  Velocity(mps) CL      CD      L/D')
   disp('-------------------------------------------------------------------------------------------------')
   fprintf('%8.1f        %8.1f        %8.2f              %8.2f            %8.2f     %8.3f     %8.3f   %8.3f   %8.4f
%8.2f\n',[t,h/cf_l,x/cf_d,W_RApa/cf_m,W_f_RApa/cf_m,M,V/cf_l,CL_RApa,CD_ac_RApa,L_D_RApa]')

   disp(' ')
   disp(' ')
   disp('Comparison Between Drag Buildup Methods:')
   disp('|-------------------------------------------------------|-------------------------------------------------------|')
   disp('|                  Roskam                               |                  AVTIE                                |')
   disp('|-------------------------------------------------------|-------------------------------------------------------|')
   disp('|   CL      CDp      CDi      CD      L/D      CLch    Fuel(kg) |   CL      CDp      CDi      CD      L/D      CLch
Fuel(kg) |')
   disp('|-------------------------------------------------------|-------------------------------------------------------|')
   fprintf('%8.4f   %8.4f   %8.4f   %8.4f   %8.4f   %8.4f        %8.1f   %8.4f   %8.4f   %8.4f   %8.4f   %8.4f   %8.4f
%8.1f\n',[CL_R,CD_v_ac_R,CD_i_R,CD_t_R,L_D_R,CL_R,W_f_R/cf_m,CL_A,CD_v_A,CD_i_A,CD_t_A,L_D_A,
CL_A_ch,W_f_A/cf_m]')
   disp('|-------------------------------------------------------|-------------------------------------------------------|')
   disp('|              Roskam/AVTIE Pan Air                     |              Roskam/AVTIE Strip                       |')
   disp('|-------------------------------------------------------|-------------------------------------------------------|')
   disp('|   CL      CDp      CDi      CD      L/D      CLch    Fuel(kg) |   CL      CDp      CDi      CD      L/D      CLch
Fuel(kg) |')
   disp('|-------------------------------------------------------|-------------------------------------------------------|')
   fprintf('%8.4f   %8.4f   %8.4f   %8.4f   %8.4f   %8.4f      %8.1f   %8.4f   %8.4f   %8.4f   %8.4f   %8.4f   %8.4f      %8.1f
\n',[CL_RApa,CD_v_ac_RApa,CD_i_pa,CD_ac_RApa,L_D_RApa,CL_RApa_ch,W_f_RApa/cf_m,CL_RAs,CD_v_ac
_RAs,CDinf_i_total_s,CD_ac_RAs,L_D_RAs,CL_RAs_ch,W_f_RAs/cf_m]')

   disp(' ')
   disp(' ')
   disp('Twist Distribution Results (PT 4 Of Flight Profile):')
   disp('Twist Method     CL       CDp       CDi       CD       L/D')
   disp('-----------------------------------------------------------------')
   fprintf('%8.0f                                %8.5f          %8.5f              %8.5f              %8.5f
%8.5f\n',[swt,CL_twstd,CD_v_twstd,CD_i_twstd,CD_twstd,L_D_twstd]')

   disp(' ')
   disp(' ')
   disp('Strip Drag Buildup Results:')
   disp('|----------------|-------------------------------------------------------|-------------------------------------------------------
|')
   disp('|   Panel   Strip|                PT 1 (Re = 5.3547e6)                   |                PT 2 (Re = 3.9888e6)                   |')
   disp('|                | CL       CDp       CDi       AOA       aoaL       AOAi   | CL       CDp       CDi       AOA       aoaL
AOAi   |')
   disp('|----------------|-------------------------------------------------------|-------------------------------------------------------
|')
   fprintf('%8.0f%8.0f  %8.4f  %8.5f  %8.5f  %8.4f       %8.4f       %8.4f     %8.4f  %8.5f  %8.5f  %8.4f       %8.4f
%8.4f\n',[pan,strp,CL_1_strip,CD_v_1_strip,CDinf_i_1_strip,alpha_A_1,aoaL_1_strip,AOAi_1_strip,CL_2_strip,CD_
v_2_strip,CDinf_i_2_strip,alpha_A_2,aoaL_2_strip,AOAi_2_strip]')
   disp('|----------------|-------------------------------------------------------|-------------------------------------------------------
|')
   disp('|   Panel   Strip|                PT 3 (Re = 3.4312e6)                   |                PT 4 (Re = 2.5544e6)                   |')
   disp('|                | CL       CDp       CDi       AOA       aoaL       AOAi   | CL       CDp       CDi       AOA       aoaL
AOAi   |')
```

```
   disp('|---------------|-------------------------------------------------------------|-------------------------------------------------------------
|')
   fprintf('%8.0f%8.0f  %8.4f  %8.5f  %8.5f  %8.4f      %8.4f      %8.4f    %8.4f  %8.5f  %8.5f  %8.4f      %8.4f
%8.4f\n',[pan,strp,CL_3_strip,CD_v_3_strip,CDinf_i_3_strip,alpha_A_3,aoaL_3_strip,AOAi_3_strip,CL_4_strip,CD_
v_4_strip,CDinf_i_4_strip,alpha_A_4,aoaL_4_strip,AOAi_4_strip]')
   disp('|---------------|-------------------------------------------------------------|-------------------------------------------------------------
|')
   disp('|   Panel   Strip|                PT 5 (Re = 2.1960e6)                |                PT 5 (Re = 3.4914e6)                |')
   disp('|               | CL       CDp       CDi       AOA       aoaL       AOAi   | CL       CDp       CDi       AOA       aoaL
AOAi   |')
   disp('|---------------|-------------------------------------------------------------|-------------------------------------------------------------
|')
   fprintf('%8.0f%8.0f  %8.4f  %8.5f  %8.5f  %8.4f      %8.4f      %8.4f    %8.4f  %8.5f  %8.5f  %8.4f      %8.4f
%8.4f\n',[pan,strp,CL_5_strip,CD_v_5_strip,CDinf_i_5_strip,alpha_A_5,aoaL_5_strip,AOAi_5_strip,CL_6_strip,CD_
v_6_strip,CDinf_i_6_strip,alpha_A_6,aoaL_6_strip,AOAi_6_strip]')
   disp('|---------------|-------------------------------------------------------------|')
   disp('|   Panel   Strip|                PT 7 (Re = 5.5494e6)                |')
   disp('|               | CL       CDp       CDi       AOA       aoaL       AOAi   |')
   disp('|---------------|-------------------------------------------------------------|')
   fprintf('%8.0f%8.0f          %8.4f          %8.5f          %8.5f          %8.4f                                %8.4f
%8.4f\n',[pan,strp,CL_7_strip,CD_v_7_strip,CDinf_i_7_strip,alpha_A_7,aoaL_7_strip,AOAi_7_strip]')
   disp(' ')
   disp(' ')

%=========================================================================


% Experimental Plot - CD on X-Axis, CL on Y-Axis
plot(M_05(:,5),M_05(:,4),'b'),hold on,plot(M_05(:,7),M_05(:,6),'g'),hold on
plot(M_05(:,9),M_05(:,8),'r'),hold on,plot(M_05(:,11),M_05(:,10),'m'),hold on
plot(M_04(:,5),M_04(:,4),'b:'),hold on,plot(M_04(:,7),M_04(:,6),'g:'),hold on
plot(M_04(:,9),M_04(:,8),'r:'),hold on,plot(M_04(:,11),M_04(:,10),'m:'),hold on
axis([0 0.05 0 1.4]),grid
legend('Re = 5e5','Re = 1e6','Re = 2e6','Re = 1e7',4)
ylabel('2-D XFOIL Lift Coefficient (C_l)'),xlabel('2-D XFOIL Drag Coefficient (C_d)'),figure

% Experimental Plot - CD on X-Axis, CL on Y-Axis (Zoom In)
plot(M_05(:,5),M_05(:,4),'b'),hold on,plot(M_05(:,7),M_05(:,6),'g'),hold on
plot(M_05(:,9),M_05(:,8),'r'),hold on,plot(M_05(:,11),M_05(:,10),'m'),hold on
plot(M_04(:,5),M_04(:,4),'b:'),hold on,plot(M_04(:,7),M_04(:,6),'g:'),hold on
plot(M_04(:,9),M_04(:,8),'r:'),hold on,plot(M_04(:,11),M_04(:,10),'m:')
axis([0.004 0.015 0.0 1.2]),grid
legend('Re = 5e5','Re = 1e6','Re = 2e6','Re = 1e7',1)
ylabel('2-D XFOIL Lift Coefficient (C_l)'),xlabel('2-D XFOIL Drag Coefficient (C_d)'),figure

% Experimental Plot - CL on X-Axis, CD on Y-Axis
plot(M_05(:,4),M_05(:,5),'b'),hold on,plot(M_05(:,6),M_05(:,7),'g'),hold on
plot(M_05(:,8),M_05(:,9),'r'),hold on,plot(M_05(:,10),M_05(:,11),'m'),hold on
plot(M_04(:,4),M_04(:,5),'b:'),hold on,plot(M_04(:,6),M_04(:,7),'g:'),hold on
plot(M_04(:,8),M_04(:,9),'r:'),hold on,plot(M_04(:,10),M_04(:,11),'m:')
legend('Re = 5e5','Re = 1e6','Re = 2e6','Re = 1e7',2),grid
ylabel('2-D XFOIL Drag Coefficient ( C _d )'),xlabel('2-D XFOIL Lift Coefficient (C _l )'),figure

% Plot Of Point 1 Strip Panels On Experimental Drag Polar
plot(M_05(:,5),M_05(:,4),'b'),hold on,plot(M_05(:,7),M_05(:,6),'g'),hold on
plot(M_05(:,9),M_05(:,8),'r'),hold on,plot(M_05(:,11),M_05(:,10),'m'),hold on
plot(M_04(:,5),M_04(:,4),'b:'),hold on,plot(M_04(:,7),M_04(:,6),'g:'),hold on
plot(M_04(:,9),M_04(:,8),'r:'),hold on,plot(M_04(:,11),M_04(:,10),'m:')
plot(CD_v_1_strip,CL_1_strip,'k'),axis([0.004 0.015 0.0 1.2])
legend('Re = 5e5','Re = 1e6','Re = 2e6','Re = 1e7',1)
title('Point 1 - Begin Ingress')
ylabel('Lift Coefficient (CL)'),xlabel('Drag Coefficient (CD)'),figure
```

```
% Plot Of Point 2 Strip Panels On Experimental Drag Polar
plot(M_05(:,5),M_05(:,4),'b'),hold on,plot(M_05(:,7),M_05(:,6),'g'),hold on
plot(M_05(:,9),M_05(:,8),'r'),hold on,plot(M_05(:,11),M_05(:,10),'m'),hold on
plot(M_04(:,5),M_04(:,4),'b:'),hold on,plot(M_04(:,7),M_04(:,6),'g:'),hold on
plot(M_04(:,9),M_04(:,8),'r:'),hold on,plot(M_04(:,11),M_04(:,10),'m:')
plot(CD_v_2_strip,CL_2_strip,'k'),axis([0.004 0.015 0.0 1.2])
legend('Re = 5e5','Re = 1e6','Re = 2e6','Re = 1e7',1)
title('Point 2 - Middle Ingress')
ylabel('Lift Coefficient (CL)'),xlabel('Drag Coefficient (CD)'),figure

% Plot Of Point 3 Strip Panels On Experimental Drag Polar
plot(M_05(:,5),M_05(:,4),'b'),hold on,plot(M_05(:,7),M_05(:,6),'g'),hold on
plot(M_05(:,9),M_05(:,8),'r'),hold on,plot(M_05(:,11),M_05(:,10),'m'),hold on
plot(M_04(:,5),M_04(:,4),'b:'),hold on,plot(M_04(:,7),M_04(:,6),'g:'),hold on
plot(M_04(:,9),M_04(:,8),'r:'),hold on,plot(M_04(:,11),M_04(:,10),'m:')
plot(CD_v_3_strip,CL_3_strip,'k'),axis([0.004 0.015 0.0 1.2])
legend('Re = 5e5','Re = 1e6','Re = 2e6','Re = 1e7',1)
title('Point 3 - End Ingress/Begin Loiter')
ylabel('Lift Coefficient (CL)'),xlabel('Drag Coefficient (CD)'),figure

% Plot Of Point 4 Strip Panels On Experimental Drag Polar
plot(M_05(:,5),M_05(:,4),'b'),hold on,plot(M_05(:,7),M_05(:,6),'g'),hold on
plot(M_05(:,9),M_05(:,8),'r'),hold on,plot(M_05(:,11),M_05(:,10),'m'),hold on
plot(M_04(:,5),M_04(:,4),'b:'),hold on,plot(M_04(:,7),M_04(:,6),'g:'),hold on
plot(M_04(:,9),M_04(:,8),'r:'),hold on,plot(M_04(:,11),M_04(:,10),'m:')
plot(CD_v_4_strip,CL_4_strip,'k'),axis([0.004 0.015 0.0 1.2])
legend('Re = 5e5','Re = 1e6','Re = 2e6','Re = 1e7',1)
title('Point 4 - Middle Loiter')
ylabel('Lift Coefficient (CL)'),xlabel('Drag Coefficient (CD)'),figure

% Plot Of Point 5 Strip Panels On Experimental Drag Polar
plot(M_05(:,5),M_05(:,4),'b'),hold on,plot(M_05(:,7),M_05(:,6),'g'),hold on
plot(M_05(:,9),M_05(:,8),'r'),hold on,plot(M_05(:,11),M_05(:,10),'m'),hold on
plot(M_04(:,5),M_04(:,4),'b:'),hold on,plot(M_04(:,7),M_04(:,6),'g:'),hold on
plot(M_04(:,9),M_04(:,8),'r:'),hold on,plot(M_04(:,11),M_04(:,10),'m:')
plot(CD_v_5_strip,CL_5_strip,'k'),axis([0.004 0.015 0.0 1.2])
legend('Re = 5e5','Re = 1e6','Re = 2e6','Re = 1e7',1)
title('Point 5 - End Loiter/Begin Egress')
ylabel('Lift Coefficient (CL)'),xlabel('Drag Coefficient (CD)'),figure

% Plot Of Point 6 Strip Panels On Experimental Drag Polar
plot(M_05(:,5),M_05(:,4),'b'),hold on,plot(M_05(:,7),M_05(:,6),'g'),hold on
plot(M_05(:,9),M_05(:,8),'r'),hold on,plot(M_05(:,11),M_05(:,10),'m'),hold on
plot(M_04(:,5),M_04(:,4),'b:'),hold on,plot(M_04(:,7),M_04(:,6),'g:'),hold on
plot(M_04(:,9),M_04(:,8),'r:'),hold on,plot(M_04(:,11),M_04(:,10),'m:')
plot(CD_v_6_strip,CL_6_strip,'k'),axis([0.004 0.015 0.0 1.2])
legend('Re = 5e5','Re = 1e6','Re = 2e6','Re = 1e7',1)
title('Point 6 - Middle Egress')
ylabel('Lift Coefficient (CL)'),xlabel('Drag Coefficient (CD)'),figure

% Plot Of Point 7 Strip Panels On Experimental Drag Polar
plot(M_05(:,5),M_05(:,4),'b'),hold on,plot(M_05(:,7),M_05(:,6),'g'),hold on
plot(M_05(:,9),M_05(:,8),'r'),hold on,plot(M_05(:,11),M_05(:,10),'m'),hold on
plot(M_04(:,5),M_04(:,4),'b:'),hold on,plot(M_04(:,7),M_04(:,6),'g:'),hold on
plot(M_04(:,9),M_04(:,8),'r:'),hold on,plot(M_04(:,11),M_04(:,10),'m:')
plot(CD_v_7_strip,CL_7_strip,'k'),axis([0.004 0.015 0.0 1.2])
legend('Re = 5e5','Re = 1e6','Re = 2e6','Re = 1e7',1)
title('Point 7 - End Egress')
ylabel('Lift Coefficient (CL)'),xlabel('Drag Coefficient (CD)')
```

A.2 The Atmosphere Code

```
%=========================================================================

%                       ATMOSPHERE CODE

%=========================================================================

%       This code determines all atmospheric properties at any requested
%   altitude.

%=========================================================================


%=========================================================================
function [Tatm, patm, rhoatm,muatm] = HALE_Atmosphere(h)
%=========================================================================

   g=32.174;
   R=1716.0;

% Troposphere (gradient layer), h < 36,500 ft...
   a0=-0.003560;
   h0=0;
   T0=518.69;
   p0=2116.2;
   rho0=p0/( R*T0 );

   Tatm= T0 + a0*( h-h0 );
   patm= p0*(Tatm/T0)^( -g/(a0*R) );
   rhoatm= rho0*(Tatm/T0)^( -g/(a0*R)-1 );
   muatm = ((0.317*10^-10)*734.7*Tatm^1.5)/(Tatm+216);

% Stratosphere (isothermal layer), 36,500 < h < 82,500 ft...
   if h >= 36500
      h1=36500;
      T1=389.99;
      p1=464.86;
      rho1=p1/( R*T1 );

      Tatm= T1;
      patm= p1*exp( -g/(R*T1)*(h-h1) );
      rhoatm= rho1*exp( -g/(R*T1)*(h-h1) );
      muatm = ((0.317*10^-10)*734.7*Tatm^1.5)/(Tatm+216);
   end

% Mesosphere (gradient layer), 82,500 < h <155,000 ft...
   if h >= 82500
      a2=0.00162731;
      h2=82500;
      T2=390.24;
      p2=51.592;
      rho2=p2/( R*T2 );

      Tatm= T2 + a2*( h-h2 );
      patm= p2*(Tatm/T2)^( -g/(a2*R) );
      rhoatm= rho2*(Tatm/T2)^( -g/(a2*R)-1 );
      muatm = ((0.317*10^-10)*734.7*Tatm^1.5)/(Tatm+216);
   end
```

```
% Print out the data...
%    disp('h(feet)    T(deg.R)      p(psf)   rho(slug/ft3)')
%    fprintf('%7.0f  %10.2f  %10.4f      %10.8f\n', h, Tatm, patm, rhoatm)

return
```

A.3 The AVTIE Output Organizational Code

```
%=========================================================================

%                AVTIE OUTPUT INTERPOLATION CODE

%=========================================================================

%       This code organizes AVTIE outputs for easy calculation within
%   the MATLAB performance code.

