Research Summary - Year Two
for Contract NAG-1-1662

DUAL-MISSION LARGE AIRCRAFT
FEASIBILITY STUDY AND AERODYNAMIC INVESTIGATION

Dr. Dimitri Mavris
Principal Investigator

March 28, 1997

Aerospace Systems Design Laboratory
School of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0150
http://www.cad.gatech.edu/asdl

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ALCCA	Aircraft Life Cycle Cost Analysis
ANOVA	Analysis of Variance
AR	Aspect Ratio
ASDL	Aerospace Systems Design Laboratory
BDAP	Boeing Design and Analysis Program
C_{lmax}	Maximum Lift Coefficient
CONUS	CONtinental United States
DOC	Direct Operating Costs
DoE	Design of Experiments
DMLA	Dual Mission Large Aircraft
FAA	Federal Aviation Administration
FLOPS	Flight Optimization System
FPS3D	Full Potential Solver, 3-Dimensional
FY	Fiscal Year
GRA	Global Reach Aircraft
GPS	Global Positioning System
GUI	Graphical User Interface
HLFC	Hybrid Laminar Flow Control
IHPTET	Integrated High Performance Turbine Engine Technology
IOC	Indirect Operating Costs
IPPD	Integrated Product and Process Development
KB	Kilobytes
LMASC	Lockheed-Martin Aeronautical Systems Company
LdgFL	Landing Field Length
L/D	Lift-to-Drag ratio
NLF	Natural Laminar Flow
$/RPM	Required Average Yield per Revenue Passenger-Mile
RAM	Rapid Aircraft Modeler
RDS	Robust Design Simulation

RDT&E	Research, Development, Testing, and Evaluation
RSE	Response Surface Equation
RSM	Response Surface Methodology
SAB	Scientific Advisory Board
SFC	Specific Fuel Consumption
SHT	Horizontal Tail Area
STARS	Simplifying Time by Automating the Response Surfaces
SVT	Vertical Tail Area
SW	Wing Area
t/c	Thickness-to-Chord ratio
TCA	Wing Thickness-to-Chord ratio
TCM	Tailored Cost Model
TOC	Total Operating Costs
TOFL	Take-off Field Length
TOGW	Take-off Gross Weight
TWR	Thrust-to-Weight ratio
USAF	United States Air Force
VAPP	Approach speed
VCMN	Cruise Mach Number
VLT	Very Large Transport

Executive Summary

For the second time since its inception, The United States Air Force Scientific Advisory Board (SAB) published its review of the service's future needs and present shortcomings in terms of aircraft technology and mission fulfillment. This year, the SAB identified global mobility as a key area in which the present Air Force fleet possesses a shortfall. The current aging transport aircraft in active use lack the performance, availability, and affordability needed to effectively fly the needed missions of today and tomorrow. In assessing possibilities for new aircraft systems, the SAB gave top priority to the development of a Global Reach Aircraft, or GRA. The GRA embodies the SAB's vision of a large, subsonic transport capable of transporting 150,000 pounds of payload over an unrefueled range of 12,000 nautical miles.

The combination of explosive air passenger growth, shrinking traffic handling capability at existing airports, and recent financial difficulties in the airline industry have combined to necessitate the development of a VLT, or Very Large Transport. These first two realities mean an increase in the number of air passengers, yet fewer air routes and terminal gates to handle the surge in travel. The third bears with it added pressure on the airlines to maintain affordable ticket prices while achieving a satisfactory return on their investment. Based on studies performed in industry and at the NASA Langley Research Center, a VLT aircraft would be an advanced, dual-deck, 800-passenger subsonic transport capable of affordable operations across 7500 nautical mile international routes. The large passenger capacity allows for the transportation of more passengers with fewer aircraft, thereby easing air route congestion and airport crowding. The VLT's intended affordability should bring about ticket prices as much as 30% less than those for the Boeing 747-400, currently the world's largest passenger aircraft, while yielding sound airline profits.

As part of a two-year study under contract from the NASA Langley Research Center, the Aerospace Systems Design Laboratory (ASDL) at the Georgia Institute of Technology is currently developing the Dual-Mission Large Aircraft concept. A Dual-Mission Large Aircraft, or DMLA, represents the possibility of a single aircraft capable of fulfilling both the GRA and VLT roles. The DMLA, by combining the GRA and VLT into a single new aircraft, could possibly lower the aircraft manufacturer's production costs through the resulting increase in production quantity. This translates into lower aircraft acquisition costs, a primary concern for both the Air Force and commercial airlines.

This report outlines the first steps taken in this study, namely the assessment of technical and economic feasibility of the DMLA concept. In the course of this project, specialized GRA and VLT aircraft were sized for their respective missions, using baseline conventional (i.e., lacking advanced enabling technologies) aircraft models from previous work for the Air Force's Wright Laboratory and NASA-Langley. DMLA baseline aircraft were then also developed, by first sizing the aircraft for the more critical of the two missions and then analyzing the aircraft's performance over the other mission. The resulting aircraft performance values were then compared to assess technical feasibility. Finally, the life-cycle costs of each aircraft (GRA, VLT,

and DMLA) were analyzed to quantify economic feasibility. These steps were applied to both a two-engine aircraft set, and a four-engine aircraft set.

The GRA configuration used in this study is based on a concept submitted by Lockheed-Martin in response to the SAB's study, which ASDL modeled and analyzed. This aircraft design was a fixed point design leaving little room for adaptation to the VLT mission. From an aerodynamics point of view, the point design was expanded to a design space through the use of Response Surface Methodology and Design of Experiments. It was intended that the space created would capture the VLT requirements and fulfill the aspirations of the DMLA. The design space was created by extending the fixed geometric characteristics of the current GRA to a range of values and analyzing the impact on the aerodynamic and system level performance characteristics. In addition, advanced technologies were infused to the baseline aircraft to create eight feasible configurations. Each configuration was compared to the baseline and the impact of adding new technologies was quantified.

The result was a set of sized two-engine and four-engine aircraft configurations for the GRA, VLT, and DMLA roles. Performance results were obtained from the optimization of the configurations, as were economics values from the subsequent life-cycle costs analyses. In the end, successful synthesis and sizing of DMLA aircraft demonstrated technical feasibility of the concept, and the results of the life-cycle cost analysis showed economic viability as well. Additionally, the aircraft geometry variables most significantly impacting aircraft performance were identified and quantified in terms of their effect. The results were used to define a DMLA geometric design space, and assess the impact of enabling technologies on the DMLA.

1. Background

Developments in recent years have prompted the Aerospace industry to focus on aspects other than performance as a means of evaluating an aircraft's feasibility. Industry must now consider the risk of new technologies, the affordability of the aircraft, manufacturing of new materials, etc. Therefore, it is becoming increasingly important that the design methodology utilized in future systems reflect this new focus. Historically, a sequential and deterministic approach to design, applying a top-down decomposition systems engineering methodology, was utilized. Yet, this approach did not account for the manufacturing, scheduling, or economics of the aircraft. The focus of late has been toward the Integrated Product and Process Development (IPPD) approach. IPPD promotes the integration of manufacturing processes and affordability to the aircraft design disciplines. It encourages bringing more knowledge to the conceptual stages, while maintaining design freedom as the design cycle progresses. As a result, the design space and budget are not fixed or committed early in the design process.

The Aerospace Systems Design Laboratory (ASDL) has recognized the changing philosophy in industry and has developed a systematic approach to design. This unique methodology not only addresses the interdisciplinary interactions of design, but also the integration of design of manufacturing to support the IPPD environment. This is achieved using Robust Design Simulation (RDS). ASDL was founded in 1992 to support activities in this area. Since that time, ASDL has promoted and further enhanced the concept of the IPPD approach to design. Even though the RDS method encompasses the design disciplines, such as aerodynamics, structures, and propulsion, and the affordability and supportability issues, this paper focuses on the aerodynamic aspects and the implementation of ASDL's IPPD methodology. This proof of concept is applied to the Dual-Mission Large Aircraft (DMLA) concept and can be utilized to assist industry and NASA efforts in attempting to develop, quantify, and evaluate the metrics necessary for the feasibility of a joint military/commercial transport.

1.1. Military Transport Need

For the second time since its inception, the United States Air Force (USAF) Scientific Advisory Board (SAB) researched the future needs and present shortcomings in the Air Force's overall mission effectiveness. The results of their findings are compiled in a series of collected volumes, known as the New World Vistas. This document presents these findings, and identifies revolutionary aircraft concepts and associated enabling technologies that could ultimately make the Air Force both more effective and cost-efficient. The analyses and conclusions contained in the New World Vistas encompass the following assumptions (Summary 1-2):

- Future Air Force operations will be staged directly from the CONUS, or Continental United States. This comes in light of further defense spending cutbacks, which are expected to continue base closings worldwide. From an aircraft design standpoint, this assumption becomes the need for greater range and improved fuel consumption.

- The Air Force must be prepared to conduct airlift operations anywhere in the world on short notice. Future conflicts will be smaller, localized to various remote regions of the

world, where tensions may flare at any time. This necessitates aircraft with greater reliability and operational availability.

- System affordability and capability are equally important. Effective mission completion requires more than high performance aircraft; new aircraft must also be affordable, or else the Air Force could not could not afford to conduct all possible missions. Thus, life-cycle cost metric must enter early into the design of any new aircraft.

In all operations - peacetime and wartime - mobility is a limiting factor. Airlift operations require aircraft and personnel to enter extremely dangerous regions, where it may not be possible to provide protection for airlifters or conduct responses to attacks (Summary 29). In addition, the ever-changing world environment brings to the Air Force the very real possibility of the need to simultaneously supply large forces in widely divergent points around the globe (Air 119). The increasingly likely assumption that continued austerity will force base closings around the globe further exacerbates the problem (Summary 1).

Airlift provides the speed and flexibility to deploy and sustain combat forces. As an example, had the 120 C-17 Globemaster III transports planned for procurement been obtained sooner, airlift during Desert Storm could have been conducted 20%-35% faster; during the first twelve days alone, enough cargo could have been deployed amounting to twelve additional fighter squadrons (Air 126). Internal Air Force studies set the present airlift need at 49-52 million ton-miles per day, during such crisis situations as the conflict in the Baltics (Mobility 4). Present mobility system capacity falls well short of this requirement, necessary to support existing forces, even with the inclusion of the Civil Reserve Air Fleet, or CRAF (Summary 29). These results are shown below in Figure 1.1-1, which depicts the breakdown of airlift ton-miles per aircraft per year. Heavy dependence upon the CRAF is all too evident; the "CRAF III" component is a hypothetical, last-resort use of all available commercial aircraft. Figure 1.1-2 depicts the C-5, C-141, and C-17 aircraft, the backbone of the Air Force's transport capability.
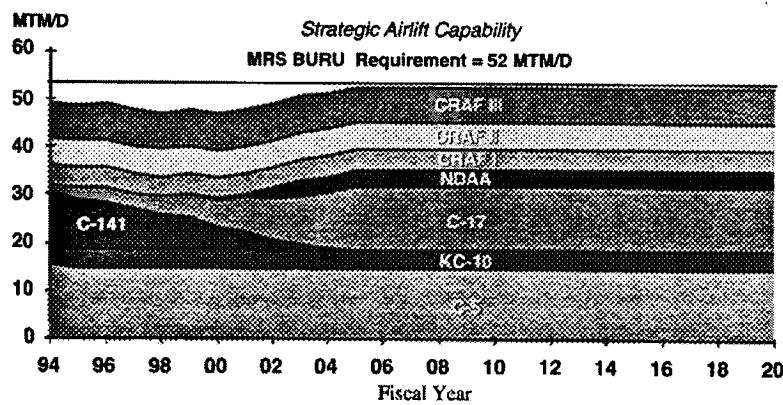
Figure 1.1-1. Breakdown of present airlift capability.
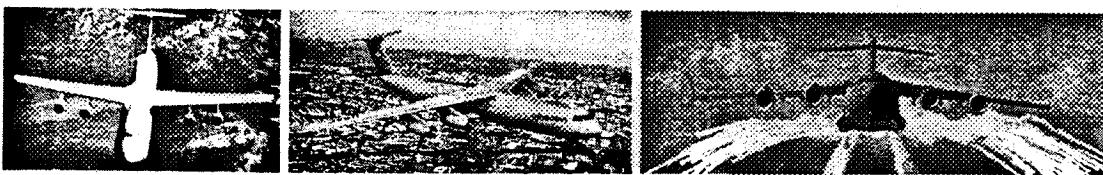
Figure 1.1-2. C-5, C-141, and C-17 transport aircraft.

The Air Force's transport fleet not only lacks strength in numbers, the few aircraft in service are lacking in capability. The **C-5 Galaxy**, before the entry of the C-17 into service, was the sole large aircraft capable of transporting outsize Army cargo (Wilkinson 30). Poor austere field performance is evident from its 12,200 foot takeoff field length and 4900 foot landing field length (C-5 2). Its operational availability of 67% in Desert Storm is unacceptable for future rapid deployments (Mobility 6). The aircraft spends much of its time on the ground, receiving as much as 60 maintenance man-hours for each flight-hour (Wilkinson 32). These maintenance-intensive operations are partially caused by the vast array of complex FEBA (Forward Edge of the Battle Area) equipment carried aboard, which is presently never used - this aircraft is far too valuable too risk exposure to the dangers of FEBA operations (Wilkinson 37). Overall mission performance is rather poor as well; the cruise Mach number is only 0.72, far less than commercial aircraft, and the ferry range is a mere 5165 nautical miles (C-5 3).

The **C-141 Starlifter** is a tactical, airdrop-capable transport with aeromedical capability (C-141 1). Despite its many roles, the C-141 must soon be retired, as it is an aging aircraft first introduced into service in 1964 (Mobility 5). An extremely low cruise Mach number of 0.66 and short range of 2200 nautical miles results from the operational use of this older technology (C-141 2). Although requiring less maintenance than the C-5, the C-141 was operationally available only 78% of the time - still not enough for the rapidly changing world of tomorrow (Mobility 6).

The **C-17 Globemaster III** is the latest addition to the transport fleet, having been first introduced in 1993 (Air 119). It features better performance than its predecessors, in terms of higher cruise speed (Mach number 0.77) and longer range (5200 nautical miles with a payload of 130,000 pounds) (C-17 2). Furthermore, payload bay flexibility makes the aircraft suitable for the different missions required in the Air Force, described later (C-17 1). The combination of needs for front-line operations and improved reliability were also addressed, as the C-17 has thus far demonstrated 82.5% operational availability, requiring only 18.6 maintenance man-hours per flight-hour (C-17 1). Unfortunately, Congressional budget cutting may mean too few of these aircraft purchased for service to take up the transport capability (Mobility 4).

According to the New World Vistas, future airlifters will be required to fly missions in each of the following four areas (Mobility 5-6):

- Airlift of personnel. Present shortfalls in transport capability would require 90% of all personnel in a large-scale contingency to travel via CRAF.

- Rapid deployment of troops, supplies, and equipment. The increasingly unpredictable nature of tomorrow will require deployments on extremely short notice.

- Aeromedical evacuation. Currently, as the C-141 is retired from service, the C-17 and CRAF form nearly all of the current aeromedical transportation capability. However, C-

17s are still few in number, and the CRAF cannot begin to fulfill wartime needs. Furthermore, CRAF aircraft suffer from slow patient on/offloading, and lack the capability to support several pieces of medical equipment.

- <u>Humanitarian aid</u>. Whether by natural disaster or military and political aggression, the lives of innocent people forever will be jeopardized, sometimes due to conflicts outside the scope of the United States' foreign policy. The likely decimation of ground infrastructure forces aerial transportation of relief supplies or evacuation of civilians.

The Mobility Volume of the <u>New World Vistas</u> (p. 11) prioritizes the development of the following aircraft in response to the needs of these missions, and to fill the gaps in the present capability of fulfilling these missions:

1. Global Reach Aircraft
2. Supersonic Military Transport
3. Ground-Effect Transport

First priority is given to the development of a Global Reach Aircraft, or GRA. To complete the missions outlined above, such an aircraft must posses the following:

- The ability to fly 12,000 nautical miles, deliver cargo, and continue on to a terminal refueling point *without refueling*. Aerial refueling is a logistics-intensive operation; long range will remove transport dependence on the refueling fleet (Summary 30-33). Also, global range is the key to reaching any point in the world nonstop (Mobility 1), addressing the need to stage missions primarily from the CONUS. Lockheed-Martin Aeronautical Systems Company, in pursuing GRA development, revised this range to 7500 nautical miles, on the grounds that any conceivable location on the globe can be reached from the United States East or West Coast within this range (Mission 1). Figure 1.1-3 below illustrates the profile of a typical GRA mission.
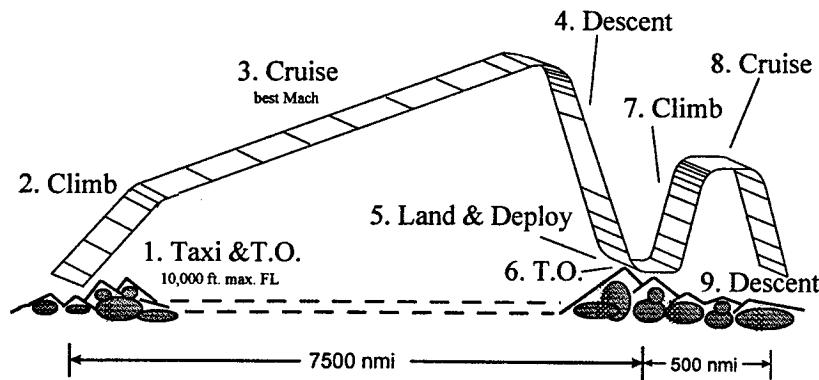
Figure 1.1-3. GRA mission profile.

- A payload capacity of 150,000 pounds, while holding takeoff gross weight (TOGW) under 1,000,000 pounds (Summary 30-33).

- The infusion of new aerodynamic technologies, to greatly improve cruise lift-to-drag ratio (L/D) over existing aircraft (Summary 30-33).

- A 20% increase in propulsive efficiency (Summary 30-33).

- All-weather operation, by utilizing a GPS system more resistant to enemy signal jamming (Summary 30-33).

- Point-of-use delivery capability. Items shipped spend enormous amounts of time on the ground. Also, some landing fields are austere, and place the aircraft in danger of attack. Furthermore, ground transport from more remote fields not only further delays shipping, it also places the items shipped in greater danger of attack (Summary 30-33).

- Use of improved protection systems, to reduce dependency on fighter escorts. If possible, improved ECM protection systems are needed (Summary 30-33).

- Improved survivability - an especially important aspect, given the proliferation of missiles to third-world nations (Mobility 9).

- Improved reliability and maintainability. The C-5, the only aircraft in service capable of transporting outsize cargo, suffers from especially poor reliability and maintainability (Mobility 4). Furthermore, during Desert Storm, C-141 availability started at 87% and fell to 78%; C-5 availability started at 79% and fell to 67%. Sustained airlift of new materiel and reinforcement of existing materiel in the future requires operational availability in excess of 90% (Mobility 41).