%=========================================================================


%=========================================================================

%                COLUMNIZING DATA FROM AVTIE OUTPUTS

%=========================================================================

   if meth == 1;
      CL_A = av_out_begining(:,1);
      CD_t_A = av_out_begining(:,2);
      CD_v_A = av_out_begining(:,3);
      CD_i_A = av_out_begining(:,4);
      fuel_A = av_out_begining(:,5);
      alpha_A = av_out_begining(:,6);
      twist_A = av_out_begining(:,7);
      CD_i_pa = av_out_begining(:,8);
      CL_1_strip = PT_1_begining(:,3);
      A_1_strip = PT_1_begining(:,6);
      CD_i_1_strip = PT_1_begining(:,5);
      alpha_1_strip = PT_1_begining(:,7);
      CL_2_strip = PT_2_begining(:,3);
      A_2_strip = PT_2_begining(:,6);
      CD_i_2_strip = PT_2_begining(:,5);
      alpha_2_strip = PT_2_begining(:,7);
      CL_3_strip = PT_3_begining(:,3);
      A_3_strip = PT_3_begining(:,6);
      CD_i_3_strip = PT_3_begining(:,5);
      alpha_3_strip = PT_3_begining(:,7);
      CL_4_strip = PT_4_begining(:,3);
      A_4_strip = PT_4_begining(:,6);
      CD_i_4_strip = PT_4_begining(:,5);
      alpha_4_strip = PT_4_begining(:,7);
      CL_5_strip = PT_5_begining(:,3);
      A_5_strip = PT_5_begining(:,6);
      CD_i_5_strip = PT_5_begining(:,5);
      alpha_5_strip = PT_5_begining(:,7);
      CL_6_strip = PT_6_begining(:,3);
      A_6_strip = PT_6_begining(:,6);
      CD_i_6_strip = PT_6_begining(:,5);
```

```
alpha_6_strip = PT_6_begining(:,7);
CL_7_strip = PT_7_begining(:,3);
A_7_strip = PT_7_begining(:,6);
CD_i_7_strip = PT_7_begining(:,5);
alpha_7_strip = PT_7_begining(:,7);
pan = PT_1_begining(:,1);
strp = PT_1_begining(:,2);
else if meth == 2;
    CL_A = av_out_end(:,1);
    CD_t_A = av_out_end(:,2);
    CD_v_A = av_out_end(:,3);
    CD_i_A = av_out_end(:,4);
    fuel_A = av_out_end(:,5);
    alpha_A = av_out_end(:,6);
    twist_A = av_out_end(:,7);
    CD_i_pa = av_out_end(:,8);
    CL_1_strip = PT_1_end(:,3);
    A_1_strip = PT_1_end(:,6);
    CD_i_1_strip = PT_1_end(:,5);
    alpha_1_strip = PT_1_end(:,7);
    CL_2_strip = PT_2_end(:,3);
    A_2_strip = PT_2_end(:,6);
    CD_i_2_strip = PT_2_end(:,5);
    alpha_2_strip = PT_2_end(:,7);
    CL_3_strip = PT_3_end(:,3);
    A_3_strip = PT_3_end(:,6);
    CD_i_3_strip = PT_3_end(:,5);
    alpha_3_strip = PT_3_end(:,7);
    CL_4_strip = PT_4_end(:,3);
    A_4_strip = PT_4_end(:,6);
    CD_i_4_strip = PT_4_end(:,5);
    alpha_4_strip = PT_4_end(:,7);
    CL_5_strip = PT_5_end(:,3);
    A_5_strip = PT_5_end(:,6);
    CD_i_5_strip = PT_5_end(:,5);
    alpha_5_strip = PT_5_end(:,7);
    CL_6_strip = PT_6_end(:,3);
    A_6_strip = PT_6_end(:,6);
    CD_i_6_strip = PT_6_end(:,5);
    alpha_6_strip = PT_6_end(:,7);
    CL_7_strip = PT_7_end(:,3);
    A_7_strip = PT_7_end(:,6);
    CD_i_7_strip = PT_7_end(:,5);
    alpha_7_strip = PT_7_end(:,7);
    pan = PT_1_end(:,1);
    strp = PT_1_end(:,2);
else
    CL_A = av_out_avg(:,1);
    CD_t_A = av_out_avg(:,2);
    CD_v_A = av_out_avg(:,3);
    CD_i_A = av_out_avg(:,4);
    fuel_A = av_out_avg(:,5);
    alpha_A = av_out_avg(:,6);
    twist_A = av_out_avg(:,7);
    CD_i_pa = av_out_avg(:,8);
    CL_1_strip = PT_1_avg(:,3);
    A_1_strip = PT_1_avg(:,6);
    CD_i_1_strip = PT_1_avg(:,5);
    alpha_1_strip = PT_1_avg(:,7);
    CL_2_strip = PT_2_avg(:,3);
    A_2_strip = PT_2_avg(:,6);
    CD_i_2_strip = PT_2_avg(:,5);
    alpha_2_strip = PT_2_avg(:,7);
    CL_3_strip = PT_3_avg(:,3);
    A_3_strip = PT_3_avg(:,6);
    CD_i_3_strip = PT_3_avg(:,5);
    alpha_3_strip = PT_3_avg(:,7);
    CL_4_strip = PT_4_avg(:,3);
    A_4_strip = PT_4_avg(:,6);
```

```
        CD_i_4_strip = PT_4_avg(:,5);
        alpha_4_strip = PT_4_avg(:,7);
        CL_5_strip = PT_5_avg(:,3);
        A_5_strip = PT_5_avg(:,6);
        CD_i_5_strip = PT_5_avg(:,5);
        alpha_5_strip = PT_5_avg(:,7);
        CL_6_strip = PT_6_avg(:,3);
        A_6_strip = PT_6_avg(:,6);
        CD_i_6_strip = PT_6_avg(:,5);
        alpha_6_strip = PT_6_avg(:,7);
        CL_7_strip = PT_7_avg(:,3);
        A_7_strip = PT_7_avg(:,6);
        CD_i_7_strip = PT_7_avg(:,5);
        alpha_7_strip = PT_7_avg(:,7);
        pan = PT_1_avg(:,1);
        strp = PT_1_avg(:,2);
    end
end
alpha_A_1 = alpha_1_strip;
alpha_A_2 = alpha_2_strip;
alpha_A_3 = alpha_3_strip;
alpha_A_4 = alpha_4_strip;
alpha_A_5 = alpha_5_strip;
alpha_A_6 = alpha_6_strip;
alpha_A_7 = alpha_7_strip;


%=========================================================================

%                   Strip Locations

%=========================================================================


% Strip Locations (Meters from fuselage centerline)
strips = [...
        0.0000
        0.9877
        3.2841
        6.7681
        10.2522
        13.7363
        17.2203
        20.7044
        0.0000
        0.9877
        3.2841
        6.7681
        10.2522
        13.7363
        17.2203
        20.7044
        21.7291
        22.7556
        23.7820
        24.8085
        25.3975
        25.9363
        26.4750
        27.0137
        27.5525
        28.0912
        28.6300
        29.1687
        29.7074
        30.2462
        30.7849
```

```
        31.3236
        32.0721
        32.3216
        32.5711
        32.8205];


%=========================================================================

%               INTERPOLATIONS FROM AVTIE OUTPUTS

%=========================================================================


   % MATLAB INTERPOLATION FOR STRIP SECTIONS (ROSKAM/AVTIE STRIP METHOD)

   % Forcing All Lift Coefficients To Be Within Drag Polar Range
   for qq = 1:length(PT_1_begining);
      if CL_1_strip(qq) <= 0.05670;
         CL_1_strip(qq) = 0.05671;
      else
      end
      if CL_2_strip(qq) <= 0.05670;
         CL_2_strip(qq) = 0.05671;
      else
      end
      if CL_3_strip(qq) <= 0.05670;
         CL_3_strip(qq) = 0.05671;
      else
      end
      if CL_4_strip(qq) <= 0.05670;
         CL_4_strip(qq) = 0.05671;
      else
      end
      if CL_5_strip(qq) <= 0.05670;
         CL_5_strip(qq) = 0.05671;
      else
      end
      if CL_6_strip(qq) <= 0.05670;
         CL_6_strip(qq) = 0.05671;
      else
      end
      if CL_7_strip(qq) <= 0.05670;
         CL_7_strip(qq) = 0.05671;
      else
      end
   end

   % MatLab Viscous Drag Interpolation For Re = 5.0e5 Using Strip Lift Coefficient (Strip)
   RN_w = RNL*fiw_MAC;
   for p = 1:length(PT_1_begining);
      for s = 1:length(M_05)-1;
         if CL_1_strip(p) > M_05(s,4)
            CD_v_1_strip_Re5e5(p) = ((CL_1_strip(p)-M_05(s,4))/(M_05(s+1,4)-M_05(s,4)))*(M_05(s+1,5)-M_05(s,5))+M_05(s,5);
            aoaL_1_strip_Re5e5(p)          =          ((CD_v_1_strip_Re5e5(p)-M_05(s,5))/(M_05(s+1,5)-M_05(s,5)))*(M_05(s+1,1)-
M_05(s,1))+M_05(s,1);
         else
         end
         if CL_2_strip(p) > M_05(s,4)
            CD_v_2_strip_Re5e5(p) = ((CL_2_strip(p)-M_05(s,4))/(M_05(s+1,4)-M_05(s,4)))*(M_05(s+1,5)-M_05(s,5))+M_05(s,5);
            aoaL_2_strip_Re5e5(p)          =          ((CD_v_2_strip_Re5e5(p)-M_05(s,5))/(M_05(s+1,5)-M_05(s,5)))*(M_05(s+1,1)-
M_05(s,1))+M_05(s,1);
         else
         end
         if CL_3_strip(p) > M_05(s,4)
            CD_v_3_strip_Re5e5(p) = ((CL_3_strip(p)-M_05(s,4))/(M_05(s+1,4)-M_05(s,4)))*(M_05(s+1,5)-M_05(s,5))+M_05(s,5);
            aoaL_3_strip_Re5e5(p)          =          ((CD_v_3_strip_Re5e5(p)-M_05(s,5))/(M_05(s+1,5)-M_05(s,5)))*(M_05(s+1,1)-
M_05(s,1))+M_05(s,1);
```

```
        else
        end
        if CL_4_strip(p) > M_05(s,4)
           CD_v_4_strip_Re5e5(p) = ((CL_4_strip(p)-M_05(s,4))/(M_05(s+1,4)-M_05(s,4)))*(M_05(s+1,5)-M_05(s,5))+M_05(s,5);
           aoaL_4_strip_Re5e5(p)        =        ((CD_v_4_strip_Re5e5(p)-M_05(s,5))/(M_05(s+1,5)-M_05(s,5)))*(M_05(s+1,1)-
M_05(s,1))+M_05(s,1);
        else
        end
        if CL_5_strip(p) > M_05(s,4)
           CD_v_5_strip_Re5e5(p) = ((CL_5_strip(p)-M_05(s,4))/(M_05(s+1,4)-M_05(s,4)))*(M_05(s+1,5)-M_05(s,5))+M_05(s,5);
           aoaL_5_strip_Re5e5(p)        =        ((CD_v_5_strip_Re5e5(p)-M_05(s,5))/(M_05(s+1,5)-M_05(s,5)))*(M_05(s+1,1)-
M_05(s,1))+M_05(s,1);
        else
        end
        if CL_6_strip(p) > M_05(s,4)
           CD_v_6_strip_Re5e5(p) = ((CL_6_strip(p)-M_05(s,4))/(M_05(s+1,4)-M_05(s,4)))*(M_05(s+1,5)-M_05(s,5))+M_05(s,5);
           aoaL_6_strip_Re5e5(p)        =        ((CD_v_6_strip_Re5e5(p)-M_05(s,5))/(M_05(s+1,5)-M_05(s,5)))*(M_05(s+1,1)-
M_05(s,1))+M_05(s,1);
        else
        end
        if CL_7_strip(p) > M_05(s,4)
           CD_v_7_strip_Re5e5(p) = ((CL_7_strip(p)-M_05(s,4))/(M_05(s+1,4)-M_05(s,4)))*(M_05(s+1,5)-M_05(s,5))+M_05(s,5);
           aoaL_7_strip_Re5e5(p)        =        ((CD_v_7_strip_Re5e5(p)-M_05(s,5))/(M_05(s+1,5)-M_05(s,5)))*(M_05(s+1,1)-
M_05(s,1))+M_05(s,1);
        else
        end
     end
  end
  CD_v_1_strip_Re5e5 = CD_v_1_strip_Re5e5';
  CD_v_2_strip_Re5e5 = CD_v_2_strip_Re5e5';
  CD_v_3_strip_Re5e5 = CD_v_3_strip_Re5e5';
  CD_v_4_strip_Re5e5 = CD_v_4_strip_Re5e5';
  CD_v_5_strip_Re5e5 = CD_v_5_strip_Re5e5';
  CD_v_6_strip_Re5e5 = CD_v_6_strip_Re5e5';
  CD_v_7_strip_Re5e5 = CD_v_7_strip_Re5e5';
  aoaL_1_strip_Re5e5 = aoaL_1_strip_Re5e5';
  aoaL_2_strip_Re5e5 = aoaL_2_strip_Re5e5';
  aoaL_3_strip_Re5e5 = aoaL_3_strip_Re5e5';
  aoaL_4_strip_Re5e5 = aoaL_4_strip_Re5e5';
  aoaL_5_strip_Re5e5 = aoaL_5_strip_Re5e5';
  aoaL_6_strip_Re5e5 = aoaL_6_strip_Re5e5';
  aoaL_7_strip_Re5e5 = aoaL_7_strip_Re5e5';

  % MatLab Viscous Drag Interpolation For Re = 1.0e6 Using Strip Lift Coefficient (Strip)
  for p = 1:length(PT_1_begining);
     for s = 1:length(M_05)-1;
        if CL_1_strip(p) > M_05(s,6)
           CD_v_1_strip_Re1e6(p) = ((CL_1_strip(p)-M_05(s,6))/(M_05(s+1,6)-M_05(s,6)))*(M_05(s+1,7)-M_05(s,7))+M_05(s,7);
           aoaL_1_strip_Re1e6(p)        =        ((CD_v_1_strip_Re1e6(p)-M_05(s,7))/(M_05(s+1,7)-M_05(s,7)))*(M_05(s+1,1)-
M_05(s,1))+M_05(s,1);
        else
        end
        if CL_2_strip(p) > M_05(s,6)
           CD_v_2_strip_Re1e6(p) = ((CL_2_strip(p)-M_05(s,6))/(M_05(s+1,6)-M_05(s,6)))*(M_05(s+1,7)-M_05(s,7))+M_05(s,7);
           aoaL_2_strip_Re1e6(p)        =        ((CD_v_2_strip_Re1e6(p)-M_05(s,7))/(M_05(s+1,7)-M_05(s,7)))*(M_05(s+1,1)-
M_05(s,1))+M_05(s,1);
        else
        end
        if CL_3_strip(p) > M_05(s,6)
           CD_v_3_strip_Re1e6(p) = ((CL_3_strip(p)-M_05(s,6))/(M_05(s+1,6)-M_05(s,6)))*(M_05(s+1,7)-M_05(s,7))+M_05(s,7);
           aoaL_3_strip_Re1e6(p)        =        ((CD_v_3_strip_Re1e6(p)-M_05(s,7))/(M_05(s+1,7)-M_05(s,7)))*(M_05(s+1,1)-
M_05(s,1))+M_05(s,1);
        else
        end
        if CL_4_strip(p) > M_05(s,6)
           CD_v_4_strip_Re1e6(p) = ((CL_4_strip(p)-M_05(s,6))/(M_05(s+1,6)-M_05(s,6)))*(M_05(s+1,7)-M_05(s,7))+M_05(s,7);
           aoaL_4_strip_Re1e6(p)        =        ((CD_v_4_strip_Re1e6(p)-M_05(s,7))/(M_05(s+1,7)-M_05(s,7)))*(M_05(s+1,1)-
M_05(s,1))+M_05(s,1);
        else
```

```
        end
        if CL_5_strip(p) > M_05(s,6)
           CD_v_5_strip_Re1e6(p) = ((CL_5_strip(p)-M_05(s,6))/(M_05(s+1,6)-M_05(s,6)))*(M_05(s+1,7)-M_05(s,7))+M_05(s,7);
           aoaL_5_strip_Re1e6(p)        =        ((CD_v_5_strip_Re1e6(p)-M_05(s,7))/(M_05(s+1,7)-M_05(s,7)))*(M_05(s+1,1)-
M_05(s,1))+M_05(s,1);
        else
        end
        if CL_6_strip(p) > M_05(s,6)
           CD_v_6_strip_Re1e6(p) = ((CL_6_strip(p)-M_05(s,6))/(M_05(s+1,6)-M_05(s,6)))*(M_05(s+1,7)-M_05(s,7))+M_05(s,7);
           aoaL_6_strip_Re1e6(p)        =        ((CD_v_6_strip_Re1e6(p)-M_05(s,7))/(M_05(s+1,7)-M_05(s,7)))*(M_05(s+1,1)-
M_05(s,1))+M_05(s,1);
        else
        end
        if CL_7_strip(p) > M_05(s,6)
           CD_v_7_strip_Re1e6(p) = ((CL_7_strip(p)-M_05(s,6))/(M_05(s+1,6)-M_05(s,6)))*(M_05(s+1,7)-M_05(s,7))+M_05(s,7);
           aoaL_7_strip_Re1e6(p)        =        ((CD_v_7_strip_Re1e6(p)-M_05(s,7))/(M_05(s+1,7)-M_05(s,7)))*(M_05(s+1,1)-
M_05(s,1))+M_05(s,1);
        else
        end
     end
  end
  CD_v_1_strip_Re1e6 = CD_v_1_strip_Re1e6';
  CD_v_2_strip_Re1e6 = CD_v_2_strip_Re1e6';
  CD_v_3_strip_Re1e6 = CD_v_3_strip_Re1e6';
  CD_v_4_strip_Re1e6 = CD_v_4_strip_Re1e6';
  CD_v_5_strip_Re1e6 = CD_v_5_strip_Re1e6';
  CD_v_6_strip_Re1e6 = CD_v_6_strip_Re1e6';
  CD_v_7_strip_Re1e6 = CD_v_7_strip_Re1e6';
  aoaL_1_strip_Re1e6 = aoaL_1_strip_Re1e6';
  aoaL_2_strip_Re1e6 = aoaL_2_strip_Re1e6';
  aoaL_3_strip_Re1e6 = aoaL_3_strip_Re1e6';
  aoaL_4_strip_Re1e6 = aoaL_4_strip_Re1e6';
  aoaL_5_strip_Re1e6 = aoaL_5_strip_Re1e6';
  aoaL_6_strip_Re1e6 = aoaL_6_strip_Re1e6';
  aoaL_7_strip_Re1e6 = aoaL_7_strip_Re1e6';

  % MatLab Viscous Drag Interpolation For Re = 2.0e6 Using Strip Lift Coefficient (Strip)
  for p = 1:length(PT_1_begining);
     for s = 1:length(M_05)-1;
        if CL_1_strip(p) > M_05(s,8)
           CD_v_1_strip_Re2e6(p) = ((CL_1_strip(p)-M_05(s,8))/(M_05(s+1,8)-M_05(s,8)))*(M_05(s+1,9)-M_05(s,9))+M_05(s,9);
           aoaL_1_strip_Re2e6(p)        =        ((CD_v_1_strip_Re2e6(p)-M_05(s,9))/(M_05(s+1,9)-M_05(s,9)))*(M_05(s+1,1)-
M_05(s,1))+M_05(s,1);
        else
        end
        if CL_2_strip(p) > M_05(s,8)
           CD_v_2_strip_Re2e6(p) = ((CL_2_strip(p)-M_05(s,8))/(M_05(s+1,8)-M_05(s,8)))*(M_05(s+1,9)-M_05(s,9))+M_05(s,9);
           aoaL_2_strip_Re2e6(p)        =        ((CD_v_2_strip_Re2e6(p)-M_05(s,9))/(M_05(s+1,9)-M_05(s,9)))*(M_05(s+1,1)-
M_05(s,1))+M_05(s,1);
        else
        end
        if CL_3_strip(p) > M_05(s,8)
           CD_v_3_strip_Re2e6(p) = ((CL_3_strip(p)-M_05(s,8))/(M_05(s+1,8)-M_05(s,8)))*(M_05(s+1,9)-M_05(s,9))+M_05(s,9);
           aoaL_3_strip_Re2e6(p)        =        ((CD_v_3_strip_Re2e6(p)-M_05(s,9))/(M_05(s+1,9)-M_05(s,9)))*(M_05(s+1,1)-
M_05(s,1))+M_05(s,1);
        else
        end
        if CL_4_strip(p) > M_05(s,8)
           CD_v_4_strip_Re2e6(p) = ((CL_4_strip(p)-M_05(s,8))/(M_05(s+1,8)-M_05(s,8)))*(M_05(s+1,9)-M_05(s,9))+M_05(s,9);
           aoaL_4_strip_Re2e6(p)        =        ((CD_v_4_strip_Re2e6(p)-M_05(s,9))/(M_05(s+1,9)-M_05(s,9)))*(M_05(s+1,1)-
M_05(s,1))+M_05(s,1);
        else
        end
        if CL_5_strip(p) > M_05(s,8)
           CD_v_5_strip_Re2e6(p) = ((CL_5_strip(p)-M_05(s,8))/(M_05(s+1,8)-M_05(s,8)))*(M_05(s+1,9)-M_05(s,9))+M_05(s,9);
           aoaL_5_strip_Re2e6(p)        =        ((CD_v_5_strip_Re2e6(p)-M_05(s,9))/(M_05(s+1,9)-M_05(s,9)))*(M_05(s+1,1)-
M_05(s,1))+M_05(s,1);
        else
        end
```

```
        if CL_6_strip(p) > M_05(s,8)
           CD_v_6_strip_Re2e6(p) = ((CL_6_strip(p)-M_05(s,8))/(M_05(s+1,8)-M_05(s,8)))*(M_05(s+1,9)-M_05(s,9))+M_05(s,9);
           aoaL_6_strip_Re2e6(p)          =          ((CD_v_6_strip_Re2e6(p)-M_05(s,9))/(M_05(s+1,9)-M_05(s,9)))*(M_05(s+1,1)-
M_05(s,1))+M_05(s,1);
        else
        end
        if CL_7_strip(p) > M_05(s,8)
           CD_v_7_strip_Re2e6(p) = ((CL_7_strip(p)-M_05(s,8))/(M_05(s+1,8)-M_05(s,8)))*(M_05(s+1,9)-M_05(s,9))+M_05(s,9);
           aoaL_7_strip_Re2e6(p)          =          ((CD_v_7_strip_Re2e6(p)-M_05(s,9))/(M_05(s+1,9)-M_05(s,9)))*(M_05(s+1,1)-
M_05(s,1))+M_05(s,1);
        else
        end
     end
  end
  CD_v_1_strip_Re2e6 = CD_v_1_strip_Re2e6';
  CD_v_2_strip_Re2e6 = CD_v_2_strip_Re2e6';
  CD_v_3_strip_Re2e6 = CD_v_3_strip_Re2e6';
  CD_v_4_strip_Re2e6 = CD_v_4_strip_Re2e6';
  CD_v_5_strip_Re2e6 = CD_v_5_strip_Re2e6';
  CD_v_6_strip_Re2e6 = CD_v_6_strip_Re2e6';
  CD_v_7_strip_Re2e6 = CD_v_7_strip_Re2e6';
  aoaL_1_strip_Re2e6 = aoaL_1_strip_Re2e6';
  aoaL_2_strip_Re2e6 = aoaL_2_strip_Re2e6';
  aoaL_3_strip_Re2e6 = aoaL_3_strip_Re2e6';
  aoaL_4_strip_Re2e6 = aoaL_4_strip_Re2e6';
  aoaL_5_strip_Re2e6 = aoaL_5_strip_Re2e6';
  aoaL_6_strip_Re2e6 = aoaL_6_strip_Re2e6';
  aoaL_7_strip_Re2e6 = aoaL_7_strip_Re2e6';