- A higher cruise speed. Again, future conflicts can flare up at any time; faster airlifters would shorten reaction times. Furthermore, military aircraft are currently too slow to use commercial air routes, forcing longer tracks across the globe to given destinations. Thus, cruise Mach numbers in excess of 0.80 are a necessity (Mobility 4).

- Finally, any new system must be affordable. The need to improve effectiveness in light of declining military spending necessitates lower acquisition, operations, and support costs (Mobility 41).

Carrying 150,000 pounds of payload over 7500 nautical miles unrefueled, the global range of a GRA would provide great flexibility in mobility operations; all refueling assets (air and ground) that would be needed otherwise can now be refocused on other missions. Reliability can be improved by using proven, commercial aircraft subsystems, resulting in greater aircraft availability; this in turn reduces the number of aircraft needed. If this aircraft were developed commercially first, then adapted to this military role, the aircraft's affordability would be vastly increased (Mobility 12). Furthermore, global range supports CONUS-based power projection (Mobility 37).

1.2. Commercial Transport Need

In recent times, airlines worldwide have fallen on hard financial times, and in an age of satellite communications, computer networking, and electronic mail, many feel that long range travel may not be needed. Contrary to this somewhat pessimistic perception, recent surveys predict that air travel will double by the year 2005. This growth will be especially large in the Asian-Pacific markets, where economic analysts predict this region to be the air transport market for the next twenty years (Kirby 1).

As a result of the increased traffic, airport congestion will reach unbearable levels without considerable expansion of existing airports or construction of new ones. Added to this are the problems of limited government financing and environmental group opposition, which hamper airport construction of expansion. Thus, the increased congestion, along with the predicted growth over the coming years, has pointed to the need for a high capacity, long range aircraft that can meet the increased travel demand as well as maximize landing and takeoff slot utilization at existing airports (Mecham). For example, gates at London's Heathrow Airport have been rated the most difficult to obtain due to crowding. In a recent Airbus survey, twelve airlines from Europe, the United States, and the Asian-Pacific region expressed a need for an airplane much larger than the 747-400 in the near future, capable of transporting between 600 and 1000 passengers. In fact, Upali Wickrama, the chief of forecasting and economic planning for the International Civil Aviation Organization, predicts that by 2015 there will be a demand for an additional 443 aircraft with 400-600 seats and 360 aircraft with greater than 600 seats (Lenorvitz). Based on economic viability studies performed in ASDL, an 800-passenger VLT proved to be the most profitable over a wider range of markets when compared to 600- and 1000-passenger VLT aircraft (Kirby 36).

Though these studies favorably show the need for a Very Large Transport (VLT), another prediction that deserves considerable attention is that air travel is expected to move from the business market to the more price sensitive tourist market. Since tourism focuses more on "luxury" than business travel, tourists will only be willing to travel abroad if it is affordable and comfortable. Consequently, airlines are looking for a 600 to a 1000 passenger airplane with an affordable ticket price for the passenger while maintaining a reasonable Return On Investment (ROI). As a result, the following goals were established for the development of the VLT concept:

- Achieve at least a 30% reduction in passenger ticket fare as compared to the Boeing 747-400;

- Achieve a high ROI for the airlines;

- Achieve a low aircraft unit cost to reduce the risk of investment for the airlines; and

- Minimize the number of aircraft required to meet the predicted market demand needs. This also would reduce the number of gates needed to serve a given airport.

Based on Airbus Market studies, and current long-range commercial transport aircraft, a VLT mission profile would resemble that shown below in Figure 1.2-1.

3. Cruise
M = 0.85

4. Descent

6. Reserve
14% res. fuel

2. Climb

1. Taxi &T.O.
11,000 ft. max. FL

5. Landing

|← 7500 nmi + added FAA 200 nmi →|
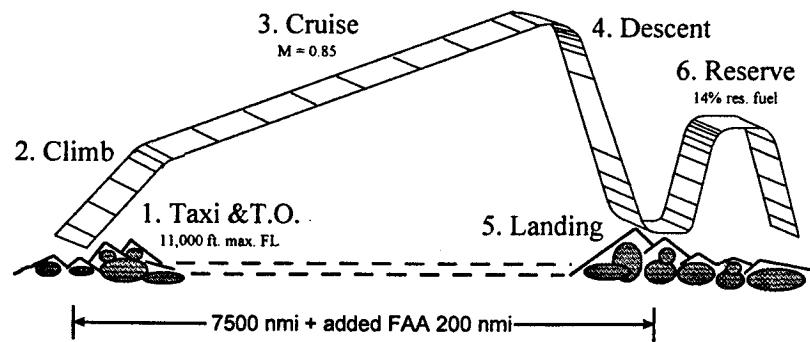
Figure 1.2-1. VLT mission profile.

2. The DMLA Solution

In assessing the needs of both military and commercial aircraft customers for a new large subsonic transport, it is conceivable that a single aircraft can be manufactured on the same assembly line, capable of fulfilling both the GRA and VLT missions. Lower acquisition cost, desirable to both military and commercial customers, is the primary benefit of producing a common aircraft on a single production line. This not only eliminates duplication of production facilities, it also decreases the per unit cost through the increase in the number of units produced. For example, as will be demonstrated later, it may be cheaper to produce a total of 900 DMLA aircraft - 500 VLT variants and 400 GRA variants, for instance - than to separately produce 500 VLT aircraft and 400 GRA aircraft with no commonality.

These notions form the genesis of the Dual-Mission Large Aircraft, or DMLA, concept. Figure 2-1 below shows VLT and GRA variants of a possible DMLA configuration.

VLT Variant   **GRA Variant**

Figure 2-1. Possible DMLA configuration and variants.

2.1. Feasibility Study Motivation and Objectives

ASDL, under contract to the NASA Langley Research Center, engaged in the DMLA study as a two-year research effort. The proposed research endeavors to develop a DMLA capable of fulfilling the above GRA and VLT missions. In the course of this effort, the differences in performance and economics of producing a DMLA will be quantified and compared against those for two separate, specialized aircraft.

The ultimate conclusion of this effort will furnish answers to the following questions:

1. Will a DMLA adequately fulfill both the VLT and GRA missions, or are two separate, specialized aircraft needed?

2. Are the manufacturing costs lower for a DMLA or a two-aircraft family?

3. Are the customer's (military and commercial) aircraft life-cycle costs less if they operate a DMLA, or if each operates a specialized aircraft?

4. How do the answers to the above questions change with the infusion of enabling new technologies?

5. What are the aerodynamic and structural characteristics and difficulties associated with a DMLA?

Ultimately, the conclusion of this effort will deliver a DMLA configuration capable of effectively fulfilling the given GRA and VLT missions, from both a performance and economics

standpoint. If a single DMLA proves to be impractical, the research must pursue answers to the above questions for a two-aircraft family (i.e., specialized GRA and VLT aircraft).

The DMLA effort draws upon work done under the following contracts and sponsors:

- <u>Notional Aircraft</u>, for the USAF Wright Laboratory. One result of this effort was a sized GRA configuration.

- <u>Very Large Transport study</u>, for the NASA Langley Research Center. NASA-Langley provided to ASDL several VLT configurations, including a sized 800-passenger conventional aircraft.

The sized aircraft configurations developed by the end of these efforts became the starting point for the research project undertaken and described here.

This research problem intended to determine the technical and economic feasibility of the DMLA concept. Comparison of conventional, specialized GRA and VLT aircraft against their DMLA-variant counterparts lies at the heart of the problem solution. The tasks described below form, on a preliminary level, the very first steps in the execution of the overall effort detailed previously. As the study progresses, the level of detail will increase from that of this research problem.

The following tasks were completed in this portion of the study:

- Analysis of the GRA and VLT mission profiles, from a synthesis and sizing point of view. The mission more critical to the sizing of a DMLA needed to be identified.

- Definition of initial designs - that is, the development and sizing of conventional GRA, VLT, and DMLA baseline configurations. Both two- and four-engine aircraft "sets" were developed, to add a dimension of comparison to this study.

- Performance optimization of all defined baseline aircraft.

- Analysis of resulting life-cycle costs for resulting configurations.

- Assessment of possible new technologies and alternate configurations that may be applicable to the remainder of the full two-year study.

The "Approach" chapter below details the rationale behind these steps.

"Conventional" implies an aircraft with no new technologies infused, i.e., an aircraft containing only those technologies utilized in practice to this day, or scheduled for utilization in the short term (within the next year). Conventional aircraft were modeled to provide the most equivalent bases for comparison, as new technologies could improve some aircraft more than others, making the given aircraft more favorable when, in actuality, the reverse may be true. For example, a conventional GRA may achieve better performance than a conventional DMLA variant; the identical application of some technology may benefit the DMLA more than the GRA, skewing the results for comparison of the two aircraft. Baseline aircraft must have the same performance starting point in order to fairly gauge technical feasibility.

2.2. Aerodynamic Study Motivation and Objectives

As stated previously, a feasible GRA design has been identified. This design is a fixed point and leaves little room for adaptation to the VLT mission requirements. Therefore, the point design needs to be expanded to a design space. With hope, the space would capture the VLT requirements and fulfill the aspirations of a DMLA. To extend this point design to a design space, the fixed geometric characteristics of the current GRA must be expanded to a range of values. As an example, the GRA currently has a wing aspect ratio; is this value the optimal? This topic raises a few questions:

1. Does the aspect ratio, or any other design variable, have to be fixed at its current value, or can it vary?

2. If the geometric characteristics deviate, what is the impact on performance and system level metrics?

3. Which geometric characteristics influence these parameters the most?

4. What physical limits must be imposed on those geometric characteristics?

The focus of this study was to respond to these questions and, hence, identify a feasible design space for the GRA. This space was defined by considering all of the geometric characteristics which influence the aerodynamic performance and system level performance of feasible GRA designs.