  % MatLab Viscous Drag Interpolation For Re = 1.0e7 Using Strip Lift Coefficient (Strip)
  for p = 1:length(PT_1_begining);
     for s = 1:length(M_05)-1;
        if CL_1_strip(p) > M_05(s,10)
           CD_v_1_strip_Re1e7(p)          =          ((CL_1_strip(p)-M_05(s,10))/(M_05(s+1,10)-M_05(s,10)))*(M_05(s+1,11)-
M_05(s,11))+M_05(s,11);
           aoaL_1_strip_Re1e7(p)       =       ((CD_v_1_strip_Re1e7(p)-M_05(s,11))/(M_05(s+1,11)-M_05(s,11)))*(M_05(s+1,1)-
M_05(s,1))+M_05(s,1);
        else
        end
        if CL_2_strip(p) > M_05(s,10)
           CD_v_2_strip_Re1e7(p)          =          ((CL_2_strip(p)-M_05(s,10))/(M_05(s+1,10)-M_05(s,10)))*(M_05(s+1,11)-
M_05(s,11))+M_05(s,11);
           aoaL_2_strip_Re1e7(p)       =       ((CD_v_2_strip_Re1e7(p)-M_05(s,11))/(M_05(s+1,11)-M_05(s,11)))*(M_05(s+1,1)-
M_05(s,1))+M_05(s,1);
        else
        end
        if CL_3_strip(p) > M_05(s,10)
           CD_v_3_strip_Re1e7(p)          =          ((CL_3_strip(p)-M_05(s,10))/(M_05(s+1,10)-M_05(s,10)))*(M_05(s+1,11)-
M_05(s,11))+M_05(s,11);
           aoaL_3_strip_Re1e7(p)       =       ((CD_v_3_strip_Re1e7(p)-M_05(s,11))/(M_05(s+1,11)-M_05(s,11)))*(M_05(s+1,1)-
M_05(s,1))+M_05(s,1);
        else
        end
        if CL_4_strip(p) > M_05(s,10)
           CD_v_4_strip_Re1e7(p)          =          ((CL_4_strip(p)-M_05(s,10))/(M_05(s+1,10)-M_05(s,10)))*(M_05(s+1,11)-
M_05(s,11))+M_05(s,11);
           aoaL_4_strip_Re1e7(p)       =       ((CD_v_4_strip_Re1e7(p)-M_05(s,11))/(M_05(s+1,11)-M_05(s,11)))*(M_05(s+1,1)-
M_05(s,1))+M_05(s,1);
        else
        end
        if CL_5_strip(p) > M_05(s,10)
           CD_v_5_strip_Re1e7(p)          =          ((CL_5_strip(p)-M_05(s,10))/(M_05(s+1,10)-M_05(s,10)))*(M_05(s+1,11)-
M_05(s,11))+M_05(s,11);
           aoaL_5_strip_Re1e7(p)       =       ((CD_v_5_strip_Re1e7(p)-M_05(s,11))/(M_05(s+1,11)-M_05(s,11)))*(M_05(s+1,1)-
M_05(s,1))+M_05(s,1);
        else
        end
        if CL_6_strip(p) > M_05(s,10)
```

```
            CD_v_6_strip_Re1e7(p)            =            ((CL_6_strip(p)-M_05(s,10))/(M_05(s+1,10)-M_05(s,10)))*(M_05(s+1,11)-
M_05(s,11))+M_05(s,11);
            aoaL_6_strip_Re1e7(p)        =        ((CD_v_6_strip_Re1e7(p)-M_05(s,11))/(M_05(s+1,11)-M_05(s,11)))*(M_05(s+1,1)-
M_05(s,1))+M_05(s,1);
         else
         end
         if CL_7_strip(p) > M_05(s,10)
            CD_v_7_strip_Re1e7(p)            =            ((CL_7_strip(p)-M_05(s,10))/(M_05(s+1,10)-M_05(s,10)))*(M_05(s+1,11)-
M_05(s,11))+M_05(s,11);
            aoaL_7_strip_Re1e7(p)        =        ((CD_v_7_strip_Re1e7(p)-M_05(s,11))/(M_05(s+1,11)-M_05(s,11)))*(M_05(s+1,1)-
M_05(s,1))+M_05(s,1);
         else
         end
      end
   end
   CD_v_1_strip_Re1e7 = CD_v_1_strip_Re1e7';
   CD_v_2_strip_Re1e7 = CD_v_2_strip_Re1e7';
   CD_v_3_strip_Re1e7 = CD_v_3_strip_Re1e7';
   CD_v_4_strip_Re1e7 = CD_v_4_strip_Re1e7';
   CD_v_5_strip_Re1e7 = CD_v_5_strip_Re1e7';
   CD_v_6_strip_Re1e7 = CD_v_6_strip_Re1e7';
   CD_v_7_strip_Re1e7 = CD_v_7_strip_Re1e7';
   aoaL_1_strip_Re1e7 = aoaL_1_strip_Re1e7';
   aoaL_2_strip_Re1e7 = aoaL_2_strip_Re1e7';
   aoaL_3_strip_Re1e7 = aoaL_3_strip_Re1e7';
   aoaL_4_strip_Re1e7 = aoaL_4_strip_Re1e7';
   aoaL_5_strip_Re1e7 = aoaL_5_strip_Re1e7';
   aoaL_6_strip_Re1e7 = aoaL_6_strip_Re1e7';
   aoaL_7_strip_Re1e7 = aoaL_7_strip_Re1e7';

   % Arranging ALL AOA Values Into A Single Matrix
   AOA_1 = [aoaL_1_strip_Re5e5,aoaL_1_strip_Re1e6,aoaL_1_strip_Re2e6,aoaL_1_strip_Re1e7];
   AOA_2 = [aoaL_2_strip_Re5e5,aoaL_2_strip_Re1e6,aoaL_2_strip_Re2e6,aoaL_2_strip_Re1e7];
   AOA_3 = [aoaL_3_strip_Re5e5,aoaL_3_strip_Re1e6,aoaL_3_strip_Re2e6,aoaL_3_strip_Re1e7];
   AOA_4 = [aoaL_4_strip_Re5e5,aoaL_4_strip_Re1e6,aoaL_4_strip_Re2e6,aoaL_4_strip_Re1e7];
   AOA_5 = [aoaL_5_strip_Re5e5,aoaL_5_strip_Re1e6,aoaL_5_strip_Re2e6,aoaL_5_strip_Re1e7];
   AOA_6 = [aoaL_6_strip_Re5e5,aoaL_6_strip_Re1e6,aoaL_6_strip_Re2e6,aoaL_6_strip_Re1e7];
   AOA_7 = [aoaL_7_strip_Re5e5,aoaL_7_strip_Re1e6,aoaL_7_strip_Re2e6,aoaL_7_strip_Re1e7];

   % MatLab AOA Interpolation Between Renold's Numbers
   for ff = 1:length(PT_1_begining)
      for gg = 1:length(RN_w)
         for hh = 1:length(RN_list)
            if RN_w(gg) > RN_list(hh)
               aoaL_1_strip(ff)            =            ((RN_w(gg)-RN_list(hh))/(RN_list(hh+1)-RN_list(hh)))*(AOA_1(ff,hh+1)-
AOA_1(ff,hh))+AOA_1(ff,hh);
               aoaL_2_strip(ff)            =            ((RN_w(gg)-RN_list(hh))/(RN_list(hh+1)-RN_list(hh)))*(AOA_2(ff,hh+1)-
AOA_2(ff,hh))+AOA_2(ff,hh);
               aoaL_3_strip(ff)            =            ((RN_w(gg)-RN_list(hh))/(RN_list(hh+1)-RN_list(hh)))*(AOA_3(ff,hh+1)-
AOA_3(ff,hh))+AOA_3(ff,hh);
               aoaL_4_strip(ff)            =            ((RN_w(gg)-RN_list(hh))/(RN_list(hh+1)-RN_list(hh)))*(AOA_4(ff,hh+1)-
AOA_4(ff,hh))+AOA_4(ff,hh);
               aoaL_5_strip(ff)            =            ((RN_w(gg)-RN_list(hh))/(RN_list(hh+1)-RN_list(hh)))*(AOA_5(ff,hh+1)-
AOA_5(ff,hh))+AOA_5(ff,hh);
               aoaL_6_strip(ff)            =            ((RN_w(gg)-RN_list(hh))/(RN_list(hh+1)-RN_list(hh)))*(AOA_6(ff,hh+1)-
AOA_6(ff,hh))+AOA_6(ff,hh);
               aoaL_7_strip(ff)            =            ((RN_w(gg)-RN_list(hh))/(RN_list(hh+1)-RN_list(hh)))*(AOA_7(ff,hh+1)-
AOA_7(ff,hh))+AOA_7(ff,hh);
            else
            end
            if flt_prof == 1
               aoaL_5_strip(ff) = ((RN_w(5)-RN_list(2))/(RN_list(3)-RN_list(2)))*(AOA_5(ff,3)-AOA_5(ff,2))+AOA_5(ff,2);
            else
            end
         end
      end
   end
   aoaL_1_strip = aoaL_1_strip';
   aoaL_2_strip = aoaL_2_strip';
```

```
aoaL_3_strip = aoaL_3_strip';
aoaL_4_strip = aoaL_4_strip';
aoaL_5_strip = aoaL_5_strip';
aoaL_6_strip = aoaL_6_strip';
aoaL_7_strip = aoaL_7_strip';

% Backing Out AOAi For Each Strip
AOAi_1_strip = alpha_A_1-aoaL_1_strip;
AOAi_2_strip = alpha_A_2-aoaL_2_strip;
AOAi_3_strip = alpha_A_3-aoaL_3_strip;
AOAi_4_strip = alpha_A_4-aoaL_4_strip;
AOAi_5_strip = alpha_A_5-aoaL_5_strip;
AOAi_6_strip = alpha_A_6-aoaL_6_strip;
AOAi_7_strip = alpha_A_7-aoaL_7_strip;

% Arranging All Strip CD Values Into A Single Matrix
CDM_1 = [CD_v_1_strip_Re5e5,CD_v_1_strip_Re1e6,CD_v_1_strip_Re2e6,CD_v_1_strip_Re1e7];
CDM_2 = [CD_v_2_strip_Re5e5,CD_v_2_strip_Re1e6,CD_v_2_strip_Re2e6,CD_v_2_strip_Re1e7];
CDM_3 = [CD_v_3_strip_Re5e5,CD_v_3_strip_Re1e6,CD_v_3_strip_Re2e6,CD_v_3_strip_Re1e7];
CDM_4 = [CD_v_4_strip_Re5e5,CD_v_4_strip_Re1e6,CD_v_4_strip_Re2e6,CD_v_4_strip_Re1e7];
CDM_5 = [CD_v_5_strip_Re5e5,CD_v_5_strip_Re1e6,CD_v_5_strip_Re2e6,CD_v_5_strip_Re1e7];
CDM_6 = [CD_v_6_strip_Re5e5,CD_v_6_strip_Re1e6,CD_v_6_strip_Re2e6,CD_v_6_strip_Re1e7];
CDM_7 = [CD_v_7_strip_Re5e5,CD_v_7_strip_Re1e6,CD_v_7_strip_Re2e6,CD_v_7_strip_Re1e7];

% MatLab Drag Interpolation Between Renold's Numbers
for jj = 1:length(PT_1_begining);
    for nn = 1:length(RN_w);
        for mm = 1:length(RN_list);
            if RN_w(nn) > RN_list(mm)
                CD_v_1_strip(jj)        =       ((RN_w(nn)-RN_list(mm))/(RN_list(mm+1)-RN_list(mm)))*(CDM_1(jj,mm+1)-
CDM_1(jj,mm))+CDM_1(jj,mm);
                CD_v_2_strip(jj)        =       ((RN_w(nn)-RN_list(mm))/(RN_list(mm+1)-RN_list(mm)))*(CDM_2(jj,mm+1)-
CDM_2(jj,mm))+CDM_2(jj,mm);
                CD_v_3_strip(jj)        =       ((RN_w(nn)-RN_list(mm))/(RN_list(mm+1)-RN_list(mm)))*(CDM_3(jj,mm+1)-
CDM_3(jj,mm))+CDM_3(jj,mm);
                CD_v_4_strip(jj)        =       ((RN_w(nn)-RN_list(mm))/(RN_list(mm+1)-RN_list(mm)))*(CDM_4(jj,mm+1)-
CDM_4(jj,mm))+CDM_4(jj,mm);
                CD_v_5_strip(jj)        =       ((RN_w(nn)-RN_list(mm))/(RN_list(mm+1)-RN_list(mm)))*(CDM_5(jj,mm+1)-
CDM_5(jj,mm))+CDM_5(jj,mm);
                CD_v_6_strip(jj)        =       ((RN_w(nn)-RN_list(mm))/(RN_list(mm+1)-RN_list(mm)))*(CDM_6(jj,mm+1)-
CDM_6(jj,mm))+CDM_6(jj,mm);
                CD_v_7_strip(jj)        =       ((RN_w(nn)-RN_list(mm))/(RN_list(mm+1)-RN_list(mm)))*(CDM_7(jj,mm+1)-
CDM_7(jj,mm))+CDM_7(jj,mm);
            else
            end
            if flt_prof == 1
                CD_v_5_strip(jj) = ((RN_w(5)-RN_list(2))/(RN_list(3)-RN_list(2)))*(CDM_5(jj,3)-CDM_5(jj,2))+CDM_5(jj,2);
            else
            end
        end
    end
end
CD_v_1_strip = CD_v_1_strip';
CD_v_2_strip = CD_v_2_strip';
CD_v_3_strip = CD_v_3_strip';
CD_v_4_strip = CD_v_4_strip';
CD_v_5_strip = CD_v_5_strip';
CD_v_6_strip = CD_v_6_strip';
CD_v_7_strip = CD_v_7_strip';

% Rotating Freestream Component Of Local Lift Vector
AOAi_1_strip_rad = AOAi_1_strip*cf_a;
AOAi_2_strip_rad = AOAi_2_strip*cf_a;
AOAi_3_strip_rad = AOAi_3_strip*cf_a;
AOAi_4_strip_rad = AOAi_4_strip*cf_a;
AOAi_5_strip_rad = AOAi_5_strip*cf_a;
AOAi_6_strip_rad = AOAi_6_strip*cf_a;
AOAi_7_strip_rad = AOAi_7_strip*cf_a;
CDinf_i_1_strip = CL_1_strip.*sin(AOAi_1_strip_rad);
```

```
CDinf_i_2_strip = CL_2_strip.*sin(AOAi_2_strip_rad);
CDinf_i_3_strip = CL_3_strip.*sin(AOAi_3_strip_rad);
CDinf_i_4_strip = CL_4_strip.*sin(AOAi_4_strip_rad);
CDinf_i_5_strip = CL_5_strip.*sin(AOAi_5_strip_rad);
CDinf_i_6_strip = CL_6_strip.*sin(AOAi_6_strip_rad);
CDinf_i_7_strip = CL_7_strip.*sin(AOAi_7_strip_rad);

% Rotating Local Drag Component Back To Freestream Frame
CDinf_v_1_strip = CD_v_1_strip.*sin(AOAi_1_strip_rad);
CDinf_v_2_strip = CD_v_2_strip.*sin(AOAi_2_strip_rad);
CDinf_v_3_strip = CD_v_3_strip.*sin(AOAi_3_strip_rad);
CDinf_v_4_strip = CD_v_4_strip.*sin(AOAi_4_strip_rad);
CDinf_v_5_strip = CD_v_5_strip.*sin(AOAi_5_strip_rad);
CDinf_v_6_strip = CD_v_6_strip.*sin(AOAi_6_strip_rad);
CDinf_v_7_strip = CD_v_7_strip.*sin(AOAi_7_strip_rad);

% MatLab Total Lift And Drag Calculation (Strip)
for ii = 1:length(PT_1_begining);
   % Point 1
      CLinf_1_num(ii) = A_1_strip(ii)*CL_1_strip(ii);
      CDinf_v_1_num(ii) = A_1_strip(ii)*CDinf_v_1_strip(ii);
      CDinf_i_1_num(ii) = A_1_strip(ii)*CDinf_i_1_strip(ii);
      area_1_strip(ii) = A_1_strip(ii);
   % Point 2
      CLinf_2_num(ii) = A_2_strip(ii)*CL_2_strip(ii);
      CDinf_v_2_num(ii) = A_2_strip(ii)*CDinf_v_2_strip(ii);
      CDinf_i_2_num(ii) = A_2_strip(ii)*CDinf_i_2_strip(ii);
      area_2_strip(ii) = A_2_strip(ii);
   % Point 3
      CLinf_3_num(ii) = A_3_strip(ii)*CL_3_strip(ii);
      CDinf_v_3_num(ii) = A_3_strip(ii)*CDinf_v_3_strip(ii);
      CDinf_i_3_num(ii) = A_3_strip(ii)*CDinf_i_3_strip(ii);
      area_3_strip(ii) = A_3_strip(ii);
   % Point 4
      CLinf_4_num(ii) = A_4_strip(ii)*CL_4_strip(ii);
      CDinf_v_4_num(ii) = A_4_strip(ii)*CDinf_v_4_strip(ii);
      CDinf_i_4_num(ii) = A_4_strip(ii)*CDinf_i_4_strip(ii);
      area_4_strip(ii) = A_4_strip(ii);
   % Point 5
      CLinf_5_num(ii) = A_5_strip(ii)*CL_5_strip(ii);
      CDinf_v_5_num(ii) = A_5_strip(ii)*CDinf_v_5_strip(ii);
      CDinf_i_5_num(ii) = A_5_strip(ii)*CDinf_i_5_strip(ii);
      area_5_strip(ii) = A_5_strip(ii);
    % Point 6
      CLinf_6_num(ii) = A_6_strip(ii)*CL_6_strip(ii);
      CDinf_v_6_num(ii) = A_6_strip(ii)*CDinf_v_6_strip(ii);
      CDinf_i_6_num(ii) = A_6_strip(ii)*CDinf_i_6_strip(ii);
      area_6_strip(ii) = A_6_strip(ii);
   % Point 7
      CLinf_7_num(ii) = A_7_strip(ii)*CL_7_strip(ii);
      CDinf_v_7_num(ii) = A_7_strip(ii)*CDinf_v_7_strip(ii);
      CDinf_i_7_num(ii) = A_7_strip(ii)*CDinf_i_7_strip(ii);
      area_7_strip(ii) = A_7_strip(ii);
end
% Point 1
   CLinf_total_1_s = (sum(CLinf_1_num))/(sum(area_1_strip));
   CDinf_v_total_1_s = (sum(CDinf_v_1_num))/(sum(area_1_strip));
   CDinf_i_total_1_s = (sum(CDinf_i_1_num))/(sum(area_1_strip));
% Point 2
   CLinf_total_2_s = (sum(CLinf_2_num))/(sum(area_2_strip));
   CDinf_v_total_2_s = (sum(CDinf_v_2_num))/(sum(area_2_strip));
   CDinf_i_total_2_s = (sum(CDinf_i_2_num))/(sum(area_2_strip));
% Point 3
   CLinf_total_3_s = (sum(CLinf_3_num))/(sum(area_3_strip));
   CDinf_v_total_3_s = (sum(CDinf_v_3_num))/(sum(area_3_strip));
   CDinf_i_total_3_s = (sum(CDinf_i_3_num))/(sum(area_3_strip));
% Point 4
   CLinf_total_4_s = (sum(CLinf_4_num))/(sum(area_4_strip));
   CDinf_v_total_4_s = (sum(CDinf_v_4_num))/(sum(area_4_strip));
```

```
    CDinf_i_total_4_s = (sum(CDinf_i_4_num))/(sum(area_4_strip));
% Point 5
    CLinf_total_5_s = (sum(CLinf_5_num))/(sum(area_5_strip));
    CDinf_v_total_5_s = (sum(CDinf_v_5_num))/(sum(area_5_strip));
    CDinf_i_total_5_s = (sum(CDinf_i_5_num))/(sum(area_5_strip));
% Point 6
    CLinf_total_6_s = (sum(CLinf_6_num))/(sum(area_6_strip));
    CDinf_v_total_6_s = (sum(CDinf_v_6_num))/(sum(area_6_strip));
    CDinf_i_total_6_s = (sum(CDinf_i_6_num))/(sum(area_6_strip));
% Point 7
    CLinf_total_7_s = (sum(CLinf_7_num))/(sum(area_7_strip));
    CDinf_v_total_7_s = (sum(CDinf_v_7_num))/(sum(area_7_strip));
    CDinf_i_total_7_s = (sum(CDinf_i_7_num))/(sum(area_7_strip));

% Summing Up Lift And Drag Coefficients For Entire Aircraft
CLinf_total_s = [CLinf_total_1_s
                 CLinf_total_2_s
                 CLinf_total_3_s
                 CLinf_total_4_s
                 CLinf_total_5_s
                 CLinf_total_6_s
                 CLinf_total_7_s];
CDinf_v_total_s = [CDinf_v_total_1_s
                   CDinf_v_total_2_s
                   CDinf_v_total_3_s
                   CDinf_v_total_4_s
                   CDinf_v_total_5_s
                   CDinf_v_total_6_s
                   CDinf_v_total_7_s];
CDinf_i_total_s = [CDinf_i_total_1_s
                   CDinf_i_total_2_s
                   CDinf_i_total_3_s
                   CDinf_i_total_4_s
                   CDinf_i_total_5_s
                   CDinf_i_total_6_s
                   CDinf_i_total_7_s];
CDinf_total_s = [CDinf_v_total_1_s+CDinf_i_total_1_s
                 CDinf_v_total_2_s+CDinf_i_total_2_s
                 CDinf_v_total_3_s+CDinf_i_total_3_s
                 CDinf_v_total_4_s+CDinf_i_total_4_s
                 CDinf_v_total_5_s+CDinf_i_total_5_s
                 CDinf_v_total_6_s+CDinf_i_total_6_s
                 CDinf_v_total_7_s+CDinf_i_total_7_s];
L_D_total_s = CLinf_total_s./CDinf_total_s;


%=========================================================================

%                       TWIST TRIAL RUNS

%=========================================================================


    % Lift Coefficeint
    % By Strips
    CL_twst_1 = twst_run_1(:,3);
    CL_twst_2 = twst_run_2(:,3);
    CL_twst_3 = twst_run_3(:,3);
    CL_twst_4 = twst_run_4(:,3);
    % By Vehicle
    CL_twst_v_1 = 0.68013;
    CL_twst_v_2 = 0.68284;
    CL_twst_v_3 = 0.68367;
    CL_twst_v_4 = 0.68667;

    % Induced Drag
    CDi_twst_1 = twst_run_1_CDi;
    CDi_twst_2 = twst_run_2_CDi;
    CDi_twst_3 = twst_run_3_CDi;
```

```
CDi_twst_4 = twst_run_4_CDi;


% MatLab Viscous Drag Interpolation For Re = 2.0e6 Using Strip Lift Coefficient (Strip)
for p = 1:length(PT_1_begining);
   for s = 1:length(M_05)-1;
      if CL_twst_1(p) > M_05(s,8)
         CD_v_twst_1_Re2e6(p) = ((CL_twst_1(p)-M_05(s,8))/(M_05(s+1,8)-M_05(s,8)))*(M_05(s+1,9)-M_05(s,9))+M_05(s,9);
      else
      end
      if CL_twst_2(p) > M_05(s,8)
         CD_v_twst_2_Re2e6(p) = ((CL_twst_2(p)-M_05(s,8))/(M_05(s+1,8)-M_05(s,8)))*(M_05(s+1,9)-M_05(s,9))+M_05(s,9);
      else
      end
      if CL_twst_3(p) > M_05(s,8)
         CD_v_twst_3_Re2e6(p) = ((CL_twst_3(p)-M_05(s,8))/(M_05(s+1,8)-M_05(s,8)))*(M_05(s+1,9)-M_05(s,9))+M_05(s,9);
      else
      end
      if CL_twst_4(p) > M_05(s,8)
         CD_v_twst_4_Re2e6(p) = ((CL_twst_4(p)-M_05(s,8))/(M_05(s+1,8)-M_05(s,8)))*(M_05(s+1,9)-M_05(s,9))+M_05(s,9);
      else
      end
   end
end
CD_v_twst_1_Re2e6 = CD_v_twst_1_Re2e6';
CD_v_twst_2_Re2e6 = CD_v_twst_2_Re2e6';
CD_v_twst_3_Re2e6 = CD_v_twst_3_Re2e6';
CD_v_twst_4_Re2e6 = CD_v_twst_4_Re2e6';


% MatLab Viscous Drag Interpolation For Re = 1.0e7 Using Strip Lift Coefficient (Strip)
for p = 1:length(PT_1_begining);
   for s = 1:length(M_05)-1;
      if CL_twst_1(p) > M_05(s,10)
         CD_v_twst_1_Re1e7(p)          =          ((CL_twst_1(p)-M_05(s,10))/(M_05(s+1,10)-M_05(s,10)))*(M_05(s+1,11)-
M_05(s,11))+M_05(s,11);
      else
      end
      if CL_twst_2(p) > M_05(s,10)
         CD_v_twst_2_Re1e7(p)          =          ((CL_twst_2(p)-M_05(s,10))/(M_05(s+1,10)-M_05(s,10)))*(M_05(s+1,11)-
M_05(s,11))+M_05(s,11);
      else
      end
      if CL_twst_3(p) > M_05(s,10)
         CD_v_twst_3_Re1e7(p)          =          ((CL_twst_3(p)-M_05(s,10))/(M_05(s+1,10)-M_05(s,10)))*(M_05(s+1,11)-
M_05(s,11))+M_05(s,11);
      else
      end
      if CL_twst_4(p) > M_05(s,10)
         CD_v_twst_4_Re1e7(p)          =          ((CL_twst_4(p)-M_05(s,10))/(M_05(s+1,10)-M_05(s,10)))*(M_05(s+1,11)-
M_05(s,11))+M_05(s,11);
      else
      end
   end
end
CD_v_twst_1_Re1e7 = CD_v_twst_1_Re1e7';
CD_v_twst_2_Re1e7 = CD_v_twst_2_Re1e7';
CD_v_twst_3_Re1e7 = CD_v_twst_3_Re1e7';
CD_v_twst_4_Re1e7 = CD_v_twst_4_Re1e7';


% Arranging All Strip CD Values Into A Single Matrix
CDM_twst_1 = [CD_v_twst_1_Re2e6,CD_v_twst_1_Re1e7];
CDM_twst_2 = [CD_v_twst_2_Re2e6,CD_v_twst_2_Re1e7];
CDM_twst_3 = [CD_v_twst_3_Re2e6,CD_v_twst_3_Re1e7];
CDM_twst_4 = [CD_v_twst_4_Re2e6,CD_v_twst_4_Re1e7];
```

```
% MatLab Drag Interpolation Between Renold's Numbers
RN_twst = RN_w(4);
RN_H = RN_list(4);
RN_L = RN_list(3);
for jj = 1:length(PT_1_begining);
   CD_v_twst_1(jj) = ((RN_twst-RN_L)/(RN_H-RN_L))*(CDM_twst_1(jj,2)-CDM_twst_1(jj,1))+CDM_twst_1(jj,1);
   CD_v_twst_2(jj) = ((RN_twst-RN_L)/(RN_H-RN_L))*(CDM_twst_2(jj,2)-CDM_twst_2(jj,1))+CDM_twst_2(jj,1);
   CD_v_twst_3(jj) = ((RN_twst-RN_L)/(RN_H-RN_L))*(CDM_twst_3(jj,2)-CDM_twst_3(jj,1))+CDM_twst_3(jj,1);
   CD_v_twst_4(jj) = ((RN_twst-RN_L)/(RN_H-RN_L))*(CDM_twst_4(jj,2)-CDM_twst_4(jj,1))+CDM_twst_4(jj,1);
end
CD_v_twst_1 = CD_v_twst_1';
CD_v_twst_2 = CD_v_twst_2';
CD_v_twst_3 = CD_v_twst_3';
CD_v_twst_4 = CD_v_twst_4';


% Total Lift And Viscous Drag Calculations
for ii = 1:length(PT_1_begining);
   % Twist 1
      CLtwst_1(ii) = A_1_strip(ii)*CL_twst_1(ii);
      CDvtwst_1(ii) = A_1_strip(ii)*CD_v_twst_1(ii);
      area_1_strip(ii) = A_1_strip(ii);
   % Twist 2
      CLtwst_2(ii) = A_1_strip(ii)*CL_twst_2(ii);
      CDvtwst_2(ii) = A_1_strip(ii)*CD_v_twst_2(ii);
      area_1_strip(ii) = A_1_strip(ii);
   % Twist 3
      CLtwst_3(ii) = A_1_strip(ii)*CL_twst_3(ii);
      CDvtwst_3(ii) = A_1_strip(ii)*CD_v_twst_3(ii);
      area_1_strip(ii) = A_1_strip(ii);
   % Twist 4
      CLtwst_4(ii) = A_1_strip(ii)*CL_twst_4(ii);
      CDvtwst_4(ii) = A_1_strip(ii)*CD_v_twst_4(ii);
      area_1_strip(ii) = A_1_strip(ii);
end
   % Twist 1
      CLtwst_tot_1 = (sum(CLtwst_1))/(sum(area_1_strip));
      CDvtwst_tot_1 = (sum(CDvtwst_1))/(sum(area_1_strip));
   % Twist 2
      CLtwst_tot_2 = (sum(CLtwst_2))/(sum(area_1_strip));
      CDvtwst_tot_2 = (sum(CDvtwst_2))/(sum(area_1_strip));
   % Twist 3
      CLtwst_tot_3 = (sum(CLtwst_3))/(sum(area_1_strip));
      CDvtwst_tot_3 = (sum(CDvtwst_3))/(sum(area_1_strip));
   % Twist 4
      CLtwst_tot_4 = (sum(CLtwst_4))/(sum(area_1_strip));
      CDvtwst_tot_4 = (sum(CDvtwst_4))/(sum(area_1_strip));
```

A.4 XFOIL Generated Drag Polar Code

```
%=========================================================================

%          DRAG POLARS FOR THE LRN-1015 AT MACH 0.40 AND 0.50

%=========================================================================

%      This code is used to plot the XFOIL generated drag polars for the
%   LRN-1015 airfoil at Mach numbers of 0.40 and 0.50.

%=========================================================================

RN_list = [5.0e5    1.0e6   2.0e6   1.0e7];
```

% For Mach = 0.40
M_04 = [...
%RN-LIST 1.0e3 5.0e5 1.0e6 2.0e6 1.0e7
%alfa CL CD CL CD CL CD CL CD CL CD
-5.0 -0.19470 0.15389 0.03940 0.02552 0.04270 0.01675 0.04230 0.01164 0.03910 0.00625
-4.9 -0.19570 0.15328 0.04440 0.02395 0.05440 0.01637 0.05380 0.01119 0.05230 0.00618
-4.8 -0.19660 0.15268 0.05010 0.02294 0.06620 0.01596 0.06570 0.01087 0.06550 0.00611
-4.7 -0.19730 0.15209 0.06010 0.02217 0.07800 0.01553 0.07780 0.01060 0.07860 0.00604
-4.6 -0.19800 0.15152 0.07090 0.02159 0.08990 0.01511 0.09000 0.01035 0.09180 0.00598
-4.5 -0.19850 0.15096 0.08210 0.02115 0.10180 0.01469 0.10210 0.01011 0.10490 0.00591
-4.4 -0.19880 0.15041 0.09350 0.02097 0.11370 0.01428 0.11430 0.00988 0.11810 0.00584
-4.3 -0.19900 0.14988 0.10510 0.02083 0.12560 0.01389 0.12640 0.00965 0.13120 0.00578
-4.2 -0.19910 0.14936 0.11670 0.02062 0.13770 0.01357 0.13860 0.00943 0.14440 0.00572
-4.1 -0.19890 0.14885 0.12830 0.02027 0.15000 0.01337 0.15080 0.00922 0.15760 0.00566
-4.0 -0.19870 0.14836 0.13970 0.01975 0.16230 0.01319 0.16300 0.00902 0.17070 0.00561
-3.9 -0.19820 0.14788 0.15110 0.01915 0.17460 0.01299 0.17540 0.00887 0.18390 0.00556
-3.8 -0.19760 0.14742 0.16250 0.01861 0.18670 0.01274 0.18800 0.00876 0.19710 0.00551
-3.7 -0.19690 0.14697 0.17380 0.01813 0.19850 0.01241 0.20060 0.00864 0.21020 0.00546
-3.6 -0.19590 0.14653 0.18520 0.01769 0.21010 0.01208 0.21310 0.00851 0.22340 0.00542
-3.5 -0.19480 0.14612 0.19640 0.01731 0.22170 0.01179 0.22550 0.00835 0.23660 0.00538
-3.4 -0.19350 0.14571 0.20760 0.01696 0.23340 0.01156 0.23790 0.00820 0.24980 0.00535
-3.3 -0.19210 0.14533 0.21880 0.01666 0.24530 0.01140 0.25050 0.00809 0.25650 0.00534
-3.2 -0.19040 0.14495 0.23000 0.01643 0.25550 0.01170 0.26450 0.00832 0.27680 0.00543
-3.1 -0.18860 0.14459 0.24280 0.01681 0.27220 0.01199 0.27690 0.00816 0.28980 0.00535
-3.0 -0.18660 0.14425 0.25380 0.01652 0.28360 0.01182 0.28940 0.00802 0.30280 0.00528
-2.9 -0.18440 0.14392 0.26490 0.01627 0.29360 0.01149 0.30200 0.00791 0.31590 0.00523
-2.8 -0.18200 0.14360 0.27600 0.01605 0.30360 0.01096 0.31470 0.00783 0.32900 0.00519
-2.7 -0.17950 0.14330 0.28730 0.01588 0.31470 0.01058 0.32760 0.00779 0.34210 0.00516
-2.6 -0.17680 0.14301 0.29870 0.01576 0.32680 0.01040 0.34050 0.00778 0.35510 0.00511
-2.5 -0.17390 0.14274 0.31030 0.01569 0.33920 0.01030 0.35350 0.00777 0.36810 0.00508
-2.4 -0.17090 0.14248 0.32090 0.01539 0.35160 0.01021 0.36650 0.00776 0.38100 0.00506
-2.3 -0.16780 0.14223 0.33170 0.01512 0.36410 0.01013 0.37950 0.00776 0.39380 0.00505
-2.2 -0.16450 0.14200 0.34300 0.01495 0.37660 0.01005 0.39190 0.00763 0.40650 0.00504
-2.1 -0.16100 0.14178 0.35470 0.01481 0.38910 0.00998 0.40440 0.00752 0.41910 0.00505
-2.0 -0.15740 0.14157 0.36670 0.01469 0.40160 0.00991 0.41710 0.00745 0.43170 0.00505
-1.9 -0.15370 0.14138 0.37880 0.01458 0.41420 0.00984 0.42980 0.00739 0.44440 0.00505
-1.8 -0.14990 0.14119 0.39100 0.01447 0.42680 0.00978 0.44250 0.00734 0.45730 0.00504
-1.7 -0.14600 0.14103 0.40330 0.01438 0.43940 0.00971 0.45520 0.00729 0.47020 0.00503
-1.6 -0.14190 0.14087 0.41570 0.01430 0.45210 0.00965 0.46800 0.00725 0.48330 0.00502
-1.5 -0.13780 0.14073 0.42800 0.01423 0.46470 0.00960 0.48070 0.00722 0.49640 0.00500
-1.4 -0.13350 0.14060 0.45250 0.01410 0.47740 0.00955 0.49340 0.00719 0.50960 0.00498
-1.3 -0.12910 0.14048 0.45210 0.01394 0.49000 0.00950 0.50610 0.00716 0.52270 0.00496
-1.2 -0.12470 0.14038 0.46210 0.01250 0.50200 0.00931 0.51880 0.00713 0.53590 0.00495
-1.1 -0.12020 0.14029 0.47000 0.01200 0.51750 0.00861 0.54300 0.00688 0.54900 0.00493
-1.0 -0.11550 0.14021 0.47630 0.01183 0.52050 0.00802 0.55420 0.00661 0.56210 0.00492
-0.9 -0.11360 0.14076 0.48730 0.01185 0.53070 0.00761 0.56560 0.00637 0.57520 0.00490
-0.8 -0.10870 0.14067 0.49820 0.01190 0.54130 0.00737 0.57690 0.00613 0.58830 0.00488
-0.7 -0.10380 0.14059 0.51020 0.01199 0.55250 0.00728 0.58820 0.00590 0.60130 0.00486
-0.6 -0.09880 0.14052 0.52090 0.01203 0.56450 0.00726 0.59890 0.00570 0.61420 0.00484
-0.5 -0.09370 0.14046 0.53190 0.01211 0.57720 0.00728 0.61120 0.00550 0.62680 0.00482
-0.4 -0.08860 0.14042 0.54380 0.01220 0.58910 0.00728 0.62310 0.00533 0.63860 0.00481
-0.3 -0.08340 0.14039 0.55380 0.01224 0.60160 0.00730 0.63510 0.00520 0.65020 0.00481
-0.2 -0.07810 0.14037 0.56340 0.01234 0.61440 0.00734 0.64760 0.00513 0.66240 0.00480
-0.1 -0.07280 0.14036 0.57360 0.01249 0.62650 0.00734 0.66010 0.00509 0.67000 0.00475
0.00 -0.06750 0.14037 0.58260 0.01286 0.63850 0.00737 0.67300 0.00507 0.68740 0.00471
0.10 -0.06210 0.14038 0.59320 0.01306 0.65120 0.00739 0.68580 0.00505 0.70030 0.00468
0.20 -0.05660 0.14041 0.60440 0.01303 0.66400 0.00740 0.69880 0.00505 0.71300 0.00464
0.30 -0.05110 0.14044 0.61550 0.01304 0.67700 0.00740 0.71190 0.00506 0.72620 0.00462
0.40 -0.04560 0.14049 0.62660 0.01305 0.69000 0.00739 0.72500 0.00507 0.73920 0.00460
0.50 -0.04000 0.14055 0.63790 0.01306 0.70320 0.00736 0.73810 0.00508 0.75200 0.00458
0.60 -0.03440 0.14061 0.64930 0.01307 0.71630 0.00733 0.75130 0.00509 0.76460 0.00459

0.70	-0.02880 0.14069	0.66080 0.01307	0.72950 0.00730	0.76440 0.00511	0.77520 0.00469
0.80	-0.02310 0.14078	0.67250 0.01307	0.74260 0.00728	0.77750 0.00512	0.78560 0.00478
0.90	-0.01740 0.14088	0.68420 0.01307	0.75540 0.00724	0.79060 0.00513	0.79780 0.00478
1.00	-0.01170 0.14099	0.69600 0.01307	0.76790 0.00724	0.80350 0.00513	0.80000 0.00487
1.10	-0.00600 0.14111	0.70820 0.01307	0.78040 0.00727	0.81620 0.00513	0.81750 0.00496
1.20	-0.00020 0.14124	0.72070 0.01303	0.79270 0.00734	0.82890 0.00514	0.82950 0.00496
1.30	0.00550 0.14138	0.73320 0.01295	0.80510 0.00739	0.84140 0.00516	0.84040 0.00504
1.40	0.01130 0.14152	0.74520 0.01283	0.81780 0.00738	0.85390 0.00518	0.85040 0.00514
1.50	0.01710 0.14168	0.75750 0.01272	0.83050 0.00736	0.86640 0.00521	0.86200 0.00519
1.60	0.02290 0.14185	0.77020 0.01259	0.84330 0.00736	0.87900 0.00524	0.87440 0.00524
1.70	0.02870 0.14203	0.78310 0.01248	0.85600 0.00735	0.89150 0.00527	0.88600 0.00531
1.80	0.03460 0.14221	0.79620 0.01237	0.86880 0.00733	0.90410 0.00531	0.89560 0.00545
1.90	0.04040 0.14241	0.80950 0.01226	0.88160 0.00732	0.91700 0.00533	0.90660 0.00552
2.00	0.04630 0.14262	0.82290 0.01215	0.89440 0.00731	0.93020 0.00534	0.91870 0.00556
2.10	0.05210 0.14283	0.83630 0.01204	0.90730 0.00730	0.94340 0.00536	0.92850 0.00570
2.20	0.05800 0.14306	0.85000 0.01194	0.92020 0.00730	0.95660 0.00537	0.93740 0.00583
2.30	0.06380 0.14329	0.86390 0.01182	0.93310 0.00731	0.96970 0.00539	0.94850 0.00591
2.40	0.06970 0.14353	0.87780 0.01172	0.94600 0.00732	0.98270 0.00541	0.95610 0.00611
2.50	0.07550 0.14378	0.89160 0.01162	0.95890 0.00733	0.99560 0.00543	0.96710 0.00618
2.60	0.08140 0.14405	0.90580 0.01152	0.97180 0.00733	1.00820 0.00547	0.97650 0.00629
2.70	0.08720 0.14432	0.92510 0.01150	0.98460 0.00733	1.02040 0.00552	0.98600 0.00638
2.80	0.09310 0.14459	0.93210 0.01148	0.99740 0.00732	1.03220 0.00558	0.99780 0.00633
2.90	0.09890 0.14488	0.94310 0.01148	1.01000 0.00732	1.04380 0.00566	1.00950 0.00610
3.00	0.10480 0.14518	0.95390 0.01143	1.02210 0.00731	1.05560 0.00573	1.02150 0.00617
3.10	0.11060 0.14549	0.96490 0.01141	1.03410 0.00732	1.06850 0.00576	1.03190 0.00630
3.20	0.11640 0.14580	0.97600 0.01141	1.04600 0.00736	1.08130 0.00579	1.04200 0.00645
3.30	0.12230 0.14612	0.98740 0.01140	1.05830 0.00738	1.09370 0.00583	1.05320 0.00655
3.40	0.12810 0.14646	0.99900 0.01138	1.07090 0.00739	1.10540 0.00589	1.06520 0.00661
3.50	0.13390 0.14680	1.01090 0.01134	1.08350 0.00740	1.11570 0.00602	1.07620 0.00672
3.60	0.13970 0.14715	1.02310 0.01127	1.09610 0.00742	1.12550 0.00617	1.08620 0.00687
3.70	0.14550 0.14751	1.03570 0.01120	1.10860 0.00744	1.13690 0.00625	1.09640 0.00701
3.80	0.15130 0.14788	1.04850 0.01113	1.12100 0.00747	1.14480 0.00648	1.10750 0.00711
3.90	0.15700 0.14825	1.06140 0.01106	1.13310 0.00749	1.15220 0.00674	1.11860 0.00721
4.00	0.16280 0.14864	1.07410 0.01098	1.14480 0.00754	1.15780 0.00707	1.12690 0.00743
4.10	0.16850 0.14903	1.08690 0.01086	1.15570 0.00760	1.16480 0.00735	1.13640 0.00760
4.20	0.17420 0.14944	1.09970 0.01074	1.16600 0.00769	1.16990 0.00770	1.14780 0.00769
4.30	0.17990 0.14985	1.11220 0.01065	1.17820 0.00773	1.17740 0.00796	1.15960 0.00776
4.40	0.18560 0.15027	1.12410 0.01062	1.18990 0.00779	1.18390 0.00824	1.17080 0.00785
4.50	0.19130 0.15070	1.13360 0.01065	1.19990 0.00791	1.18890 0.00859	1.18100 0.00799
4.60	0.19700 0.15113	1.14350 0.01069	1.20610 0.00814	1.19760 0.00877	1.19050 0.00815
4.70	0.20260 0.15158	1.15390 0.01072	1.21580 0.00828	1.19000 0.00960	1.18800 0.00860
4.80	0.20820 0.15203	1.16480 0.01074	1.21320 0.00886	1.18080 0.01031	1.18600 0.00926
4.90	0.21380 0.15250	1.17560 0.01073	1.20640 0.00957	1.17270 0.01119	1.18030 0.00990
5.00	0.21940 0.15297	1.18620 0.01070	1.19800 0.01031	1.16770 0.01208	1.17900 0.01000
5.10	0.22500 0.15345	1.19680 0.01066	1.18880 0.01103	1.16810 0.01296	1.17660 0.01110
5.20	0.23050 0.15394	1.20670 0.01065	1.18180 0.01189	1.16760 0.01388	1.17500 0.01200
5.30	0.23610 0.15444	1.21580 0.01072	1.17530 0.01301	1.16980 0.01456	1.17390 0.01295
5.40	0.24160 0.15495	1.22730 0.01098	1.16870 0.01430	1.16930 0.01538	1.18320 0.01321
5.50	0.24700 0.15546	1.23060 0.01132	1.16600 0.01527	1.17360 0.01589	1.19240 0.01348
5.60	0.25250 0.15599	1.22750 0.01187	1.16190 0.01628	1.17460 0.01657	1.20130 0.01379
5.70	0.25790 0.15652	1.22190 0.01269	1.16350 0.01700	1.17660 0.01718	1.21010 0.01406
5.80	0.26340 0.15706	1.21780 0.01369	1.16560 0.01750	1.17920 0.01762	1.21900 0.01432
5.90	0.26870 0.15761	1.21060 0.01501	1.16530 0.01821	1.18120 0.01839	1.22790 0.01458
6.00	0.27410 0.15817	1.20740 0.01603	1.16860 0.01873	1.18300 0.01904	1.23680 0.01483
6.10	0.27940 0.15874	1.20060 0.01727	1.17140 0.01927	1.18550 0.01953	1.24550 0.01510
6.20	0.28480 0.15931	1.19900 0.01816	1.17270 0.01989	1.18790 0.02018	1.25300 0.01548
6.30	0.29010 0.15990	1.19370 0.01926	1.17500 0.02044	1.19020 0.02074	1.26100 0.01582
6.40	0.29530 0.16049	1.19210 0.02014	1.17800 0.02085	1.19480 0.02120	1.26920 0.01611
6.50	0.30060 0.16109	1.18980 0.02103	1.18370 0.02123	1.19770 0.02175	1.27760 0.01639
6.60	0.30580 0.16170	1.18700 0.02195	1.18630 0.02169	1.20160 0.02225	1.28600 0.01667
6.70	0.31100 0.16232	1.18840 0.02264	1.18910 0.02262	1.20800 0.02263	1.29430 0.01695
6.80	0.31610 0.16295	1.18710 0.02349	1.19460 0.02313	1.20700 0.02300	1.30260 0.01722