Since the vision of the DMLA is both commercial and military in scope, the needs of both customers must be addressed; most notably, the cruise Mach number. For commercial subsonic transport aircraft (e.g. A340, MD11, and B747-400)[11], a cruise Mach number of 0.82 to 0.85 is typical, yet, the current GRA capability is 0.78. For the DMLA to be a real possibility, this Mach number must be increased without extreme degradation in aircraft performance. Additionally, if the GRA cannot achieve a higher Mach number with conventional configuration, areas of possible advanced technology infusion must be identified. If technologies are needed, the impact on performance characteristics and system level objectives must be quantified.

To quantify the answers to the above questions, six system-level performance metrics were identified as objectives for this study:

1. Take-Off Gross Weight (TOGW)

2. Fuel weight

3. Empty weight

4. Wing weight

5. Block time, and

6. Research, Development, Testing, and Evaluation (RDT&E) costs.

Each objective was minimized by determining the optimum geometric characteristics of the wing and empennage. These objectives were subject to four constraints:

1. Approach speed (VAPP) less than 150 knots

2. Landing Field Length (LdgFL) less than 4,000 ft

3. Take-Off Field Length (TOFL) less than 10,000 ft, and

4. Aircraft unit acquisition price less than 200.0 million dollars (FY92)

For this study, three technologies were identified: advanced propulsive systems, hybrid laminar flow control on the wing, empennage, and nacelles, and use of composites on the empennage, nacelles, and fuselage. Each one of these technologies and their modeling will be explained in detail later.

3. Technical Feasibility Assessment

The overall approach to this study centered on validating the DMLA concept by first demonstrating technical feasibility, then assessing economic viability. Separate, specialized GRA and VLT aircraft were sized and optimized around their respective mission profiles; previous work into these two aircraft provided the baseline configuration "starting points." Then, DMLA variants of these aircraft were developed by analyzing the performance of the specialized aircraft with the more critical mission in the other aircraft's mission. Capability for the resulting DMLA aircraft to complete both missions, compared to the mission performance of the specialized GRA and VLT aircraft, then proves technical feasibility. Analysis of the economics of the GRA, VLT, and DMLA aircraft, and comparison of the results, illustrates economic viability.

3.1. Research Conducted

Research into previous work and existing aircraft was performed to initiate this study. The work completed on the precursor projects to this study, as detailed previously, was validated in terms of its applicability to the DMLA concept. Furthermore, the geometric and performance characteristics of the previously sized configurations were taken to be used as starting points for the aircraft modeled here. Additionally, this research furnished the previously described mission profiles, which were consistently used in this effort.

Aside from the New World Vistas, additional configuration information came through publicly available literature, both from periodicals and on the World Wide Web. (See References page for details.) Specifics on military design constraints, and shortcomings in existing aircraft, were provided though this task.

Additional literature was utilized to assess new technology possibilities, as well as alternate configurations of possible application to a DMLA design. Possible benefits of each were noted, and described later.

3.2. Tools Used

3.2.1. FLOPS: Flight Optimization System

FLOPS is a multidisciplinary system of computer programs used for the conceptual and preliminary design and analysis of aircraft configurations. Developed by the NASA Langley Research Center, FLOPS consists of several disciplinary modules (such as aerodynamics, weights, and propulsion), as well as a mission performance analysis module (McCullers 1).

The FLOPS program is most accurate for analyzing conventional, large subsonic transport aircraft, based on the nature of its disciplinary analysis modules. Each utilizes empirical relations derived from historical regressions of data for existing aircraft, which are primarily the conventional, large subsonic type.

Furthermore, FLOPS contains an internal gradient-based optimizer. Various aircraft design variables can be parametrically varied to minimize a performance objective function, described in detail later.

All aircraft sizing and analysis tasks for this study utilized FLOPS. This tool's use is valid for this study, as each of the aircraft in question is a large subsonic transport.

3.2.2. ALCCA: Aircraft Life-Cycle Cost Analysis

ALCCA is a program used for the prediction of all life-cycle costs associated with commercial aircraft. This includes manufacturing cost, acquisition price, and all operating and support costs (both direct and indirect). Developed by NASA-Langley, and further developed by ASDL, the program also calculates return on investment for the manufacturer and the airline. ALCCA also captures the effects of such economic variables as passenger load factor, fuel costs, and aircraft purchase financing (Marx 1).

As part of its ASDL development, ALCCA has been linked with FLOPS, providing the capability to perform a conceptual aircraft design and immediately determine its life-cycle costs. The linkage is also valuable through the automatic passing of aircraft design characteristics from FLOPS to ALCCA for the detailed calculation of manufacturing costs.

Although lacking a military aircraft analysis capability, ALCCA was utilized to analyze life-cycle costs for all aircraft designed in this study. The method of using ALCCA to analyze the GRA aircraft is described in detail later.

3.2.3. TCM: Tailored Cost Model

TCM was originally developed by Greg Bell at the McDonnell Douglas Corporation. The initial version was a series of Louts spreadsheets linked to perform a detailed economic analysis of military aircraft life-cycle costs. It utilizes cost-estimating relationships based on correlations against historical data for existing military systems (Osburg 2). TCM was further developed by Jan Osburg in ASDL, who imported TCM to Microsoft Excel, and then streamlined its execution (to reduce computational resource requirements) and improved its ease-of-use.

Clearly, TCM is the more accurate of the two cost analysis programs described here for analyzing the GRA aircraft developed in this study. A lack of experience using TCM, and limited time in which to gain such experience, prevented its implementation in this study.

3.3. Description of Baseline Aircraft

3.3.1. Global Reach Aircraft configuration

The GRA used in this concept is derived from a Lockheed-Martin Aeronautical Systems Company (LMASC) concept for a twin-engine, conventional wing-body-configuration aircraft. The design incorporates a large, high wing with a T-tail empennage arrangement, as depicted below in Figure 3.3.1-1. Figure 3.3.1-2 shows the comparison of C-5 Galaxy and GRA dimensions.
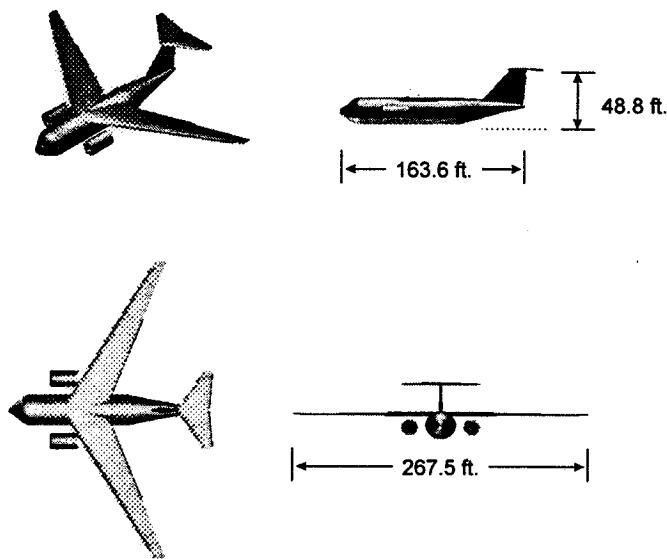
Figure 3.3.1-1. LMASC GRA arrangement.
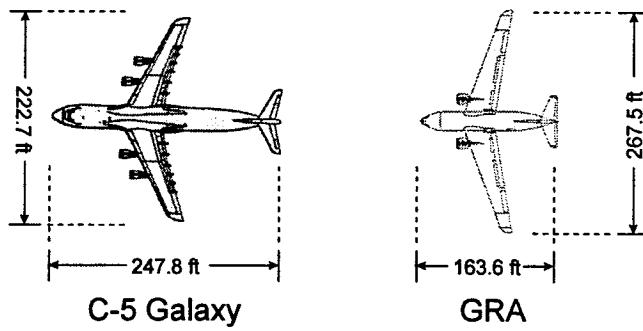
Figure 3.3.1-2. Comparison of C-5 and GRA.

The GRA is a new aircraft design capable of transporting 150,000 lb. of cargo over global distances up to 12,000 nautical miles at subsonic speeds in the vicinity of Mach number 0.8. It was first conceived to meet these requirements, set forth by the Air Force Scientific Advisory Board in the New World Vistas. Advanced technologies include natural laminar flow control (NLFC), composite wings and empennage, and twin IHPTET (Integrated High Performance Turbine Engine Technology) powerplants. The number of crew include two flight crew, two

backup flight crew, and two Air Force loadmasters. Table 3.3.1-1 below lists the important characteristics of the GRA.

Table 3.3.1-1. GRA Detailed Information.

Parameter	Value
Fuselage Length (ft)	163.6
Wing Span (ft)	267.5
Wing Area (sq ft)	6815.0
T/W	0.246
TOFL (ft)	10000
LDFL (ft)	3300
Approach Speed (kts)	155.0
TOGW (lbs)	834901
Fuel Required (lbs)	425000
No. Flight Crew	4 + 2
Range (nmi)	12500
Payload (lbs)	150000
Cruise Mach no.	0.80

The aerodynamic performance characteristics for the GRA baseline are shown in Figure 3.3.1-3 for climb and cruise Mach numbers, and take-off and landing in Figure 3.3.1-4. The GRA baseline cruises at Mach 0.78 at a lift-to-drag ratio of 24.7[18]. As can be seen below, the drag rise effects substantially reduce the maximum Lift-to-Drag (L/D) ratio as Mach number increases. This effect must be minimized through optimizing the geometry so as to achieve a higher cruise Mach number. Once again, the C_{Lmax} achieved at take-off and landing were 1.89 and 2.7, respectively.

Legend:
- M=.68
- M=.72
- M=.76
- M=.78
- M=.80
- M=.82
- M=.84

Drag Coefficient, C_D
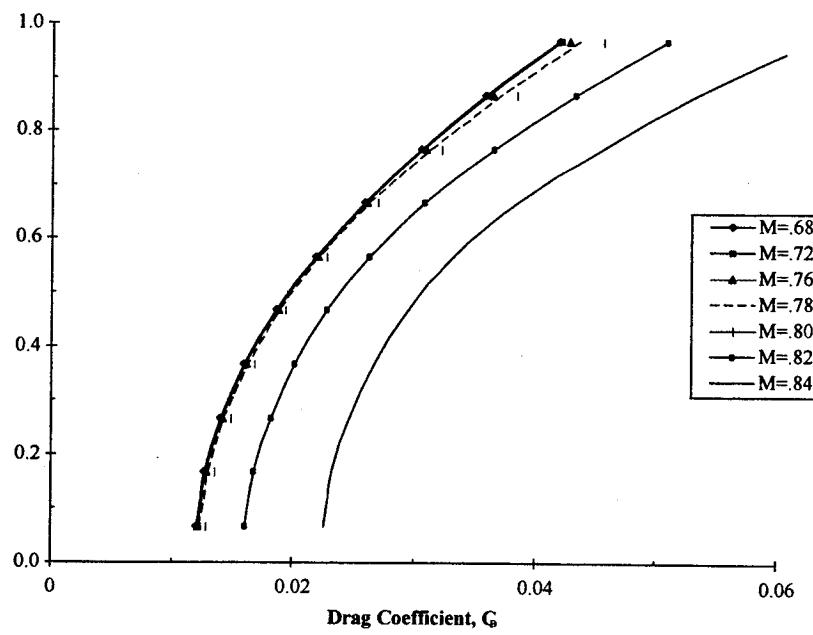
Figure 3.3.1-3. GRA Cruise Drag Polars
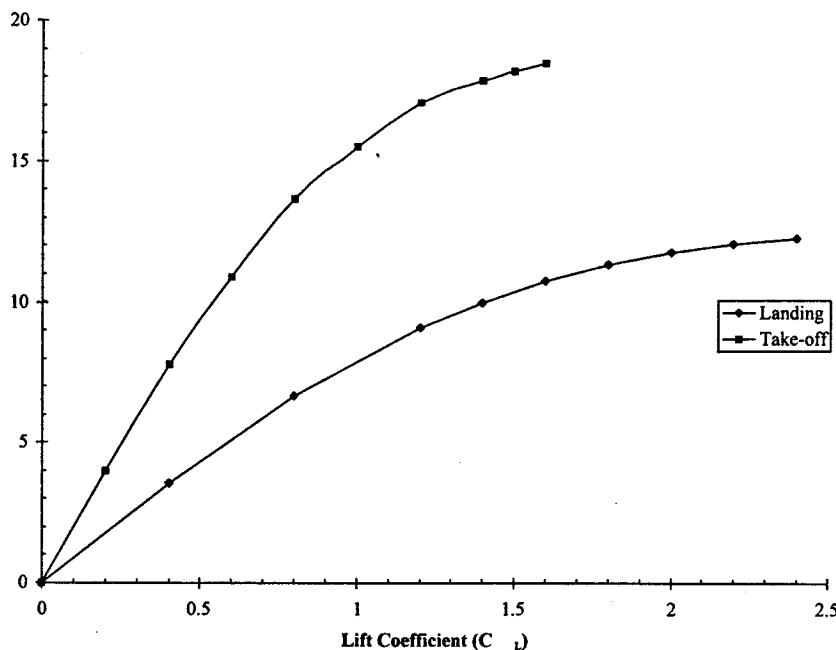
Figure 3.3.1-4. GRA Landing and Take-off Drag Polars

3.3.2. Very Large Transport configuration

The VLT, as envisioned by industry and government, is an advanced, dual passenger deck, four-engine advanced subsonic aircraft. The geometric layout of an 800-passenger VLT is provided below in Figure 3.3.2-1. Figure 3.3.2-2 shows the comparison of Boeing 747-400 and VLT dimensions. The baseline configurations of the VLT have been recreated at ASDL based on work performed by Dennis Bartlett at the NASA Langley Research Center.
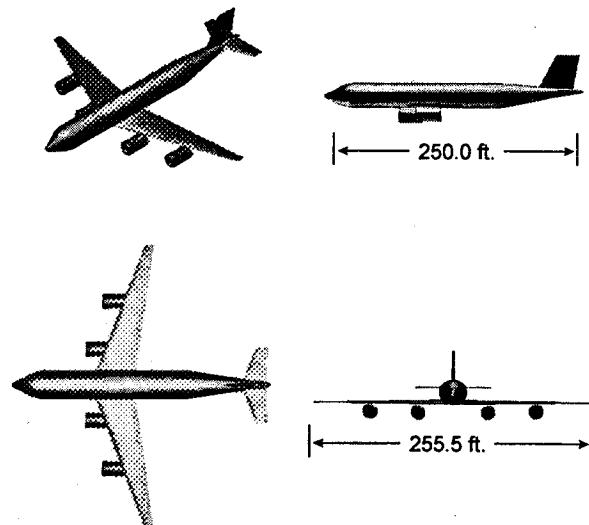
Figure 3.3.2-1. General VLT geometric layout.
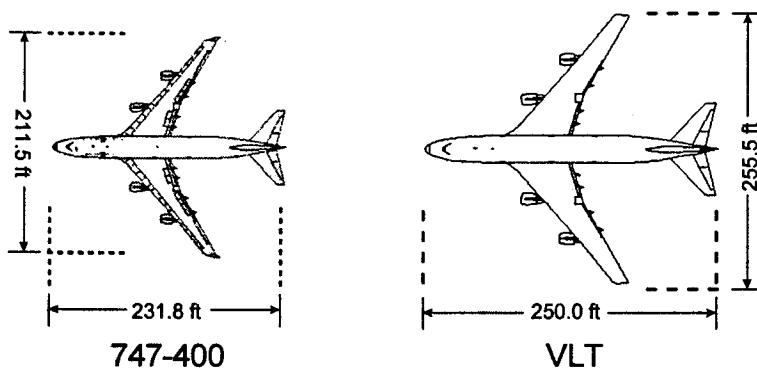
Figure 3.3.2-2. Comparison of 747-400 and VLT.

The configurations were sized by FLOPS with an engine technology level representative of 1996 entry into service for the subsonic mission depicted above. The design cruise Mach number was 0.85, consistent with current subsonic transports. The number of crew include two flight crew and two backup flight crew, plus 38 flight attendants and galley crew. These and other characteristics, encompassed by the above configurations, are listed in Table 3.3.2-1.

Table 3.3.2-1. VLT Detailed Information.

Parameter	Value
Fuselage Length (ft)	250.0
Wing Span (ft)	255.0
Wing Area (sq ft)	5934.0
T/W	0.257
TOFL (ft)	11000
LDFL (ft)	5500
Approach Speed (kts)	150.0
TOGW (lbs)	914039
Fuel Required (lbs)	334148
No. Flight Crew	4 + 38
Range (nmi)	7500
Pax. Cap.	800
Cruise Mach no.	0.85

Originally, NASA-Langley and ASDL developed three VLT variants, corresponding to three passenger capacities (600, 800, and 1000). Figure 3.3.2-3 below depicts the geometric differences between each of these aircraft and the Boeing 747-400. Sized to identical constraints, the 800-passenger VLT proved to be the most economically viable, so this configuration was selected for the DMLA study (Kirby 37).
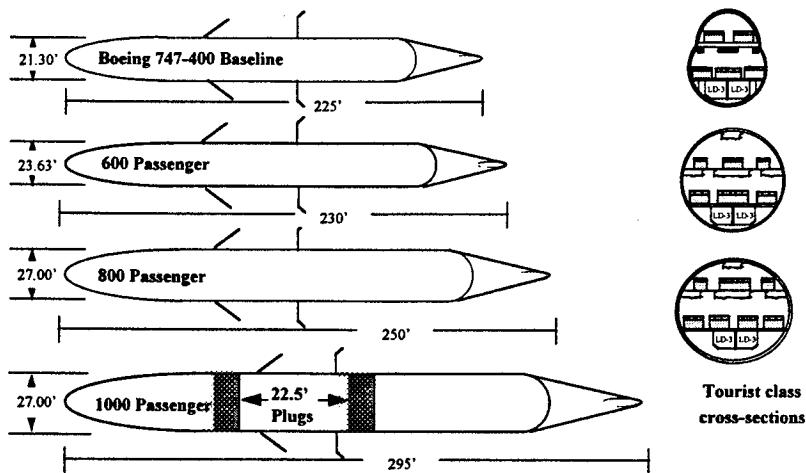
Figure 3.3.2-3. Comparison of geometries of VLT variants.

3.3.3. Description of Powerplant Models

The twin-engine aircraft in this study used the engine model developed by LMASC for the above GRA configuration. Starting with a General Electric Cf6-80E engine, nominally rated at 70,000 lb. of thrust, LMASC first scaled the engine to the required 105,000 lb. of thrust. They then assumed a 7% overall improvement in specific fuel consumption (sfc) from IHPTET technologies, resulting in a nominal cruise sfc of 0.553.