```
6.90    0.32130 0.16359    1.18520 0.02441    1.19990 0.02364    1.21500 0.02350    1.31080 0.01750
7.00    0.32640 0.16423    1.18830 0.02503    1.20490 0.02415    1.22140 0.02414    1.31890 0.01779
7.10    0.33150 0.16489    1.19010 0.02571    1.20900 0.02471    1.22390 0.02474    1.32690 0.01808
7.20    0.33650 0.16555    1.19090 0.02644    1.21250 0.02529    1.22830 0.02523    1.33410 0.01840
7.30    0.34150 0.16623    1.19390 0.02704    1.21750 0.02577    1.23480 0.02560    1.33780 0.01898
7.40    0.34650 0.16691    1.19950 0.02749    1.22370 0.02620    1.24190 0.02593    1.34410 0.01936
7.50    0.35150 0.16760    1.20430 0.02797    1.22890 0.02667    1.23760 0.02689    1.35120 0.01970
7.60    0.35650 0.16830    1.20800 0.02852    1.23270 0.02721    1.23370 0.02780    1.35790 0.02008
7.70    0.36140 0.16900    1.21060 0.02915    1.23560 0.02779    1.24000 0.02830    1.36460 0.02045
7.80    0.36630 0.16972    1.21320 0.02980    1.23990 0.02830    1.24500 0.02870    1.37120 0.02083
7.90    0.37110 0.17045    1.21810 0.03033    1.24580 0.02875    1.24970 0.02907    1.37790 0.02120
8.00    0.37600 0.17118    1.22300 0.03089    1.25130 0.02923    1.25600 0.02946    1.38460 0.02158
8.10    0.38080 0.17192    1.22760 0.03148    1.25590 0.02978    1.26230 0.02985    1.39130 0.02195
8.20    0.38560 0.17268    1.23170 0.03210    1.27970 0.02919    1.28000 0.02900    1.39790 0.02233
8.30    0.39030 0.17344    1.23520 0.03275    1.28330 0.02981    1.31800 0.02801    1.40460 0.02270
8.40    0.39500 0.17421    1.25160 0.03257    1.28550 0.03057    1.32000 0.02850    1.41130 0.02308
8.50    0.39970 0.17499    1.25630 0.03314    1.28940 0.03121    1.32300 0.02900    1.41800 0.02345
8.60    0.40440 0.17577    1.25990 0.03379    1.29480 0.03174    1.32600 0.02950    1.42460 0.02383
8.70    0.40900 0.17657    1.26270 0.03450    1.29970 0.03229    1.32880 0.03037    1.43130 0.02420
8.80    0.41360 0.17737    1.26560 0.03521    1.30380 0.03291    1.32400 0.03100    1.43800 0.02458
8.90    0.41820 0.17819    1.27130 0.03573    1.30680 0.03359    1.32000 0.03200    1.44470 0.02495
9.00    0.42270 0.17901    1.27660 0.03628    1.31050 0.03423    1.31620 0.03388    1.45130 0.02533
9.10    0.42720 0.17984    1.28100 0.03689    1.31610 0.03475    1.32240 0.03437    1.45800 0.02570
9.20    0.43170 0.18068    1.28400 0.03760    1.32140 0.03529    1.32870 0.03485    1.46500 0.02606
9.30    0.43620 0.18153    1.28620 0.03837    1.32620 0.03587    1.33540 0.03531    1.47190 0.02641
9.40    0.44070 0.18241    1.28790 0.03872    1.32825 0.03619    1.33885 0.03553    1.47530 0.02659
9.50    0.44520 0.18329    1.28960 0.03907    1.33030 0.03650    1.34230 0.03575    1.47870 0.02676
9.60    0.44960 0.18418    1.29450 0.03967    1.33390 0.03716    1.34880 0.03622    1.48570 0.02712
9.70    0.45400 0.18507    1.29830 0.04038    1.33810 0.03779    1.35520 0.03671    1.49260 0.02748
9.80    0.45840 0.18598    1.30180 0.04111    1.34350 0.03835    1.36150 0.03720    1.49960 0.02786
9.90    0.46280 0.18690    1.30510 0.04185    1.34920 0.03889    1.36760 0.03770    1.50670 0.02824
10.0    0.46710 0.18782    1.30790 0.04260    1.35360 0.03957    1.37360 0.03821    1.51370 0.02863];

% For Mach = 0.50
M_05 = [...
%RN-LIST     1.0e3          5.0e5          1.0e6          2.0e6          1.0e7
%alfa     CL     CD        CL     CD        CL     CD        CL     CD        CL     CD
-5.0    -0.19060 0.16284    0.05000 0.02900    0.05670 0.02156    0.05000 0.01500    0.05020 0.00869
-4.9    -0.19190 0.16216    0.05500 0.02800    0.06700 0.02075    0.06000 0.01450    0.06370 0.00842
-4.8    -0.19320 0.16148    0.06000 0.02700    0.07770 0.01998    0.07680 0.01392    0.07730 0.00817
-4.7    -0.19440 0.16081    0.06840 0.02646    0.08880 0.01924    0.08920 0.01347    0.09090 0.00794
-4.6    -0.19550 0.16014    0.07740 0.02573    0.10020 0.01852    0.10200 0.01309    0.10450 0.00772
-4.5    -0.19650 0.15949    0.08710 0.02510    0.11180 0.01784    0.11490 0.01273    0.11810 0.00751
-4.4    -0.19740 0.15885    0.09700 0.02447    0.12370 0.01720    0.12780 0.01236    0.13180 0.00731
-4.3    -0.19820 0.15821    0.10730 0.02383    0.13570 0.01660    0.14070 0.01198    0.14530 0.00711
-4.2    -0.19890 0.15759    0.11810 0.02345    0.14780 0.01605    0.15370 0.01159    0.15890 0.00691
-4.1    -0.19950 0.15698    0.12940 0.02316    0.16010 0.01555    0.16650 0.01121    0.17240 0.00672
-4.0    -0.19990 0.15639    0.14090 0.02282    0.17250 0.01518    0.17940 0.01084    0.18590 0.00653
-3.9    -0.20010 0.15580    0.15230 0.02233    0.18510 0.01498    0.19220 0.01048    0.19940 0.00634
-3.8    -0.20030 0.15523    0.16330 0.02167    0.19750 0.01485    0.20500 0.01016    0.21290 0.00616
-3.7    -0.20020 0.15468    0.17450 0.02103    0.20930 0.01471    0.21810 0.00994    0.22650 0.00600
-3.6    -0.20000 0.15413    0.18600 0.02045    0.22200 0.01429    0.23130 0.00973    0.24020 0.00586
-3.5    -0.19960 0.15361    0.19760 0.01991    0.23440 0.01380    0.24430 0.00950    0.25410 0.00574
-3.4    -0.19910 0.15310    0.20930 0.01943    0.24690 0.01338    0.25690 0.00922    0.26850 0.00571
-3.3    -0.19840 0.15260    0.22100 0.01900    0.26120 0.01369    0.27300 0.00962    0.27590 0.00575
-3.2    -0.19750 0.15212    0.23280 0.01863    0.27420 0.01341    0.28550 0.00929    0.29920 0.00604
-3.1    -0.19650 0.15165    0.24600 0.01912    0.28700 0.01312    0.29810 0.00901    0.31220 0.00578
-3.0    -0.19520 0.15120    0.25780 0.01870    0.29950 0.01279    0.31090 0.00875    0.32550 0.00560
-2.9    -0.19380 0.15077    0.26960 0.01831    0.31180 0.01242    0.32380 0.00853    0.33920 0.00547
-2.8    -0.19220 0.15035    0.28140 0.01797    0.32360 0.01202    0.33680 0.00835    0.35300 0.00538
-2.7    -0.19030 0.14995    0.29320 0.01766    0.33520 0.01161    0.35010 0.00819    0.36690 0.00532
-2.6    -0.18830 0.14956    0.30500 0.01739    0.34680 0.01124    0.36360 0.00808    0.38080 0.00527
```

-2.5	-0.18610 0.14919	0.31690 0.01716	0.35900 0.01097	0.37720 0.00801	0.39470 0.00523
-2.4	-0.18380 0.14884	0.32920 0.01696	0.37180 0.01081	0.39090 0.00797	0.40850 0.00519
-2.3	-0.18120 0.14850	0.34180 0.01680	0.38490 0.01070	0.40480 0.00795	0.42240 0.00516
-2.2	-0.17840 0.14817	0.35360 0.01644	0.39800 0.01061	0.41860 0.00794	0.43620 0.00515
-2.1	-0.17550 0.14787	0.36550 0.01607	0.41110 0.01052	0.43230 0.00793	0.45010 0.00513
-2.0	-0.17240 0.14757	0.37820 0.01582	0.42440 0.01044	0.44540 0.00779	0.46390 0.00512
-1.9	-0.16920 0.14730	0.39130 0.01563	0.43760 0.01037	0.45870 0.00769	0.47770 0.00511
-1.8	-0.16580 0.14703	0.40440 0.01548	0.45100 0.01029	0.47210 0.00762	0.49160 0.00510
-1.7	-0.16220 0.14679	0.41750 0.01537	0.46440 0.01022	0.48550 0.00757	0.50540 0.00509
-1.6	-0.15850 0.14656	0.43050 0.01528	0.47780 0.01015	0.49900 0.00754	0.51950 0.00508
-1.5	-0.15460 0.14634	0.44360 0.01521	0.49130 0.01009	0.51260 0.00751	0.53370 0.00506
-1.4	-0.15060 0.14614	0.45680 0.01515	0.50480 0.01003	0.52620 0.00750	0.54790 0.00504
-1.3	-0.14650 0.14596	0.46990 0.01511	0.51820 0.00999	0.53970 0.00748	0.56210 0.00502
-1.2	-0.14220 0.14579	0.48250 0.01507	0.53170 0.00996	0.55330 0.007479	0.57630 0.00500
-1.1	-0.13790 0.14564	0.49550 0.01502	0.54510 0.00991	0.56680 0.00748	0.59050 0.00499
-1.0	-0.13690 0.14643	0.50400 0.01419	0.55780 0.00971	0.58040 0.00747	0.60470 0.00497
-0.9	-0.13230 0.14625	0.50920 0.01362	0.56730 0.00897	0.59410 0.00740	0.61880 0.00496
-0.8	-0.12760 0.14608	0.51620 0.01354	0.57750 0.00836	0.60660 0.00712	0.63300 0.00494
-0.7	-0.12270 0.14594	0.52600 0.01356	0.58850 0.00797	0.61880 0.00681	0.64710 0.00493
-0.6	-0.11780 0.14580	0.53680 0.01364	0.60000 0.00777	0.62410 0.00650	0.66120 0.00491
-0.5	-0.11270 0.14568	0.54820 0.01374	0.61270 0.00773	0.64320 0.00619	0.67510 0.00489
-0.4	-0.10760 0.14558	0.55890 0.01380	0.62600 0.007729	0.65500 0.00600	0.68870 0.00486
-0.3	-0.10230 0.14549	0.56920 0.01390	0.63930 0.00773	0.66800 0.00568	0.70140 0.00485
-0.2	-0.09700 0.14542	0.58020 0.01403	0.65250 0.00775	0.68080 0.00552	0.71370 0.00486
-0.1	-0.09160 0.14536	0.58970 0.01416	0.66000 0.00778	0.69370 0.00540	0.72650 0.00485
0.00	-0.08610 0.14531	0.59760 0.01432	0.67980 0.00780	0.70720 0.00535	0.73970 0.00481
0.10	-0.08060 0.14528	0.60660 0.01449	0.69240 0.00782	0.72100 0.00533	0.75350 0.00476
0.20	-0.07500 0.14526	0.61710 0.01466	0.70600 0.00784	0.73490 0.00531	0.76740 0.00470
0.30	-0.06940 0.14525	0.62890 0.01475	0.71970 0.00785	0.74880 0.00530	0.78150 0.00468
0.40	-0.06370 0.14526	0.63970 0.01477	0.73360 0.007851	0.76280 0.00529	0.79560 0.00467
0.50	-0.05790 0.14528	0.65090 0.01479	0.74750 0.00785	0.77690 0.00530	0.80950 0.00463
0.60	-0.05210 0.14532	0.66250 0.014791	0.76150 0.00784	0.79100 0.00532	0.82360 0.004631
0.70	-0.04630 0.14537	0.67430 0.01479	0.77530 0.00787	0.80510 0.00533	0.83720 0.00463
0.80	-0.04040 0.14542	0.68650 0.014789	0.78850 0.00796	0.81920 0.00535	0.85500 0.00470
0.90	-0.03450 0.14550	0.69920 0.01477	0.80180 0.00803	0.83320 0.00537	0.86010 0.00476
1.00	-0.02860 0.14558	0.71260 0.01472	0.81590 0.00800	0.84720 0.00539	0.87260 0.004761
1.10	-0.02260 0.14568	0.72670 0.01462	0.82980 0.00796	0.86100 0.00541	0.88000 0.00485
1.20	-0.01670 0.14579	0.74170 0.01450	0.84340 0.00792	0.87460 0.00543	0.89340 0.00492
1.30	-0.01060 0.14591	0.75730 0.01435	0.85710 0.00791	0.88780 0.00544	0.90550 0.00499
1.40	-0.00460 0.14604	0.77320 0.01420	0.87060 0.007891	0.90100 0.00547	0.91600 0.00511
1.50	0.00140 0.14618	0.78850 0.01398	0.88430 0.00789	0.91430 0.00551	0.92550 0.00526
1.60	0.00750 0.14633	0.80370 0.01377	0.89790 0.007889	0.92810 0.00554	0.93850 0.00531
1.70	0.01360 0.14650	0.81890 0.01358	0.91160 0.00788	0.94220 0.00555	0.95090 0.00538
1.80	0.01970 0.14668	0.83390 0.01342	0.92550 0.00787	0.95640 0.00557	0.96030 0.00556
1.90	0.02570 0.14687	0.84850 0.01329	0.93940 0.00785	0.97060 0.00558	0.97170 0.00563
2.00	0.03190 0.14707	0.86280 0.01320	0.95330 0.00783	0.98480 0.00560	0.98440 0.00569
2.10	0.03800 0.14728	0.87670 0.01317	0.96730 0.00782	0.99900 0.00562	0.99520 0.00581
2.20	0.04410 0.14750	0.88920 0.01330	0.98120 0.00781	1.01320 0.00563	1.00490 0.00595
2.30	0.05020 0.14773	0.90040 0.01331	0.99520 0.007809	1.02720 0.00566	1.01720 0.00602
2.40	0.05630 0.14797	0.91160 0.01333	1.00900 0.00781	1.04110 0.00568	1.02890 0.00611
2.50	0.06240 0.14822	0.92280 0.01336	1.02270 0.00782	1.05480 0.00572	1.03650 0.00634
2.60	0.06860 0.14849	0.93440 0.01337	1.03630 0.00783	1.06800 0.00577	1.04770 0.00644
2.70	0.07470 0.14876	0.94670 0.01336	1.04950 0.00785	1.08030 0.00584	1.05610 0.00661
2.80	0.08080 0.14905	0.95920 0.01331	1.06250 0.00789	1.09210 0.00593	1.06520 0.00674
2.90	0.08690 0.14934	0.97130 0.01322	1.07600 0.00793	1.10550 0.00599	1.07700 0.00673
3.00	0.09300 0.14964	0.98420 0.01312	1.08970 0.00795	1.11940 0.00603	1.09020 0.00660
3.10	0.09910 0.14996	0.99750 0.01301	1.10330 0.00796	1.13290 0.00608	1.09900 0.00682
3.20	0.10520 0.15029	1.01100 0.01293	1.11690 0.007961	1.14560 0.00616	1.10910 0.00700
3.30	0.11130 0.15062	1.02500 0.01283	1.13030 0.00797	1.15660 0.00629	1.12170 0.00707
3.40	0.11740 0.15097	1.03930 0.01272	1.14350 0.00798	1.16640 0.00648	1.13410 0.00716
3.50	0.12340 0.15132	1.05400 0.01257	1.15640 0.00800	1.17890 0.00656	1.14450 0.00731
3.60	0.12950 0.15169	1.06980 0.01237	1.16880 0.00802	1.19000 0.00668	1.15290 0.00754

3.70	0.13550 0.15206	1.08590 0.01218	1.18050 0.00808	1.19720 0.00695	1.16310 0.00770
3.80	0.14160 0.15245	1.10160 0.01204	1.19120 0.00817	1.20660 0.00714	1.17530 0.00779
3.90	0.14760 0.15284	1.11530 0.01202	1.20420 0.00822	1.21360 0.00741	1.18710 0.00790
4.00	0.15360 0.15325	1.12000 0.01212	1.21680 0.00828	1.21730 0.00781	1.19700 0.00807
4.10	0.15960 0.15366	1.13250 0.01223	1.22850 0.00837	1.22060 0.00821	1.20430 0.00832
4.20	0.16550 0.15408	1.14100 0.01230	1.23820 0.00850	1.22280 0.00865	1.21070 0.00852
4.30	0.17150 0.15452	1.15010 0.01233	1.24480 0.00872	1.22450 0.00909	1.22000 0.00866
4.40	0.17740 0.15496	1.16100 0.01228	1.25510 0.00886	1.22320 0.00953	1.22900 0.00885
4.50	0.18330 0.15542	1.17300 0.01217	1.26120 0.00911	1.22840 0.00987	1.23530 0.00913
4.60	0.18920 0.15588	1.18530 0.01206	1.26150 0.00955	1.22960 0.01041	1.24190 0.00942
4.70	0.19510 0.15635	1.19660 0.01199	1.26340 0.00989	1.23080 0.01103	1.25040 0.00966
4.80	0.20100 0.15683	1.20270 0.01215	1.26000 0.01000	1.23660 0.01167	1.25970 0.00987
4.90	0.20680 0.15733	1.20840 0.01234	1.25790 0.01106	1.23890 0.01253	1.27010 0.01007
5.00	0.21260 0.15783	1.21520 0.01250	1.25230 0.01204	1.23930 0.01339	1.26000 0.01150
5.10	0.21840 0.15834	1.22000 0.01265	1.24960 0.01310	1.24440 0.01396	1.25480 0.01297
5.20	0.22420 0.15886	1.23050 0.01282	1.24780 0.01403	1.24000 0.01495	1.24820 0.01418
5.30	0.23000 0.15939	1.23850 0.01323	1.24180 0.01515	1.23490 0.01595	1.24220 0.01518
5.40	0.23570 0.15993	1.24360 0.01379	1.24180 0.01587	1.22520 0.01720	1.24010 0.01598
5.50	0.24140 0.16048	1.24490 0.01453	1.23380 0.01704	1.21760 0.01827	1.23540 0.01705
5.60	0.24710 0.16104	1.24750 0.01525	1.22610 0.01817	1.21210 0.01923	1.22800 0.01790
5.70	0.25270 0.16161	1.24580 0.01615	1.21960 0.01920	1.21280 0.01989	1.22400 0.01899
5.80	0.25840 0.16219	1.24260 0.01711	1.21300 0.02000	1.21180 0.02064	1.23080 0.01935
5.90	0.26400 0.16277	1.24120 0.01798	1.20650 0.02124	1.20830 0.02158	1.23880 0.01966
6.00	0.26960 0.16337	1.23430 0.01912	1.19870 0.02232	1.21140 0.02216	1.24690 0.01997
6.10	0.27510 0.16398	1.23360 0.01993	1.19670 0.02311	1.21450 0.02273	1.25490 0.02029
6.20	0.28070 0.16460	1.23170 0.02079	1.19230 0.02403	1.21540 0.02340	1.26290 0.02062
6.30	0.28620 0.16522	1.22640 0.02182	1.19260 0.02473	1.21780 0.02401	1.27080 0.02096
6.40	0.29170 0.16586	1.22630 0.02264	1.19350 0.02550	1.22330 0.02447	1.27400 0.02162
6.50	0.29710 0.16651	1.22380 0.02358	1.19430 0.02601	1.22680 0.02503	1.27910 0.02216
6.60	0.30260 0.16716	1.21920 0.02461	1.19560 0.02659	1.22790 0.02571	1.28630 0.02256
6.70	0.30800 0.16783	1.21990 0.02537	1.19750 0.02709	1.23190 0.02627	1.29400 0.02292
6.80	0.31330 0.16850	1.21920 0.02620	1.19960 0.02759	1.23730 0.02677	1.30190 0.02327
6.90	0.31870 0.16919	1.21580 0.02716	1.20300 0.02828	1.24000 0.02743	1.30990 0.02362
7.00	0.32400 0.16988	1.21620 0.02795	1.20610 0.02896	1.24030 0.02800	1.31790 0.02396
7.10	0.32930 0.17058	1.21650 0.02879	1.21020 0.02961	1.24050 0.02909	1.32600 0.02430
7.20	0.33460 0.17130	1.21390 0.02984	1.21700 0.03010	1.24520 0.02964	1.33420 0.02464
7.30	0.33980 0.17202	1.21250 0.03083	1.22380 0.03059	1.25000 0.03020	1.34240 0.02499
7.40	0.34500 0.17275	1.21640 0.03150	1.22980 0.03113	1.25460 0.03077	1.35060 0.02535
7.50	0.35020 0.17350	1.21760 0.03233	1.23500 0.03172	1.25970 0.03131	1.35880 0.02570
7.60	0.35540 0.17425	1.21690 0.03329	1.23950 0.03235	1.26650 0.03177	1.36700 0.02607
7.70	0.36050 0.17501	1.22020 0.03401	1.24440 0.03296	1.27380 0.03220	1.37510 0.02645
7.80	0.36560 0.17578	1.22410 0.03470	1.25100 0.03348	1.28110 0.03264	1.38290 0.02683
7.90	0.37070 0.17657	1.22580 0.03553	1.25760 0.03401	1.28810 0.03310	1.39040 0.02722
8.00	0.37570 0.17736	1.22620 0.03647	1.26340 0.03458	1.29460 0.03359	1.39760 0.02763
8.10	0.38070 0.17816	1.22970 0.03723	1.26810 0.03523	1.30000 0.03400	1.40000 0.02800
8.20	0.38570 0.17897	1.23380 0.03796	1.27130 0.03600	1.30420 0.03476	1.40350 0.02850
8.30	0.39060 0.17979	1.23730 0.03873	1.27340 0.03688	1.30810 0.03539	1.40700 0.02920
8.40	0.39560 0.18062	1.24020 0.03953	1.27820 0.03756	1.31370 0.03593	1.41080 0.02962
8.50	0.40050 0.18146	1.24280 0.04034	1.28390 0.03819	1.31850 0.03647	1.41810 0.03008
8.60	0.40530 0.18231	1.24630 0.04111	1.28930 0.03881	1.33000 0.03680	1.42570 0.03052
8.70	0.41020 0.18317	1.25270 0.04167	1.29450 0.03945	1.34000 0.03710	1.43360 0.03094
8.80	0.41500 0.18404	1.25880 0.04225	1.29910 0.04012	1.35000 0.03750	1.44150 0.03136
8.90	0.41970 0.18492	1.26430 0.04286	1.31650 0.04001	1.36000 0.03780	1.44950 0.03178
9.00	0.42450 0.18581	1.26910 0.04351	1.32310 0.04056	1.36500 0.03820	1.45750 0.03219
9.10	0.42920 0.18671	1.28180 0.04365	1.32920 0.04114	1.37170 0.03867	1.46550 0.03262
9.20	0.43390 0.18762	1.28810 0.04422	1.33460 0.04177	1.37780 0.03925	1.47360 0.03304
9.30	0.43860 0.18854	1.29280 0.04489	1.33920 0.04245	1.38280 0.03990	1.48160 0.03348
9.40	0.44320 0.18947	1.29580 0.04571	1.34300 0.04320	1.38610 0.04065	1.48950 0.03392
9.50	0.44780 0.19041	1.29700 0.04667	1.34630 0.04395	1.38830 0.04148	1.49740 0.03438
9.60	0.45240 0.19136	1.29700 0.04776	1.35090 0.04463	1.39400 0.04209	1.50500 0.03484
9.70	0.45800 0.19200	1.30030 0.04863	1.35630 0.04527	1.40040 0.04267	1.51250 0.03533
9.80	0.46300 0.19300	1.30480 0.04940	1.36140 0.04594	1.39280 0.04412	1.51980 0.03583