The four-engine aircraft in this study use an engine "deck" file created by the NASA Lewis Research Center. The modeled engines, nominally rated at a sea-level-static thrust of 77,500 lb., assume 1995 technology levels. For consistency with the twin-engine aircraft powerplant, the engine was modified to 1996 technology levels within FLOPS. Specifying a 1996 level of technology forces FLOPS to improve the engine component efficiencies over their 1995 levels.

3.3.4. Modification of Baselines

As stated previously, this study is only the first step in the entire DMLA research effort. As such, technical feasibility of the DMLA concept is illustrated by examining purely conventional aircraft variants. In other words, all aircraft developed during this study embody 1996 technology levels - this stage does not consider future advanced technologies. Thus, the above GRA baseline model was "stripped" of its enabling technologies, namely its natural laminar flow control and composite materials usage. NASA Langley provided a conventional VLT baseline, so no such modifications were necessary.

Furthermore, to add a dimension of comparison among aircraft, a four-engine GRA aircraft and twin-engine VLT aircraft were created. In FLOPS, this amounts to simply using the settings for number of engines, engine thrust, engine weight, engine wing locations, and the engine definition file location from the input from one aircraft to another. In other words, these values for the twin-engine GRA were used in creating the input for the twin-engine VLT; likewise, the values for the four-engine VLT were used in creating the input for the four-engine GRA.

3.4. Design of Specialized GRA and VLT Aircraft

As mentioned previously, this research problem seeks to demonstrate technical, as well as economic, feasibility of the DMLA concept. If a DMLA aircraft is not technically feasible, then its economics have no bearing. For this reason, this study addresses performance issues first, and cost issues second. As a result, GRA and VLT aircraft sizing occurred through the performance optimization of the baseline aircraft described above. The results carried forth into the initial development of a DMLA.

3.4.1. Aircraft Optimization

The aircraft design optimization was conducted using the FLOPS internal optimizer, which minimizes the following objective function (McCullers 17):

$$OBJ = obg \times GW + off \times FW + obg \times \left(M \times \frac{L}{D}\right) + ofr \times RNG + ofc \times COST + osfc \times SFC + ofnox \times NOX$$

The terms of this equation are defined in Table 3.4-1 below.

Table 3.4-1. Responses and Weighting Factors in FLOPS Objective Function.

term	value	weighting factor
GW	gross weight	obg
FW	fuel weight	off
M	cruise Mach no.	ofm
L/D	lift-to-drag ratio	ofm
RNG	design range	ofr
COST	life-cycle cost	ofc
SFC	engine specific fuel consumption	osfc
NOX	NOx emissions	ofnox

To minimize this function, FLOPS optimizes the following parameters within a user-specified range of values: gross weight, wing aspect ratio, engine thrust (or thrust-to-weight ratio), wing area (or wing loading), wing taper ratio, wing sweep angle, and wing thickness-to-chord ratio.

The last four scale factors in Table 3.4-1, **ofr, ofc, osfc,** and **ofnox,** were each set to zero. Each aircraft's mission profile specifies a fixed range, so this parameter was not varied. Aircraft costs were later analyzed for the resulting optimized aircraft; so this figure was not included in aircraft optimization. Engine SFC is fixed by the use of predefined engine models, eliminating this parameter from optimization. Furthermore, **off** was also set to zero, based on the assumption that fuel weight would be captured in the optimization of aircraft gross weight; thus, fuel weight was removed from the optimization process.

The remaining weight factors were assigned values based on the assumed desires of the military and commercial customers. In either case, it is desirable to maximize cruise Mach number and lift-to-drag ratio, while minimizing takeoff gross weight; thus, the values for these weight factors are assigned based on the relative importances of these quantities. For all GRA aircraft, **ofm = 0.33** and **off = -0.66** (the negative sign signifying a quantity to be maximized). As described previously, the military seeks to replace its existing aging transports because of their slow cruise speeds and poor aerodynamic performance, more so than due to their gross weight. Thus, cruise Mach number and lift-to-drag ratio were given a greater weighting than gross weight. For all VLT aircraft, **ofm = 0.66** and **off = -0.33**. Airport limitations constrain the gross weight of a commercial transport, somewhat shifting the design emphasis towards this effect. Furthermore, NASA-Langley originally sized the aircraft for a commercially acceptable cruise Mach number; thus, greater weighting was given to takeoff gross weight.

Tables 3.4-2 and 3.4-3 below list the design parameters, initial values, and ranges used to optimize the GRA and VLT aircraft, respectively. The initial values were taken from the results for the original, unoptimized baseline aircraft. The ranges were set to capture as wide a range of performance results as possible, while still reflecting physically sensible values.

Table 3.4-2. GRA Design Parameters and Ranges.

Parameter	Initial	Minimum	Maximum
Gross Weight (lbs)	900000	800000	1000000
Thrust/eng. (lbs)	110000	85000	135000
Aspect Ratio	9.5	8.0	11.0
Wing Area (sq ft)	6800.0	5500.0	8100.0
Taper Ratio	0.25	0.21	0.28
Wing LE Sweep (deg)	25.0	18.0	32.0
Thickness/Chord (avg)	0.11	0.08	0.13
Cruise Mach no.	*0.78	0.65	0.86
Cruise Altitude (ft)	45000.0	25000.0	50000.0

* 0.74 for 2-engine GRA

Table 3.4-3. VLT Design Parameters and Ranges.

Parameter	Initial	Minimum	Maximum
Gross Weight (lbs)	1250000	900000	1350000
Thrust/eng. (lbs)	160000	135000	185000
Aspect Ratio	8.5	7.0	10.0
Wing Area (sq ft)	8100.0	6500.0	9000.0
Taper Ratio	0.31	0.28	0.34
Wing LE Sweep (deg)	34.5	29.0	40.0
Thickness/Chord (avg)	0.08	0.06	0.10
Cruise Mach no.	0.80	0.65	0.86
Cruise Altitude (ft)	45000.0	25000.0	50000.0

Note that for the GRA configuration, FLOPS could size the twin-engine GRA configuration successfully only if a lower starting Mach number was given. This is a limitation in the FLOPS optimizer encountered in this study, to be discussed later. Furthermore, the GRA aircraft required lower starting Mach numbers than the VLT aircraft. The original sizing, which resulted in aircraft capable of Mach 0.80 cruise speeds, assumed natural laminar flow control; the additional aerodynamic drag caused by removal of this technology reduced the aircraft cruise speed.

The results of this sizing will be given in comparison with those for the DMLA aircraft later, in the "Results and Conclusions" chapter.

3.4.2. Identification of Critical Sizing Mission

The results of optimizing the specialized GRA and VLT aircraft became the means by which the more critical of the two missions could be identified. This mission would yield an aircraft with a higher cruise Mach number, greater range, greater payload, and a larger amount of fuel required. The critical mission becomes critical to the sizing of a DMLA, as it is such a mission for which any DMLA must be optimized.

This line of reasoning emerges from a common-sense observation of the problem. Between the GRA and VLT aircraft, that with greater range, speed, payload capacity, and fuel capacity should be able to perform a mission requiring a slower, shorter-ranged aircraft with less payload and fuel capacity.

By this rationale, the VLT mission was found to be the more critical, in both the twin-engine aircraft and four-engine aircraft cases. This is shown below in Tables 3.4.2-1 and 3.4.2-2.

Table 3.4.2-1. Twin-Engine Aircraft Mission Criticality.

Parameter	GRA	VLT
Range (nmi)	7500	7500
Payload (lbs)	150000	167200
Speed (M)	0.758	0.770
Fuel Req. (lbs)	326724	474553

Table 3.4.2-2. Four-Engine Aircraft Mission Criticality.

Parameter	GRA	VLT
Range (nmi)	7500	7500
Payload (lbs)	150000	167200
Speed (M)	0.787	0.778
Fuel Req. (lbs)	345945	456250

3.5. DMLA Creation

Since the VLT mission is more critical from an aircraft sizing standpoint, then a DMLA must be optimized for this role first. The process began by modifying the resulting VLT configuration in FLOPS with some of the necessary attributes of a GRA. Since military transport aircraft contain a high-wing configuration (to allow cargo handling equipment to drive under the wings (Wilkinson 29)), and therefore a T-tail empennage arrangement, a DMLA must also possess a high wing and T-tail. (Note that FLOPS models only the latter of these design characteristics.) Also, Air Force regulations limit the wing structural load factor to 2.25, much less than the 3.75 load factor employed by commercial transports; this limits the DMLA's wing load to 2.25 as well. Otherwise, the VLT's attributes are retained in the DMLA configuration, with all other FLOPS inputs for the VLT held identical. (In particular, the same optimized variables and ranges input for the VLT were used in modeling the DMLA to achieve similar performance.) The resulting VLT variant of the DMLA was then optimized following the same approach as for the specialized VLT.

Thus, a VLT variant of the DMLA was modeled, but a GRA variant had yet to be created. This aircraft was modeled starting with the results obtained for the VLT variant's creation. At this point, however, the configuration was "frozen" - that is, all optimization features were removed, in order to retain the DMLA geometry. The design variables previously optimized were fixed at those resulting from the VLT variant's optimization, and the configuration was run in FLOPS in an analysis-only mode. Since this mode still results in the recalculation of some parameters, however, as many of these parameters needed to be fixed as well. In addition to the design variables, all component weights were fixed to their VLT variant values. The exception was fuselage weight, which differed due to the following additional modifications. First, the previous GRA study yielded an aircraft requiring a fuselage 164 ft. in length, far less then the 250 ft. length of the VLT fuselage; the study revealed that the 150,000 lb. payload could easily be carried in a fuselage of this size. (This fleshes out the assumption that on a DMLA assembly line, fuselage plugs would be implemented to lengthen a DMLA to the VLT-variant size.) Further modifications of the FLOPS input include:

- Setting of passenger capacity to zero.