```
9.90     0.46620 0.19430     1.30900 0.05018     1.36480 0.04677     1.38450 0.04561     1.52690 0.03635
10.0     0.47070 0.19530     1.31280 0.05100     1.36730 0.04768     1.38380 0.04669     1.53360 0.03689];
```

A.5 Mission Profile Code

```
%=========================================================================

%                   7 POINT FLIGHT PROFILE CODE

%=========================================================================

%       This code is used to develope new flight profiles executed in the
%   performance code.  The variables are time, altitude, and Mach number.

%=========================================================================

% Cruise Begin/End
cr_b = 3;
cr_e = 5;

% Flight Profile
if flt_prof == 2;
FLT_PROF = [...
40.00       50000.00 0.55
290.00      56500.00 0.56
540.00      60000.00 0.57
1260.00     66500.00 0.58
1980.00     70000.00 0.59
2100.00     60000.00 0.58
2480.00     50000.00 0.57];

else
FLT_PROF = [...
40.00       50000.00 0.58
290.00      56500.00 0.55
540.00      60000.00 0.53
1260.00     66500.00 0.51
1980.00     70000.00 0.51
2230.00     60000.00 0.50
2480.00     50000.00 0.50];
end


% Time Data (mins)
t = FLT_PROF(:,1);

% Alititude Data (ft)
h = FLT_PROF(:,2);

% Mach Data
M = FLT_PROF(:,3);
```

A.6 The LRN-1015 Airfoil Geometry Code

```
%=========================================================================

%                    LRN-1015 AIRFOIL CODE

%=========================================================================

%      This code is used to generate the airfoil geometry for the LRN-1015
%   airfoil.

%=========================================================================


LRN_airfoil_top = [...
   % UPPER
0.000000  0.000000
0.001621  0.017070
0.006475  0.026197
0.014529  0.036312
0.025732  0.046990
0.040010  0.057142
0.057272  0.066677
0.077405  0.075746
0.100279  0.084202
0.125745  0.092004
0.153638  0.099088
0.183777  0.105433
0.215968  0.110974
0.250000  0.115663
0.285654  0.119458
0.322698  0.122299
0.360891  0.124114
0.399987  0.124809
0.439732  0.124264
0.479867  0.122351
0.520133  0.118919
0.560268  0.113813
0.600013  0.106947
0.639109  0.098407
0.677302  0.088455
0.714346  0.077492
0.750000  0.066230
0.784032  0.055549
0.816223  0.046102
0.846362  0.037889
0.874255  0.030724
0.899721  0.024801
0.922595  0.020019
0.942728  0.015794
0.959990  0.011820
0.974268  0.008107
0.985471  0.004834
0.993525  0.002244
0.998379  0.000577
1.000000  0.000000];

LRN_airfoil_bottom = [...
   % LOWER
```

```
0.000000  0.000000
0.001621 -0.001980
0.006475 -0.004930
0.014529 -0.007510
0.025732 -0.010010
0.040010 -0.012530
0.057272 -0.014980
0.077405 -0.017230
0.100279 -0.019320
0.125745 -0.021210
0.153638 -0.022880
0.183777 -0.024300
0.215968 -0.025470
0.250000 -0.026360
0.285654 -0.026980
0.322698 -0.027300
0.360891 -0.027330
0.399987 -0.027070
0.439732 -0.026510
0.479867 -0.025650
0.520133 -0.024490
0.560268 -0.023030
0.600013 -0.021270
0.639109 -0.019240
0.677302 -0.016870
0.714346 -0.014080
0.750000 -0.010940
0.784032 -0.007660
0.816223 -0.004650
0.846362 -0.002130
0.874255 -0.000220
0.899721  0.001069
0.922595  0.001761
0.942728  0.001957
0.959990  0.001792
0.974268  0.001378
0.985471  0.000884
0.993525  0.000429
0.998379  0.000113
1.000000  0.000000];

cf_m_2_inches = 98.425196850;

x_top_loc = LRN_airfoil_top(:,1)*cf_m_2_inches;
y_top_loc = LRN_airfoil_top(:,2)*cf_m_2_inches;
x_bot_loc = LRN_airfoil_bottom(:,1)*cf_m_2_inches;
y_bot_loc = LRN_airfoil_bottom(:,2)*cf_m_2_inches;

plot(x_top_loc,y_top_loc),hold on,plot(x_bot_loc,y_bot_loc)
axis equal,axis([0 cf_m_2_inches -10 20])
xlabel('Airfoil X - Coordinate (inches)'),ylabel('Airfoil Y - Coordinate (inches)')
```

A.7 Roskam Drag Estimation Chart Regeneration Code

```
%=========================================================================
```

```
%                ROSKAM DRAG ESTIMATION CHARTS CODE

%=========================================================================

%      This code is used to replicate the drag buildup charts used by
%   Roskam for the text of the thesis.

%=========================================================================

RN_fus_chart = [...
    % R_N          M = 0.25      0.40      0.60      0.70      0.80      0.85      0.90
      3.000E+06    1.0620    1.0200    0.9800    0.9555    0.9250    0.9025    0.8680
      4.000E+06    1.0670    1.0220    0.9820    0.9580    0.9270    0.9050    0.8700
      5.000E+06    1.0710    1.0240    0.9845    0.9610    0.9290    0.9080    0.8720
      6.000E+06    1.0723    1.0270    0.9868    0.9625    0.9320    0.9105    0.8740
      7.000E+06    1.0740    1.0290    0.9890    0.9650    0.9350    0.9120    0.8780
      8.000E+06    1.0750    1.0310    0.9920    0.9680    0.9370    0.9150    0.8795
      9.000E+06    1.0760    1.0330    0.9950    0.9700    0.9390    0.9170    0.8815
      1.000E+07    1.0760    1.0365    0.9965    0.9718    0.9415    0.9200    0.8830
      1.500E+07    1.0725    1.0500    1.0085    0.9845    0.9525    0.9315    0.8960
      2.000E+07    1.0650    1.0590    1.0200    0.9955    0.9660    0.9430    0.9095
      3.000E+07    1.0440    1.0490    1.0380    1.0128    0.9890    0.9685    0.9230
      4.000E+07    0.9910    1.0175    1.0310    1.0145    1.0035    0.9900    0.9730
      5.000E+07    0.9510    1.0000    1.0240    1.0145    1.0100    1.0050    0.9940
      6.000E+07    0.9380    0.9900    1.0180    1.0145    1.0138    1.0100    1.0065
      7.000E+07    0.9340    0.9880    1.0155    1.0145    1.0140    1.0138    1.0130
      8.000E+07    0.9300    0.9830    1.0145    1.0145    1.0145    1.0145    1.0145
      9.000E+07    0.9290    0.9810    1.0145    1.0145    1.0145    1.0145    1.0145
      1.000E+08    0.9285    0.9795    1.0145    1.0145    1.0145    1.0145    1.0145
      1.500E+08    0.9258    0.9775    1.0145    1.0145    1.0145    1.0145    1.0145
      2.000E+08    0.9230    0.9755    1.0145    1.0145    1.0145    1.0145    1.0145
      3.000E+08    0.9230    0.9750    1.0145    1.0145    1.0145    1.0145    1.0145
      4.000E+08    0.9230    0.9750    1.0145    1.0145    1.0145    1.0145    1.0145
      5.000E+08    0.9230    0.9750    1.0145    1.0145    1.0145    1.0145    1.0145];

%                                                    semilogx(RN_fus_chart(:,1),RN_fus_chart(:,2),'b'),hold
on,semilogx(RN_fus_chart(:,1),RN_fus_chart(:,3),'g'),hold on
%                                                     semilogx(RN_fus_chart(:,1),RN_fus_chart(:,4),'r'),hold
on,semilogx(RN_fus_chart(:,1),RN_fus_chart(:,5),'c'),hold on
%                                                    semilogx(RN_fus_chart(:,1),RN_fus_chart(:,6),'m'),hold
on,semilogx(RN_fus_chart(:,1),RN_fus_chart(:,7),'y'),hold on
%    semilogx(RN_fus_chart(:,1),RN_fus_chart(:,8),'k'),grid
%    xlabel('Fuselage Reynolds Number (R_N_f_u_s)'),ylabel('Wing-Fuselage Interference Factor (R_w_f)')
%    text(2.25E6,1.08,'M'),text(2.0E6,1.06,'0.25'),text(2.0E6,1.02,'0.40'),text(2.0E6,0.98,'0.60'),text(2.0E6,0.955,'0.70')
%    text(2.0E6,0.925,'0.80'),text(2.0E6,0.90,'0.85'),text(2.0E6,0.868,'0.90'),figure

R_LS_chart = [...
    % D  M = 0.25 0.60      0.80      0.90
      0.50 0.810    0.880     1.000     1.098
      0.55 0.848    0.920     1.036     1.133
      0.60 0.886    0.960     1.072     1.169
      0.65 0.924    1.000     1.108     1.204
      0.70 0.962    1.040     1.144     1.240
      0.75 1.000    1.080     1.180     1.275
      0.80 1.025    1.110     1.213     1.305
      0.85 1.050    1.128     1.230     1.326
      0.90 1.065    1.140     1.250     1.345
      0.95 1.070    1.147     1.258     1.355
      1.00 1.070    1.147     1.258     1.355];

%    plot(R_LS_chart(:,1),R_LS_chart(:,2),'b'),hold on,plot(R_LS_chart(:,1),R_LS_chart(:,3),'r'),hold on
```

```
%    plot(R_LS_chart(:,1),R_LS_chart(:,4),'g'),hold on,plot(R_LS_chart(:,1),R_LS_chart(:,5),'c'),grid
%    axis([0.40 1.10 0.70 1.4]),xlabel('cos(wing sweep angle)'),ylabel('Lifting Surface Correction Factor (R_L_S)')
%    text(0.46,1.16,'M'),text(0.45,1.11,'0.90'),text(0.45,1.01,'0.80'),text(0.45,0.88,'0.60'),text(0.435,0.81,'< 0.25'),figure

C_F_chart = [...
  %  RN    M =     0.00       0.50       1.00
     4.00E+05    0.005300    0.005150    0.004900
     1.00E+06       0.004465 0.004290 0.004115
     3.00E+06       0.003790 0.003600 0.003410
     5.00E+06       0.003380 0.003253 0.003125
     1.00E+07       0.003010 0.002895 0.002780
     3.00E+07       0.002525 0.002433 0.002340
     5.00E+07       0.002325 0.002245 0.002165
     1.00E+08       0.002125 0.002040 0.001955
     3.00E+08       0.001845 0.001763 0.001680
     5.00E+08       0.001715 0.001648 0.001580];

%    semilogx(C_F_chart(:,1),C_F_chart(:,2),'b'),hold on,semilogx(C_F_chart(:,1),C_F_chart(:,3),'r'),hold on
%    semilogx(C_F_chart(:,1),C_F_chart(:,4),'g'),grid,xlabel('Reynolds Number (R_N)')
%    ylabel('Turbulent Mean Skin-Friction Coefficient (C_f)')
%    text(1.5E7,0.0041,'M'),text(1.35E7,0.0037,'0.0'),text(1.35E7,0.0034,'0.5'),text(1.35E7,0.0031,'1.0')
```

A.8 Roskam Drag Buildup Chart Interpolation Code

```
%=========================================================================

%                    ROSKAM DRAG BUILDUP CODE

%=========================================================================

%      This code consists of all the points that define the charts
%   in the Roskam drag buildup.  This is used in the performance code
%   to interpolate exact wing-fuselage interfernce factors, lifting
%   surface correction factors, and skin friction coefficient factors.