- Setting of flight crew number to six.

- Specifying a main-deck cargo floor, instead of a passenger cabin floor.

- Replacement of VLT mission definition with GRA mission.

- Fixation of wing fuel capacity to VLT variant value. Since both variants must have identical components, the wings - and therefore wing fuel tanks - must likewise match.

- Removal of fuselage fuel tanks, as all military transport fuselage volume is devoted to the aircraft cargo bay.

This last point created some difficulty in creating the DMLA aircraft. The VLT variant needed to be sized with sufficient wing fuel capacity for the GRA variant to complete its mission. When

the GRA was analyzed and insufficient fuel capacity resulted, the VLT was then resized with higher initial values of wing area and thickness to accommodate the additional fuel volume. The GRA was then reanalyzed, and the wing geometry refined until the GRA excess fuel capacity was minimized.

With these modifications made to the FLOPS input, the GRA variant of the DMLA could then be analyzed. The results of the DMLA design will be given in comparison with those for the specialized aircraft later, in the "Results and Conclusions" chapter.

4. Economic Feasibility Approach

With all aircraft sized, demonstration of the DMLA concept's economic feasibility remained. This entailed the analysis of all resulting aircraft in the ALCCA program. In all but a few instances, the economic variables and their "most likely" values used by the Boeing 747-400 project team of the AE4353 Design for Life-Cycle Cost class, last offered in the Fall of 1996, were utilized in this analysis. The same metrics for economic viability were used as well. (The work performed by the 747-400 team is referenced at the end of this report.)

4.1. Cost Analysis for all VLT Aircraft

Since ALCCA predicts the life-cycle costs for commercial aircraft, its use to analyze the economics of VLT aircraft is unquestionably justified. In line with the work performed for the AE4353 class project, the values and assumptions given in Table 4.1-1 formed the basis of the VLT aircraft economic analysis:

Table 4.1-1. VLT Economic Values and Assumptions.

Value	Setting	Value	Setting
Dollar Year	1992	Engine Learning Curve	100%
Production Year	1997	Turnaround Time (hrs)	1.5
Airline ROI	9%	Inflation Rate	6%
Manufacturer's ROI	12%	Econ. Life (years)	20
Econ. Range (nmi)	5160	Residual Value (% of acq. cost)	0.10
Fuel Cost ($/gallon)	0.65	Utilization (hrs/yr)	5000
Hull Insurance (% of acq. cost)	0.35	Production Quantity	*549
Engineering Labor Rate ($/hr)	65	Load Factor	65%
Tooling Labor Rate ($/hr)	55	Learning Curve	80%
Maintenance Labor Rate ($/hr)	19.5	Downpayment	20%
Income Tax Rate	34%	Financing	8%

* 900 for DMLA

Some of these figures are a bit optimistic: airlines may accept a 10% return on investment; flights abroad where fuel costs are higher could increase the average fuel cost to over one dollar per gallon; and airlines often do not provide a down payment when purchasing new aircraft. However, as for the GRA, analysis of each VLT aircraft included these same assumptions, validating the comparison of these results.

Tracking the same metrics as used in AE4353 allowed for the assessment of affordability. These include aircraft acquisition cost (ACQ), required average yield per revenue passenger-mile ($/RPM), direct operating cost per trip (DOC), indirect operating cost per trip (IOC), and total operating cost per trip (TOC). The "Results and Discussion" chapter includes the VLT results for these metrics, and their comparison as a gauge of economic feasibility.

4.2. Cost Analysis of GRA Aircraft

Modeling military aircraft economics in ALCCA far exceeds the program's validity. The mathematical relations in ALCCA were derived from commercial aircraft statistics. However, by modeling all GRA aircraft as commercial cargo planes, and applying the same assumptions in ALCCA to all GRA aircraft, then their economics can be compared against each other with confidence.

Starting with the same ALCCA values used for the VLT aircraft, some obvious adjustments were made to attempt improvement of the ALCCA models for the GRA aircraft. First, the number of passengers was set to zero, to tap into ALCCA's ability to model commercial cargo aircraft economics - the closest model to that for military transports. Second, the corporate tax rate was also set to zero, as the military does not owe income tax to the government for its operations. Third, the airline return on investment routine in ALCCA was disabled, as the military does not monetarily profit from its operations. Fourth, the full 7500 nautical mile design range was input as the economic range, based on the assumption that the military will operate its aircraft at or near their full design capability. Finally, the production quantity was set to 351. This is the combined total of all C-5 and C-141 aircraft in active service (C-5 2, C-141 2); it is assumed that the Air Force will replace all of its front-line C-5s and C-141s. All other assumptions and values from that for the VLT aircraft were kept, resulting in the list of economic variable settings and assumptions given in Table 4.2-1.

Table 4.2-1. GRA Economic Values and Assumptions.

Value	Setting	Value	Setting
Dollar Year	1992	Engine Learning Curve	100%
Production Year	1997	Turnaround Time (hrs)	1.5
Airline ROI	N/A	Inflation Rate	6%
Manufacturer's ROI	12%	Econ. Life (years)	20
Econ. Range (nmi)	7500	Residual Value (% of acq. cost)	0.10
Fuel Cost ($/gallon)	0.65	Utilization (hrs/yr)	5000
Hull Insurance (% of acq. cost)	0.35	Production Quantity	*351
Engineering Labor Rate ($/hr)	65	Load Factor	65%
Tooling Labor Rate ($/hr)	55	Learning Curve	80%
Maintenance Labor Rate ($/hr)	19.5	Downpayment	20%
Income Tax Rate	0%	Financing	8%

* 900 for DMLA

The same responses were tracked, with the exception of $/RPM, for the simple reason that no (revenue) passengers would be embarked on any GRA. For this same reason, summation of the components of IOC was performed manually; IOC includes components related to passenger cabin servicing, which ALCCA computes by distributing over all passenger seats, of which there are none on the GRA aircraft. Thus, the passenger-service related components of IOC were infinite, resulting in an infinite overall IOC. Thus, the remaining components - flight servicing, aircraft maintenance, cargo handling, and ground terminal fees were manually summed.

The "Results and Discussion" chapter includes the GRA results for the remaining metrics, and their comparison as a gauge of economic feasibility.

4.3. DMLA Considerations

The driving force behind the DMLA's economic viability is the impact on affordability brought about through the development of single aircraft, instead of two specialized systems, along with the affect of a longer production run of a single aircraft versus two smaller production quantities for two distinct aircraft.

The primary method for affecting this in ALCCA is to combine the proposed GRA and VLT production quantities, for a total of 900 aircraft (549 VLT variants plus 351 GRA variants). This is largely accurate, but overlooks a subtle point in the reality of DMLA production. Given that the VLT and GRA differ somewhat, the following finer points of the added costs of DMLA manufacturing were overlooked in this study:

- cost of fuselage plug insertion to create a VLT aircraft

- cost of altering the aircraft interior on the assembly line to adapt it for the given variant's mission

- cost of configuring aircraft flight decks for avionics packages appropriate to the given variant

For this reason, the results for the DMLA acquisition costs are slightly optimistic. Again, these are finer points to be addressed in future studies; the impact of their neglect is slight at most.

5. Aerodynamic Investigation Approach

To accomplish the above objectives, a method for execution was identified. The elements composing the Robust Design Simulation (RDS)[12] method developed by ASDL were adopted. One key element within RDS is the Response Surface Methodology (RSM). RSM is a method which allows the development of Response Surface Equations, say system level metrics or constraints, as functions of important design parameters. An overview of the theory behind this method is described below.

5.1. Response Surface Methodology

RSM[12] is one of the key elements comprising ASDL's RDS method. It is based on a statistical approach to building and rapidly assessing enormous empirical models. By a careful design and analysis of experiments or simulations, the RSM seeks to relate and identify the relative contributions of the various input variables to the desired system response, e.g., TOGW, VAPP, block time, etc. In most cases, the behavior of a measured or computed response is governed by certain laws which can be approximated by a deterministic relationship between the response and a set of design variables. The exact relationship between this response and the design variables is either too complex or unknown and an empirical approach is necessary. The strategy employed in such an approach is the basis of the RSM. In this study, a second degree model in k-variables is assumed to exist for each metric and constraint. This second-order polynomial, called a Response Surface Equation (RSE), for a response, R, can be represented as:

$$R = b_o + \sum_{i=1}^{k} b_i x_i + \sum_{i=1}^{k} b_{ii} x_i^2 + \sum_{i=1}^{k-1} \sum_{j>1}^{k} b_{ij} x_i x_j \quad (1)$$

where: b_i are regression coefficients for the linear terms; b_{ii} are coefficients for the pure quadratic terms; b_{ij} are coefficients for the cross-product terms (i.e. second-order interactions); x_i, x_j are the design variables; and $x_i x_j$ denotes the interactions between two design variables. Once equation (1) is derived, it can be used in lieu of more sophisticated, time consuming codes to predict and optimize the response of a sub-system or the entire system. The "optimal" settings for the design variables are identified by finding the maximum or minimum of this equation. The response equation can then be validated from the original code by performing a confirmation test with the optimal settings of the design variables. Since the RSE is essentially a regression curve, a series of experimental or computer simulation runs (or cases) need to be performed to obtain a set of output data for the varying inputs.