%=========================================================================


% Wing/Fuselage Intergerence Factor (Figure 4.1)
R_wf = [...
      1.0620      1.0480    1.0340    1.0200    1.0100    1.0000    0.9900    0.9800    0.9678    0.9555    0.9403
           0.9250    0.9025    0.8680
      1.0670      1.0520    1.0370    1.0220    1.0120    1.0020    0.9920    0.9820    0.9700    0.9580    0.9425
           0.9270    0.9050    0.8700
      1.0710      1.0553    1.0397    1.0240    1.0141    1.0043    0.9944    0.9845    0.9728    0.9610    0.9450
           0.9290    0.9080    0.8720
      1.0723      1.0572    1.0421    1.0270    1.0170    1.0069    0.9969    0.9868    0.9747    0.9625    0.9473
           0.9320    0.9105    0.8740
      1.0740      1.0590    1.0440    1.0290    1.0190    1.0090    0.9990    0.9890    0.9770    0.9650    0.9500
           0.9350    0.9120    0.8780
      1.0750      1.0603    1.0457    1.0310    1.0213    1.0115    1.0018    0.9920    0.9800    0.9680    0.9525
           0.9370    0.9150    0.8795
      1.0760      1.0617    1.0473    1.0330    1.0235    1.0140    1.0045    0.9950    0.9825    0.9700    0.9545
           0.9390    0.9170    0.8815
```

```
1.0760    1.0628    1.0497    1.0365    1.0265    1.0165    1.0065    0.9965    0.9842    0.9718    0.9567
    0.9415    0.9200    0.8830
1.0725    1.0650    1.0575    1.0500    1.0396    1.0293    1.0189    1.0085    0.9965    0.9845    0.9685
    0.9525    0.9315    0.8960
1.0650    1.0630    1.0610    1.0590    1.0493    1.0395    1.0298    1.0200    1.0078    0.9955    0.9808
    0.9660    0.9430    0.9095
1.0440    1.0457    1.0473    1.0490    1.0463    1.0435    1.0408    1.0380    1.0254    1.0128    1.0009
    0.9890    0.9685    0.9230
0.9910    0.9998    1.0087    1.0175    1.0209    1.0243    1.0276    1.0310    1.0228    1.0145    1.0090
    1.0035    0.9900    0.9730
0.9510    0.9673    0.9837    1.0000    1.0060    1.0120    1.0180    1.0240    1.0193    1.0145    1.0123
    1.0100    1.0050    0.9940
0.9380    0.9553    0.9727    0.9900    0.9970    1.0040    1.0110    1.0180    1.0163    1.0145    1.0142
    1.0138    1.0100    1.0065
0.9340    0.9520    0.9700    0.9880    0.9949    1.0018    1.0086    1.0155    1.0150    1.0145    1.0143
    1.0140    1.0138    1.0130
0.9300    0.9477    0.9653    0.9830    0.9909    0.9988    1.0066    1.0145    1.0145    1.0145    1.0145
    1.0145    1.0145    1.0145
0.9290    0.9463    0.9637    0.9810    0.9894    0.9978    1.0061    1.0145    1.0145    1.0145    1.0145
    1.0145    1.0145    1.0145
0.9285    0.9455    0.9625    0.9795    0.9883    0.9970    1.0058    1.0145    1.0145    1.0145    1.0145
    1.0145    1.0145    1.0145
0.9258    0.9430    0.9603    0.9775    0.9868    0.9960    1.0053    1.0145    1.0145    1.0145    1.0145
    1.0145    1.0145    1.0145
0.9230    0.9405    0.9580    0.9755    0.9853    0.9950    1.0048    1.0145    1.0145    1.0145    1.0145
    1.0145    1.0145    1.0145
0.9230    0.9403    0.9577    0.9750    0.9849    0.9948    1.0046    1.0145    1.0145    1.0145    1.0145
    1.0145    1.0145    1.0145
0.9230    0.9403    0.9577    0.9750    0.9849    0.9948    1.0046    1.0145    1.0145    1.0145    1.0145
    1.0145    1.0145    1.0145
0.9230    0.9403    0.9577    0.9750    0.9849    0.9948    1.0046    1.0145    1.0145    1.0145    1.0145
    1.0145    1.0145    1.0145];

% Lifting Surface Correction Factor (Figure 4.2)
R_LS = [...
    -33.987    165.76  -329.77    340.71  -192.73    57.443  -6.3556
    -51.765    247.67  -484.76    494.81  -277.62    82.011  -9.2656
    -69.542    329.57  -639.74    648.91  -362.5     106.58  -12.176
    -87.32     411.48  -794.73    803.0   -447.39    131.15  -15.086
    -105.1     493.38  -949.72    957.1   -532.27    155.71  -17.996
    -122.88    575.29  -1104.7    1111.2  -617.16    180.28  -20.906
    -140.65    657.19  -1259.7    1265.3  -702.04    204.85  -23.816
    -158.43    739.1   -1414.7    1419.4  -786.93    229.42  -26.726
    -147.58    687.32  -1314.1    1317.7  -730.33    212.94  -24.73
    -136.73    635.55  -1213.6    1216.0  -673.73    196.47  -22.735
    -125.88    583.78  -1113.1    1114.3  -617.13    179.99  -20.739
    -115.03    532.01  -1012.6    1012.6  -560.54    163.52  -18.744
    -108.76    500.18  -947.19    942.91  -519.8     151.11  -17.154
    -102.48    468.36  -881.82    873.22  -479.05    138.7   -15.564];

% Turbulent Mean Skin-Friction Coefficient (Figure 4.3)
C_f = [...
    0.004465    0.003790 0.003380 0.003010 0.002525 0.002325 0.002125 0.001845 0.001715
    0.004290    0.003600 0.003253 0.002895 0.002433 0.002245 0.002040 0.001763 0.001648
    0.004115    0.003410 0.003125 0.002780 0.002340 0.002165 0.001955 0.001680 0.001580];
```

Appendix B. MATLAB Produced Spanwise Aerodynamic Performance

This Appendix lists MATLAB output results for individual wing strip sections used for the Roskam/AVTIE strip method. Panel 0, 1, 2 and 3 refer to the forward inside wing, aft wing, joint section, and outboard wing sections. These panels are then subdivided into strips, varying for each panel.

Strip Drag Buildup Results:

Panel	Strip	PT 1 (Re = 5.3547e6)						PT 2 (Re = 3.9888e6)					
		CL	CDp	CDi	AOA	aoaL	AOAi	CL	CDp	CDi	AOA	aoaL	AOAi
0	0	0.5486	0.00640	0.03569	2.4244	-1.3055	3.7299	0.5737	0.00638	0.03864	2.7378	-1.1245	3.8624
0	1	0.5998	0.00625	0.03514	2.4244	-0.9345	3.3589	0.6266	0.00579	0.03759	2.7378	-0.7014	3.4392
0	2	0.6392	0.00567	0.03401	2.4244	-0.6255	3.0499	0.6680	0.00533	0.03670	2.7378	-0.4112	3.1490
0	3	0.6659	0.00536	0.03313	2.4244	-0.4273	2.8517	0.6961	0.00515	0.03564	2.7378	-0.1970	2.9349
0	4	0.6803	0.00524	0.03256	2.4244	-0.3187	2.7431	0.7113	0.00513	0.03498	2.7378	-0.0808	2.8186
0	5	0.7254	0.00511	0.03037	2.4244	0.0251	2.3993	0.7587	0.00505	0.03271	2.7378	0.2672	2.4706
0	6	0.5428	0.00640	0.03571	2.4244	-1.3478	3.7722	0.5686	0.00638	0.03866	2.7378	-1.1613	3.8992
0	7	0.1140	0.01046	0.01378	2.4244	-4.5170	6.9414	0.1216	0.01028	0.01524	2.7378	-4.4596	7.1974
1	0	0.5680	0.00638	0.01225	0.0699	-1.1654	1.2353	0.5868	0.00635	0.01273	0.2131	-1.0296	1.2427
1	1	0.5424	0.00640	0.01456	0.1880	-1.3509	1.5388	0.5603	0.00639	0.01525	0.3386	-1.2209	1.5595
1	2	0.5399	0.00640	0.01767	0.5066	-1.3683	1.8749	0.5584	0.00639	0.01866	0.6801	-1.2352	1.9153
1	3	0.5400	0.00640	0.02143	0.9067	-1.3677	2.2744	0.5598	0.00639	0.02281	1.1106	-1.2248	2.3355
1	4	0.5331	0.00641	0.02533	1.3057	-1.4181	2.7238	0.5538	0.00639	0.02713	1.5394	-1.2680	2.8074
1	5	0.4999	0.00645	0.02933	1.7057	-1.6583	3.3640	0.5194	0.00643	0.03157	1.9682	-1.5168	3.4850
1	6	0.5640	0.00639	0.03247	2.1058	-1.1946	3.3004	0.5850	0.00636	0.03510	2.3969	-1.0432	3.4401
1	7	0.7559	0.00505	0.02793	2.3648	0.2471	2.1177	0.7857	0.00503	0.03037	2.6742	0.4592	2.2150
2	0	0.4466	0.00661	0.03484	2.4244	-2.0509	4.4753	0.4690	0.00652	0.03780	2.7378	-1.8850	4.6228
2	1	0.4984	0.00646	0.03558	2.4244	-1.6691	4.0935	0.5223	0.00643	0.03856	2.7378	-1.4957	4.2335
2	2	0.5308	0.00642	0.03572	2.4244	-1.4346	3.8590	0.5540	0.00639	0.03869	2.7378	-1.2669	4.0047
2	3	0.5757	0.00638	0.03549	2.4244	-1.1102	3.5346	0.6009	0.00624	0.03840	2.7378	-0.9257	3.6636
3	0	0.7862	0.00503	0.02691	2.4244	0.4627	1.9617	0.8213	0.00503	0.02903	2.7378	0.7127	2.0252
3	1	0.7620	0.00504	0.02837	2.4244	0.2907	2.1337	0.7961	0.00503	0.03063	2.7378	0.5332	2.2047
3	2	0.7600	0.00504	0.02848	2.4244	0.2765	2.1479	0.7940	0.00503	0.03075	2.7378	0.5180	2.2199
3	3	0.7536	0.00506	0.02885	2.4244	0.2307	2.1937	0.7873	0.00503	0.03115	2.7378	0.4703	2.2676
3	4	0.7468	0.00507	0.02923	2.4244	0.1815	2.2429	0.7800	0.00503	0.03156	2.7378	0.4190	2.3189
3	5	0.7392	0.00509	0.02963	2.4244	0.1269	2.2975	0.7720	0.00503	0.03200	2.7378	0.3622	2.3757
3	6	0.7298	0.00511	0.03014	2.4244	0.0578	2.3666	0.7621	0.00504	0.03253	2.7378	0.2918	2.4460
3	7	0.7183	0.00512	0.03074	2.4244	-0.0283	2.4527	0.7500	0.00507	0.03315	2.7378	0.2046	2.5332
3	8	0.7026	0.00514	0.03153	2.4244	-0.1475	2.5719	0.7336	0.00510	0.03395	2.7378	0.0854	2.6524
3	9	0.6807	0.00524	0.03253	2.4244	-0.3149	2.7393	0.7107	0.00513	0.03501	2.7378	-0.0854	2.8233
3	10	0.6466	0.00560	0.03383	2.4244	-0.5744	2.9988	0.6751	0.00528	0.03646	2.7378	-0.3579	3.0958
3	11	0.5961	0.00630	0.03521	2.4244	-0.9621	3.3865	0.6226	0.00586	0.03773	2.7378	-0.7364	3.4743
3	12	0.4888	0.00647	0.03549	2.4244	-1.7396	4.1640	0.5120	0.00643	0.03846	2.7378	-1.5702	4.3081
3	13	0.4708	0.00651	0.03527	2.4244	-1.8717	4.2961	0.4933	0.00646	0.03823	2.7378	-1.7067	4.4446
3	14	0.3449	0.00700	0.03136	2.4244	-2.7918	5.2162	0.3645	0.00686	0.03421	2.7378	-2.6484	5.3863
3	15	0.3654	0.00685	0.03227	2.4244	-2.6419	5.0663	0.3974	0.00675	0.03565	2.7378	-2.4099	5.1478

Panel	Strip	PT 3 (Re = 3.4312e6)						PT 4 (Re = 2.5544e6)					
		CL	CDp	CDi	AOA	aoaL	AOAi	CL	CDp	CDi	AOA	aoaL	AOAi

0	0	0.6317	0.00574	0.04562	3.4707	-0.6708	4.1415	0.5791	0.00637	0.03903	2.7790	-1.0853	3.8642
0	1	0.6887	0.00519	0.04475	3.4707	-0.2549	3.7256	0.6330	0.00573	0.03800	2.7790	-0.6627	3.4416
0	2	0.7346	0.00510	0.04328	3.4707	0.0930	3.3776	0.6753	0.00528	0.03693	2.7790	-0.3562	3.1352
0	3	0.7659	0.00503	0.04211	3.4707	0.3190	3.1517	0.7040	0.00514	0.03581	2.7790	-0.1371	2.9160
0	4	0.7829	0.00503	0.04140	3.4707	0.4395	3.0312	0.7193	0.00512	0.03513	2.7790	-0.0209	2.7998
0	5	0.8357	0.00504	0.03870	3.4707	0.8160	2.6546	0.7672	0.00503	0.03281	2.7790	0.3283	2.4506
0	6	0.6276	0.00578	0.04559	3.4707	-0.6951	4.1658	0.5749	0.00638	0.03905	2.7790	-1.1155	3.8945
0	7	0.1373	0.00993	0.01866	3.4707	-4.3408	7.8115	0.1225	0.01026	0.01542	2.7790	-4.4531	7.2321
1	0	0.6486	0.00558	0.01442	0.7153	-0.5589	1.2742	0.4911	0.00647	0.00875	-0.7019	-1.7226	1.0207
1	1	0.6189	0.00599	0.01772	0.8528	-0.7877	1.6405	0.4732	0.00651	0.01095	-0.5283	-1.8545	1.3262
1	2	0.6154	0.00604	0.02191	1.2253	-0.8149	2.0402	0.4844	0.00648	0.01449	-0.0577	-1.7722	1.7145
1	3	0.6162	0.00603	0.02692	1.6948	-0.8090	2.5038	0.5028	0.00645	0.01906	0.5347	-1.6374	2.1721
1	4	0.6091	0.00613	0.03216	2.1627	-0.8638	3.0265	0.5138	0.00643	0.02405	1.1261	-1.5574	2.6834
1	5	0.5694	0.00638	0.03760	2.6304	-1.1556	3.7861	0.4959	0.00646	0.02946	1.7179	-1.6878	3.4057
1	6	0.6372	0.00569	0.04152	3.0987	-0.6371	3.7359	0.5766	0.00638	0.03432	2.3090	-1.1034	3.4124
1	7	0.8575	0.00511	0.03631	3.4013	0.9749	2.4265	0.7871	0.00503	0.03052	2.6913	0.4694	2.2219
2	0	0.5192	0.00643	0.04515	3.4707	-1.5183	4.9890	0.4715	0.00651	0.03819	2.7790	-1.8670	4.6460
2	1	0.5772	0.00638	0.04599	3.4707	-1.0992	4.5699	0.5265	0.00642	0.03897	2.7790	-1.4655	4.2445
2	2	0.6085	0.00614	0.04604	3.4707	-0.8680	4.3387	0.5587	0.00639	0.03909	2.7790	-1.2328	4.0117
2	3	0.6602	0.00545	0.04537	3.4707	-0.4694	3.9401	0.6060	0.00617	0.03875	2.7790	-0.8870	3.6660
3	0	0.9030	0.00525	0.03334	3.4707	1.3548	2.1159	0.8287	0.00504	0.02911	2.7790	0.7658	2.0132
3	1	0.8754	0.00515	0.03573	3.4707	1.1312	2.3395	0.8037	0.00503	0.03074	2.7790	0.5869	2.1920
3	2	0.8728	0.00513	0.03603	3.4707	1.1046	2.3661	0.8019	0.00503	0.03085	2.7790	0.5741	2.2048
3	3	0.8652	0.00513	0.03660	3.4707	1.0464	2.4242	0.7954	0.00503	0.03124	2.7790	0.5277	2.2513
3	4	0.8570	0.00510	0.03742	3.4707	0.9682	2.5025	0.7883	0.00503	0.03166	2.7790	0.4774	2.3016
3	5	0.8480	0.00507	0.03808	3.4707	0.8968	2.5738	0.7804	0.00503	0.03210	2.7790	0.4213	2.3577
3	6	0.8369	0.00504	0.03863	3.4707	0.8249	2.6458	0.7705	0.00503	0.03263	2.7790	0.3515	2.4275
3	7	0.8233	0.00503	0.03941	3.4707	0.7268	2.7438	0.7583	0.00505	0.03327	2.7790	0.2647	2.5143
3	8	0.8052	0.00503	0.04036	3.4707	0.5978	2.8729	0.7419	0.00508	0.03408	2.7790	0.1463	2.6327
3	9	0.7800	0.00503	0.04153	3.4707	0.4187	3.0519	0.7189	0.00512	0.03515	2.7790	-0.0239	2.8028
3	10	0.7411	0.00508	0.04305	3.4707	0.1405	3.3301	0.6829	0.00522	0.03666	2.7790	-0.2986	3.0775
3	11	0.6841	0.00521	0.04486	3.4707	-0.2898	3.7604	0.6299	0.00576	0.03802	2.7790	-0.6817	3.4606
3	12	0.5661	0.00639	0.04589	3.4707	-1.1794	4.6500	0.5182	0.00643	0.03889	2.7790	-1.5256	4.3045
3	13	0.5469	0.00640	0.04566	3.4707	-1.3179	4.7885	0.4992	0.00645	0.03867	2.7790	-1.6635	4.4425
3	14	0.4100	0.00672	0.04136	3.4707	-2.3188	5.7894	0.3698	0.00683	0.03473	2.7790	-2.6100	5.3890
3	15	0.4655	0.00653	0.04366	3.4707	-1.9109	5.3815	0.4069	0.00673	0.03631	2.7790	-2.3413	5.1202

Panel	Strip	PT 5 (Re = 2.1960e6)						PT 5 (Re = 3.4914e6)					
		CL	CDp	CDi	AOA	aoaL	AOAi	CL	CDp	CDi	AOA	aoaL	AOAi
0	0	0.5093	0.00644	0.03078	1.8752	-1.5896	3.4648	0.4328	0.00669	0.02292	0.8820	-2.1528	3.0348
0	1	0.5590	0.00639	0.03029	1.8752	-1.2303	3.1056	0.4778	0.00649	0.02253	0.8820	-1.8206	2.7026
0	2	0.5963	0.00630	0.02950	1.8752	-0.9607	2.8360	0.5096	0.00644	0.02196	0.8820	-1.5880	2.4700
0	3	0.6213	0.00591	0.02851	1.8752	-0.7545	2.6298	0.5306	0.00642	0.02146	0.8820	-1.4357	2.3176
0	4	0.6343	0.00572	0.02800	1.8752	-0.6551	2.5304	0.5414	0.00640	0.02116	0.8820	-1.3580	2.2400
0	5	0.6758	0.00527	0.02627	1.8752	-0.3527	2.2280	0.5759	0.00638	0.02000	0.8820	-1.1085	1.9905
0	6	0.5044	0.00645	0.03080	1.8752	-1.6255	3.5008	0.4287	0.00670	0.02292	0.8820	-2.1830	3.0650
0	7	0.1029	0.01070	0.01161	1.8752	-4.6013	6.4766	0.0855	0.01113	0.00837	0.8820	-4.7343	5.6163
1	0	0.2876	0.00776	0.00348	-2.5194	-3.2131	0.6937	0.2007	0.00852	0.00203	-3.2802	-3.8589	0.5787
1	1	0.2840	0.00779	0.00464	-2.2999	-3.2355	0.9356	0.2016	0.00850	0.00274	-3.0728	-3.8517	0.7790
1	2	0.3141	0.00739	0.00722	-1.7065	-3.0245	1.3180	0.2344	0.00801	0.00448	-2.5109	-3.6056	1.0947
1	3	0.3552	0.00692	0.01090	-0.9584	-2.7166	1.7582	0.2774	0.00785	0.00714	-1.8024	-3.2775	1.4750
1	4	0.3894	0.00676	0.01533	-0.2113	-2.4675	2.2562	0.3144	0.00739	0.01058	-1.0943	-3.0221	1.9278
1	5	0.3994	0.00674	0.02042	0.5353	-2.3955	2.9307	0.3319	0.00714	0.01449	-0.3869	-2.8887	2.5018
1	6	0.4965	0.00646	0.02568	1.2820	-1.6832	2.9653	0.4256	0.00670	0.01876	0.3203	-2.2055	2.5258
1	7	0.6939	0.00516	0.02397	1.7653	-0.2140	1.9793	0.5971	0.00629	0.01805	0.7777	-0.9546	1.7323
2	0	0.4083	0.00672	0.02995	1.8752	-2.3310	4.2063	0.3468	0.00698	0.02214	0.8820	-2.7785	3.6604
2	1	0.4589	0.00655	0.03069	1.8752	-1.9596	3.8349	0.3890	0.00676	0.02275	0.8820	-2.4701	3.3521
2	2	0.4921	0.00647	0.03082	1.8752	-1.7158	3.5910	0.4204	0.00671	0.02292	0.8820	-2.2432	3.1252
2	3	0.5337	0.00641	0.03062	1.8752	-1.4137	3.2890	0.4557	0.00656	0.02278	0.8820	-1.9832	2.8652
3	0	0.7296	0.00511	0.02316	1.8752	0.0561	1.8192	0.6231	0.00585	0.01753	0.8820	-0.7298	1.6118
3	1	0.7078	0.00513	0.02449	1.8752	-0.1073	1.9825	0.6047	0.00619	0.01877	0.8820	-0.8973	1.7793
3	2	0.7071	0.00513	0.02453	1.8752	-0.1133	1.9886	0.6048	0.00619	0.01877	0.8820	-0.8964	1.7784
3	3	0.7019	0.00514	0.02484	1.8752	-0.1531	2.0284	0.6009	0.00624	0.01896	0.8820	-0.9261	1.8081
3	4	0.6962	0.00515	0.02517	1.8752	-0.1965	2.0718	0.5965	0.00630	0.01917	0.8820	-0.9592	1.8412
3	5	0.6898	0.00518	0.02554	1.8752	-0.2464	2.1216	0.5915	0.00634	0.01939	0.8820	-0.9964	1.8784
3	6	0.6816	0.00523	0.02597	1.8752	-0.3088	2.1840	0.5849	0.00636	0.01965	0.8820	-1.0439	1.9259
3	7	0.6712	0.00531	0.02650	1.8752	-0.3870	2.2622	0.5764	0.00638	0.01998	0.8820	-1.1051	1.9870
3	8	0.6570	0.00549	0.02715	1.8752	-0.4931	2.3683	0.5645	0.00639	0.02042	0.8820	-1.1910	2.0729

3	9	0.6369	0.00569	0.02794	1.8752	-0.6393	2.5146	0.5473	0.00640	0.02098	0.8820	-1.3154	2.1974
3	10	0.6049	0.00619	0.02924	1.8752	-0.8956	2.7708	0.5195	0.00643	0.02174	0.8820	-1.5161	2.3980
3	11	0.5574	0.00639	0.03031	1.8752	-1.2425	3.1178	0.4777	0.00649	0.02253	0.8820	-1.8210	2.7030
3	12	0.4543	0.00657	0.03065	1.8752	-1.9936	3.8689	0.3837	0.00678	0.02269	0.8820	-2.5090	3.3909
3	13	0.4357	0.00668	0.03044	1.8752	-2.1313	4.0066	0.3633	0.00686	0.02243	0.8820	-2.6571	3.5391
3	14	0.3160	0.00736	0.02691	1.8752	-3.0095	4.8848	0.2567	0.00768	0.01933	0.8820	-3.4371	4.3191
3	15	0.3272	0.00720	0.02737	1.8752	-2.9247	4.7999	0.2503	0.00777	0.01906	0.8820	-3.4859	4.3679

Panel	Strip	PT 7 (Re = 5.5494e6)					
		CL	CDp	CDi	AOA	aoaL	AOAi
0	0	0.3191	0.00731	0.01334	-0.5899	-2.9858	2.3959
0	1	0.3569	0.00691	0.01317	-0.5899	-2.7041	2.1142
0	2	0.3804	0.00679	0.01290	-0.5899	-2.5324	1.9425
0	3	0.3956	0.00675	0.01265	-0.5899	-2.4226	1.8327
0	4	0.4030	0.00673	0.01251	-0.5899	-2.3695	1.7796
0	5	0.4269	0.00670	0.01197	-0.5899	-2.1963	1.6064
0	6	0.3162	0.00736	0.01334	-0.5899	-3.0082	2.4183
0	7	0.0610	0.01181	0.00459	-0.5899	-4.9054	4.3155
1	0	0.0744	0.01144	0.00056	-4.3876	-4.8176	0.4300
1	1	0.0819	0.01123	0.00081	-4.1985	-4.7623	0.5638
1	2	0.1183	0.01036	0.00165	-3.6855	-4.4848	0.7994
1	3	0.1635	0.00932	0.00315	-3.0392	-4.1422	1.1031
1	4	0.2037	0.00846	0.00513	-2.3938	-3.8355	1.4417
1	5	0.2317	0.00805	0.00759	-1.7479	-3.6261	1.8782
1	6	0.3190	0.00732	0.01049	-1.1028	-2.9871	1.8843
1	7	0.4512	0.00659	0.01048	-0.6856	-2.0168	1.3312
2	0	0.2569	0.00767	0.01275	-0.5899	-3.4359	2.8460
2	1	0.2855	0.00777	0.01313	-0.5899	-3.2263	2.6364
2	2	0.3132	0.00740	0.01334	-0.5899	-3.0313	2.4414
2	3	0.3391	0.00706	0.01328	-0.5899	-2.8354	2.2455
3	0	0.4643	0.00653	0.01078	-0.5899	-1.9200	1.3301
3	1	0.4507	0.00659	0.01125	-0.5899	-2.0200	1.4301
3	2	0.4522	0.00658	0.01120	-0.5899	-2.0091	1.4192
3	3	0.4501	0.00659	0.01127	-0.5899	-2.0247	1.4348
3	4	0.4477	0.00660	0.01135	-0.5899	-2.0426	1.4527
3	5	0.4447	0.00662	0.01144	-0.5899	-2.0647	1.4748
3	6	0.4404	0.00665	0.01158	-0.5899	-2.0964	1.5065
3	7	0.4346	0.00668	0.01175	-0.5899	-2.1396	1.5497
3	8	0.4261	0.00670	0.01199	-0.5899	-2.2019	1.6120
3	9	0.4133	0.00672	0.01230	-0.5899	-2.2946	1.7047
3	10	0.3919	0.00676	0.01272	-0.5899	-2.4496	1.8597
3	11	0.3586	0.00689	0.01315	-0.5899	-2.6916	2.1018
3	12	0.2778	0.00784	0.01301	-0.5899	-3.2752	2.6853
3	13	0.2538	0.00772	0.01270	-0.5899	-3.4590	2.8691
3	14	0.1679	0.00922	0.01030	-0.5899	-4.1088	3.5189
3	15	0.1397	0.00987	0.00909	-0.5899	-4.3231	3.7332

Appendix C. AVTIE Produced Spanwise Aerodynamic Performance

This Appendix lists AVTIE output results for individual wing strip sections used for the Roskam/AVTIE strip method. Panel 0, 1, 2 and 3 refer to the forward inside wing, aft wing, joint section, and outboard wing sections. These panels are then subdivided into strips, varying for each panel.

C.1 AVTIE Output For Mission Point 4, Method 1 In Figure 26

panel-id	strip-id	CL-2d	CD-t-2d	CD-i-2d	strip-area	strip-aoa
0	0	6.5175e-1	5.5141e-2	4.8411e-2	2.8868e+0	3.7093e+0
0	1	7.1054e-1	5.3209e-2	2.0480e-2	6.7117e+0	3.7093e+0
0	2	7.5834e-1	5.2695e-2	5.4971e-3	1.0183e+1	3.7093e+0
0	3	7.9085e-1	5.2208e-2	1.2372e-3	1.0183e+1	3.7093e+0
0	4	8.0830e-1	5.2078e-2	-1.2415e-3	1.0183e+1	3.7093e+0
0	5	8.6273e-1	4.8775e-2	3.3579e-3	1.0183e+1	3.7093e+0
0	6	6.4816e-1	5.4768e-2	-2.5561e-2	1.0183e+1	3.7093e+0
0	7	1.4222e-1	3.3994e-2	-9.0066e-2	2.9949e+0	3.7093e+0
1	0	5.5022e-1	1.9936e-2	-7.4226e-3	2.6642e+0	-2.0730e-1
1	1	5.2984e-1	2.1832e-2	1.3264e-3	6.0449e+0	-1.2450e-2
1	2	5.4146e-1	2.6005e-2	9.2411e-3	9.1563e+0	5.1726e-1
1	3	5.6199e-1	3.2680e-2	1.3648e-2	9.1544e+0	1.1837e+0
1	4	5.7461e-1	3.8796e-2	1.9326e-2	9.1542e+0	1.8492e+0
1	5	5.5304e-1	4.4517e-2	2.9002e-2	9.1550e+0	2.5157e+0
1	6	6.3882e-1	5.1547e-2	1.4685e-2	9.1571e+0	3.1809e+0
1	7	8.7479e-1	4.6101e-2	-4.0037e-2	2.6936e+0	3.6107e+0
2	0	5.3339e-1	5.5435e-2	-3.8156e-2	3.0900e+0	3.7093e+0
2	1	5.9421e-1	5.7690e-2	1.8711e-2	2.8210e+0	3.7093e+0
2	2	6.2590e-1	5.6749e-2	2.6210e-2	2.5523e+0	3.7093e+0
2	3	6.7925e-1	5.5309e-2	2.9444e-2	2.2837e+0	3.7093e+0
3	0	9.2979e-1	4.3563e-2	3.4021e-2	1.7077e+0	3.7093e+0
3	1	9.0172e-1	4.5580e-2	2.5348e-2	1.5748e+0	3.7093e+0
3	2	8.9928e-1	4.5199e-2	1.3525e-2	1.5745e+0	3.7093e+0
3	3	8.9170e-1	4.7138e-2	9.5295e-3	1.5745e+0	3.7093e+0
3	4	8.8341e-1	4.5844e-2	7.5308e-3	1.5747e+0	3.7093e+0
3	5	8.7421e-1	4.7515e-2	6.5906e-3	1.5745e+0	3.7093e+0
3	6	8.6288e-1	4.8798e-2	6.1601e-3	1.5748e+0	3.7093e+0
3	7	8.4899e-1	4.9631e-2	5.8851e-3	1.5745e+0	3.7093e+0
3	8	8.3036e-1	4.9739e-2	5.4426e-3	1.5745e+0	3.7093e+0
3	9	8.0450e-1	5.1500e-2	4.5317e-3	1.5747e+0	3.7093e+0
3	10	7.6451e-1	5.3625e-2	2.9776e-3	1.5745e+0	3.7093e+0
3	11	7.0582e-1	5.4985e-2	-5.6206e-3	1.5745e+0	3.7093e+0
3	12	5.8504e-1	5.7632e-2	-1.2800e-2	2.1876e+0	3.7093e+0
3	13	5.6550e-1	5.7945e-2	-1.2411e-2	7.2929e-1	3.7093e+0
3	14	4.2596e-1	5.1938e-2	-1.1264e-2	7.2916e-1	3.7093e+0
3	15	4.9079e-1	5.5473e-2	1.0184e-2	7.2898e-1	3.7093e+0

C.2 AVTIE Output For Mission Point 4, Method 2 In Figure 26

panel-id	strip-id	CL-2d	CD-t-2d	CD-i-2d	strip-area	strip-aoa
0	0	5.0646e-1	3.9362e-2	3.6273e-2	2.8868e+0	1.8486e+0
0	1	5.5550e-1	3.7987e-2	1.3641e-2	6.7117e+0	1.8486e+0
0	2	5.9230e-1	3.7151e-2	2.9677e-3	1.0183e+1	1.8486e+0
0	3	6.1711e-1	3.8139e-2	6.1621e-4	1.0183e+1	1.8486e+0
0	4	6.3021e-1	3.7426e-2	-8.8556e-4	1.0183e+1	1.8486e+0
0	5	6.7177e-1	3.5090e-2	2.0206e-3	1.0183e+1	1.8486e+0
0	6	5.0166e-1	3.8774e-2	-1.6512e-2	1.0183e+1	1.8486e+0
0	7	1.0274e-1	2.5960e-2	-5.7363e-2	2.9949e+0	1.8486e+0
1	0	4.3205e-1	1.5572e-2	-3.9220e-3	2.6568e+0	-1.1965e+0
1	1	4.1650e-1	1.7072e-2	2.4357e-3	6.0453e+0	-1.0441e+0
1	2	4.2726e-1	1.9427e-2	8.1706e-3	9.1574e+0	-6.3259e-1
1	3	4.4355e-1	2.3259e-2	1.0788e-2	9.1563e+0	-1.1432e-1
1	4	4.5299e-1	2.8216e-2	1.4256e-2	9.1563e+0	4.0290e-1
1	5	4.3872e-1	3.2263e-2	1.9486e-2	9.1567e+0	9.2008e-1
1	6	5.1435e-1	3.4867e-2	9.6489e-3	9.1582e+0	1.4371e+0
1	7	6.9948e-1	3.0019e-2	-1.5193e-2	2.6937e+0	1.7718e+0
2	0	4.0954e-1	3.6924e-2	-2.4771e-2	3.0900e+0	1.8486e+0
2	1	4.5882e-1	3.8750e-2	1.2103e-2	2.8210e+0	1.8486e+0
2	2	4.9149e-1	3.8015e-2	1.6748e-2	2.5523e+0	1.8486e+0
2	3	5.3283e-1	3.8001e-2	1.9109e-2	2.2837e+0	1.8486e+0
3	0	7.2765e-1	2.9121e-2	2.4699e-2	1.7077e+0	1.8486e+0
3	1	7.0559e-1	3.1670e-2	1.8148e-2	1.5748e+0	1.8486e+0
3	2	7.0445e-1	3.1535e-2	9.0723e-3	1.5745e+0	1.8486e+0
3	3	6.9901e-1	3.0901e-2	6.3743e-3	1.5745e+0	1.8486e+0
3	4	6.9311e-1	3.2735e-2	5.1465e-3	1.5747e+0	1.8486e+0
3	5	6.8649e-1	3.2086e-2	4.6119e-3	1.5745e+0	1.8486e+0
3	6	6.7814e-1	3.3499e-2	4.3718e-3	1.5748e+0	1.8486e+0
3	7	6.6770e-1	3.4765e-2	4.2027e-3	1.5745e+0	1.8486e+0
3	8	6.5342e-1	3.3646e-2	3.8544e-3	1.5745e+0	1.8486e+0
3	9	6.3321e-1	3.5546e-2	3.1001e-3	1.5747e+0	1.8486e+0
3	10	6.0130e-1	3.8304e-2	1.6665e-3	1.5745e+0	1.8486e+0
3	11	5.5393e-1	3.7795e-2	-5.7598e-3	1.5745e+0	1.8486e+0
3	12	4.5135e-1	3.9418e-2	-1.0137e-2	2.1876e+0	1.8486e+0
3	13	4.3289e-1	3.8614e-2	-1.6675e-2	7.2929e-1	1.8486e+0
3	14	3.1357e-1	3.6636e-2	-1.6350e-2	7.2916e-1	1.8486e+0
3	15	3.2294e-1	3.6497e-2	-4.5800e-3	7.2898e-1	1.8486e+0

Figure 38. AVTIE User Interface Menu

Bibliography

1. Moorhouse, D., and others. "Sensorcraft – Phase I," Air Vehicles technology assessment, March 2000.

2. Roskam, J., "Airplane Design – Part VI: Preliminary Calculation of Aerodynamic, Thrust and Power Characteristics," Ottawa, Kansas, 1990.

3. "MATLAB Version 6.5.0.180913a, Release 13", The Math Works, Inc., 2002.

4. "Adaptive Modeling Language Basic Training Manuel: Version 2.07," Technosoft Incorporated, 2001.

5. Blair, M., Canfield, R., and Roberts, R., "Joined-Wing Aeroelastic Design With Geometric Non-Linearity," AIAA IFASD 2003, presented at the International Forum on Aeroelasticity and Structural Dynamics, Amsterdam, Netherlands, 2003.

6. Drela, M., Youngren, H., "XFOIL 6.94 User Guide," Massachusetts Institute of Technology, 2001.

7. "User's Guide – PAN AIR Technology Program for Solving Potential Flow about Arbitrary Configurations," Public Domain Aeronautical Software, 1992.

8. Gern, F.H., Ko, A., Sulaeman, E., Gundlach, J.F., Kapania, R.K., Haftka, R.T., "Multidisciplinary Design Optimization of a Transonic Commercial Transport with Strut-Braced Wing," *AIAA Journal of Aircraft*, Vol. 38, No. 6, 2001, pp 1006-1014.

9. Miranda, L. R., "Boxplane Wing and Aircraft," U.S. Patent 3,834,654, Sept. 1974.

10. Wolkovich, J., Joined Wing Aircraft, U.S. Patent 3,942,747, March 1976.

11. Wolkovich, J., "The Joined-Wing: An Overview," *AIAA Journal of Aircraft*, Vol. 23, No. 3, 1986, pp. 161-178.

12. Zimmer, "Airplane with two superposed wings," U.S. Patent 4,090,681, May 1978.

13. Samuels, M. F., "Structural Weight Comparison of a Joined Wing and a Conventional Wing," *AIAA Journal of Aircraft*, Vol. 19, No. 6, 1982, pp. 485-491.

14. Hajela, P. and Chen, J. L., "Optimum Structural Sizing of Conventional Cantilever and Joined-Wing Configurations Using Equivalent Beam Models," In AIAA/AHS/ASEE Aircraft Systems, Design and Technology Meeting, Dayton, Ohio, Oct. 1986, AIAA Paper 86-2653.

15. Hajela, P., "Reduced Complexity Structural Modeling for Automated Airframe Synthesis," NASA CR 177440, May 1987.

16. Miura, H., Shyu, A., and Wolkovitch, J., "Parametric Weight Evaluation of Joined Wings by Structural Optimization," *AIAA Journal of Aircraft*, Vol. 25, No. 12, 1988, pp. 1142-1149.

17. Frediani, "Large Dimension Aircraft," U.S. Patent 5,899,409, May 1999.

18. Fairchild, M.P. "Structural Weight Comparison of a Joined Wing and a Conventional Wing", AIAA-81-0366, presented at the 19th AIAA Aerospace Sciences Meeting, Reno, NV, 12-15 January 1981.

19. Smith, S.C., Cliff, S.E., and Kroo, I.M., "The Design of a Joined-Wing Flight Demonstrator Aircraft", AIAA Paper 87-2930, AIAA/AHS/ASEE Aircraft Design, Systems and Operations Meeting, St. Louis, MO, 14-16 September 1987.

20. Lin, H-H., Jhou, J., and Stearman, R., "Influence of Joint Fixiti on the Aeroelastic Characteristics of a Joined Wing Structure," AIAA Paper 90-0980, Proceedings of the 31st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Long Beach, CA, April 1990, pp. 1442-1454.

21. Gallman, J. W., Kroo, I.M., and Smith, S.C. "Design Synthesis and Optimization of Joined-Wing Transports", AIAA-90-3197, presented at the AIAA/AHS/ASEE Aircraft Design, Systems and Operations Meeting, Dayton, OH, 17-19 September 1990.

22. Kroo, I.M., Gallman, J.W., and Smith, S.C., "Aerodynamic and Structural Studies of Joined-Wing Aircraft," *AIAA Journal of Aircraft*, Vol. 28, No. 1, January-February 1991, pp. 75-81.

23. Weisshaar, T.A., and Lee, D.H., "Aeroelastic Tailoring of Joined-Wing Configurations," AIAA-2002-1207, presented at the 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Denver, CO, 22-25 April 2002.

24. Livne, E., "Aeroelasticity of Joined-Wing Airplane Configurations: Past Work and Future Challenges – A Survey", AIAA-2001-1370, presented at the 42nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Seattle, WA, 16-19 April 2001.

25. Blair, M., and Canfield, R., "A Joined-Wing Structural Weight Modeling Study," AIAA-2002-1337, presented at the 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Material Conference, Denver, CO, 22-25 April 2002.

26. Roberts, R.W., *Sensor-Craft Analytical Certification*. MS thesis, Graduate School of Engineering, Air Force Institute of Technology (AETC), Wright-Patterson AFB OH, March 2003. AFIT/GAE/ENY/03-06.

27. Smallwood, B.P., *Structurally Integrated Antennas on a Joined-Wing Aircraft*. MS thesis, Graduate School of Engineering, Air Force Institute of Technology (AETC), Wright-Patterson AFB OH, March 2003. AFIT/GAE/ENY/03-07.

28. Rasmussen, C.C., *Optimization Process for Configuration of Flexible Joined-Wing*. MS thesis, Graduate School of Engineering, Air Force Institute of Technology (AETC), Wright-Patterson AFB OH, March 2004. AFIT/GAE/ENY/04-M14.

29. Sitz, J.J., *Aeroelastic Analysis of a Joined-Wing Sensorcraft*. MS thesis, Graduate School of Engineering, Air Force Institute of Technology (AETC), Wright-Patterson AFB OH, June 2004. AFIT/GAE/ENY/04-J12.

30. Snyder, R.D., Hur, J.Y, Strong, D.D., Beran, P.S., "Aeroelastic Analysis of a High-Altitude Long-Endurance Joined-Wing Aircraft," AIAA-2005-1948, presented at the 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Material Conference, Austin, TX, 18-21 April 2005.

31. Saarlas, M., "Aircraft Performance," Department of Aerospace Engineering, United States Naval Academy, 2003.

Vita

Ensign Ryan L. Craft was raised in Shelby, Ohio, and graduated from Shelby High School in 1999. Under sponsorship of the United States Naval Academy Foundation, he attended one year at the Western Reserve Academy preparatory school in Hudson, Ohio. In April of 2000, he was appointed to the United States Naval Academy class of 2004, where he graduated with a Bachelor of Science degree in Aerospace Engineering and earned a commission in the United States Navy on May 28th, 2004.

In June of 2004 he entered the Graduate School of Aeronautical Engineering at the Air Force Institute of Technology in Dayton, Ohio. Upon graduation in June of 2005, he will report to Pensacola Naval Air Station in Pensacola, Florida to begin flight training as a Naval Aviator.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
13 – 06 – 2005	**Master's Thesis**	Jun 2004 – Jun 2005

4. TITLE AND SUBTITLE

Drag Esimates for the Joined-Wing Sensor Craft

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Craft, Ryan L., Ensign, USN

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GAE/ENY/05-J02

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFRL/VASD
Attn: Dr. Maxwell Blair
2210 Eighth St. Rm. 220
Wright Patterson AFB, OH 45433 Phone (937) – 255 – 8430

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This research studied the drag effects of the joined-wing sensor craft technology demonstrator being developed at the Air Force Research Laboratory. Although many performance parameters have been studied and evaluated for this vehicle, to date no detailed drag estimates have been conducted for the AFRL configuration. Previous performance parameters of the aircraft have been estimated based solely on a constant lift-to-drag ratio assumption. Using the Air Vehicles Technology Integration Environment created by Dr. Maxwell Blair, and supplemented by MATLAB code, this study explored three different drag prediction methods to determine accurate estimates of both parasite and induced drag. The Roskam/AVTIE Pan Air method was determined as the best approach to estimate drag by measuring parasite drag effects using XFOIL, a respected environment within the aviation industry to accurately predict all viscous drag effects, and determined induced drag from Pan Air, a creditable software package based on inviscid flowfield solutions about three dimensional objects. This method will be incorporated into a single design environment, in conjunction with AVTIE, in order to estimate drag and aid future AFRL joined-wing design studies incorporating wing twist, aeroelastic effects, and other geometric changes to the baseline configuration.

15. SUBJECT TERMS

Sensor Craft, Joined-Wing, Drag Estimates, Air Vehicles Technology Integration Environment (AVTIE), Pan Air, XFOIL

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
REPORT	**ABSTRACT**	**c. THIS PAGE**			Dr. Robert Canfield
U	**U**	**U**	**UU**	151	**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-6565, ext 4641; e-mail: robert.canfield@afit.edu

Standard Form 298 (Rev: 8-98)
Prescribed by ANSI Std. Z39-18