5.2. Design of Experiments

Typically, one wants to obtain the RSE as a function of several variables, including second-order interactions. To evaluate all possible combinations of variables at two or three levels, an excessive amount of cases would need to be tested. In fact, if seven variables, as seen in Table 5.2-1, are to be tested at three levels (two extremes and a most likely value) for all combinations, a total of 2,187 cases would need to be evaluated. This type of evaluation is called a full factorial analysis. This type of analysis would be absurd if a detailed analysis code such as a structural Finite Element Method or a Computational Fluid Dynamics code was needed. Therefore, to reduce the amount of runs needed to analyze a response, a statistical method called the Design of Experiments (DoE) is employed.

Table 5.2-1. Number of Cases Needed to Perform a Response Surface Analysis

Type of Evaluation	Equation	For n=7 Variables
Full Factorial	3^n	2,187
Central Composite	2^n+2n+1	143
Box-Behnken	-	62
D-Optimal	$(n+1)(n+2)/2$	36

DoEs are statistical techniques which allow for a portion of the full factorial to be analyzed in a structured, predefined manner. A DoE will yield a table of input variable combinations that will capture all first and second-order effects due to changes in the design variables[14]. Table 5.2-2 displays an example of a DoE table for three variables at two levels, "-1" and "+1", representing the extreme (minimum and maximum) values of the range of interest. Each row represents a run for the given variable (x_1, x_2, and x_3) level settings and the last column denotes an output, or response, of some experimental or simulation code.

The actual combination of cases that need to be tested can be determined from a textbook, or as in the case here, through the use of a statistical analysis program called JMP[15]. JMP not only constructs the tables, but also identifies confounding structures and carries out all necessary analyses once the responses are provided from the simulation runs. The same DoE approach can be utilized for variables at three levels yet requires more runs to obtain the same information. A note should be made that through the use of statistical methods, such as the DoE, the amount of necessary variables for analysis can be reduced for the same number of runs.

In order to reduce the number of variables, another DoE is needed to identify the contribution of each originally considered variable to the response of the system. This test is commonly referred to as a screening test. The screening test is a two-level fractional factorial DoE testing the fit of a linear model. This is achieved by considering only the main effects of each variable, i.e., no interactions. This method allows for a rapid investigation of many variables so as to gain an initial understanding of the problem.

Table 5.2-2. DoE Example for 2^3 Factorial Design

Run	Factors			Response
	x1	x2	x3	
1	-1	-1	-1	y1
2	+1	-1	-1	y2
3	-1	+1	-1	y3
4	+1	+1	-1	y4
5	-1	-1	+1	y5
6	+1	-1	+1	y6
7	-1	+1	+1	y7
8	+1	+1	+1	y8

The JMP program can once again be utilized for the statistical analysis. This analysis yields a Pareto plot of the most significant contributors. The Pareto plot is based on the Pareto principle which states that 20% of the variables in a given system control 80% of the variability in the dependent variable[16]. By pre-defining a desired response fidelity, the actual values needed can be easily selected from the plot. Typically, seven to eight variables are adequate enough to capture 80% to 90% of the response. The variables not contributing a significant amount to the response can be fixed at their most likely values for the remainder of the study.

After identifying the variables which will form the RSE, a design must be selected from the list of potential candidate methods presented in Table . For the purpose of this study, the face-centered Central Composite design was used to develop the RSEs. The face-centered Central Composite design is a fractional factorial with axial points located on the face of the cube[13]. In addition, since all runs will be conducted with an analytical tool, no experimental error occurs. Therefore, a statistical environment without any error can be assumed. Consequently, all deviations from the predicted values are true measures of a model fit, i.e., a lack of fit of the assumed quadratic equation representation of the response. A lack of fit in the model expresses how well the model represents the true response. A small error in fit is indicative of higher order interactions that are not accounted for in the model. If the error between predicted and actual responses is too large, it may be necessary to create a new model which accounts for those interactions or to assess the validity of the analysis tool.

5.3. Prediction Profiles

Once the RSEs have been generated, the program JMP also allows the development of prediction profiles[13]. Simply stated, the prediction profiles are a series of graphs for each variable which represent the relative effect of each variable on the response. The prediction profiles are generated by JMP via the underlying response surface equations. The importance of the prediction profiles is that they allow a visual interpretation of the effects of the variables on the response. By examination of the prediction profiles, one can determine the variables of

strongest importance, the variables of least importance, the direction (increase or decrease) of change of response for a direction of change of variable, and the rate of change.

Interpretation of the prediction profiles is relatively straight forward. The slope of the line is the key. A positive slope means that for an increase in the variable a corresponding increase in the response value will occur. The opposite happens with a negative slope, i.e., for an increase in a variable value, a decrease in response value occurs. The value of the variable's slope is related to the magnitude of the increase in the response. In addition, the curvature of the variable's effect is important. A linear slope implies that an equal increase in variable will result in an equal (in relation to the slope value) change in response value; however, a non-linear slope implies the potential for diminishing/increasing returns. This is very important in the process of attempting to change a response value via a change in variable. Visually, one can analyze whether or not and/or to what level a change in variable level is cost effective. Finally, a horizontal or near horizontal slope is interpreted as having a minimal effect on the response value.

In a computer environment via JMP, the prediction profiles may be altered in an interactive manner. This tool allows the designer to alter the variable values in real time and observe the change in response. A quick analysis will tell the designer if changing the appropriate variable values can actually change the response to the desired value and if possible, which different variables need to be changed to what range of values. The real advantage is in the fact that the process occurs instantaneously without the need for complicated re-analysis due to the underlying response surface equations.

There is a function in the prediction profiles option that translates a multi-objective problem to one objective problem in the form of a "Desirability" function. This idea was pioneered by Derringer and Suich in 1980[17]. Consider the prediction profile in Figure , there are 10 different objective functions defined on the left. On the right of the figure is the desirability of each objective. The slopes of each individual objective shows the direction of highest desirability. For example, a negative slope implies minimization, while a positive slope implies maximization. As is evident, minimization of the individual objectives is the goal. Yet, the VAPP, LdgFL, TOFL, and the Acquisition price are constraints. For the constraints shown, the mean value of the desirability is set to zero at the constraint value and then minimized as the desirability increases to one. For the VAPP constraint, no values in the DoE table ever violate the limit of 155 knots; hence VAPP was subject to minimization. JMP will determine the most desirable analysis case in the DoE table which will maximize the desirability function based on the desirability of each objective defined on the right. Higher is always better for the desirability function value shown on the bottom left.

One of the main aspirations of this study was to maximize the cruise Mach number, VCMN. As shown in Figure 5.3-1, the maximum desirability resulted in the mid-value of the Mach number. The user has the option of changing this result, as shown in Figure 5.3-2. Yet, as is evident, the value of the desirability function decreases from 0.696955 to 0.660988. The user would then have to determine the trade-offs for a high desirability (that is, optimal solution) as compared to the study purpose of a high Mach number.
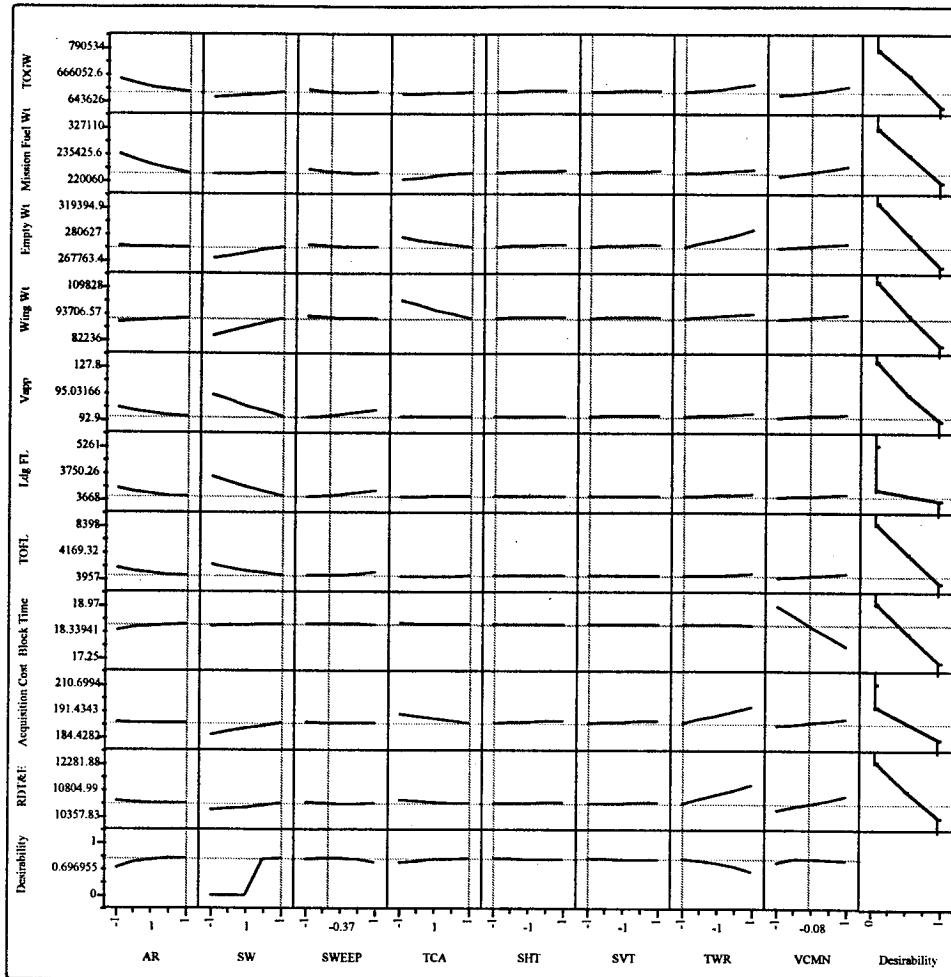
Figure 5.3-1. Desirability Option on JMP
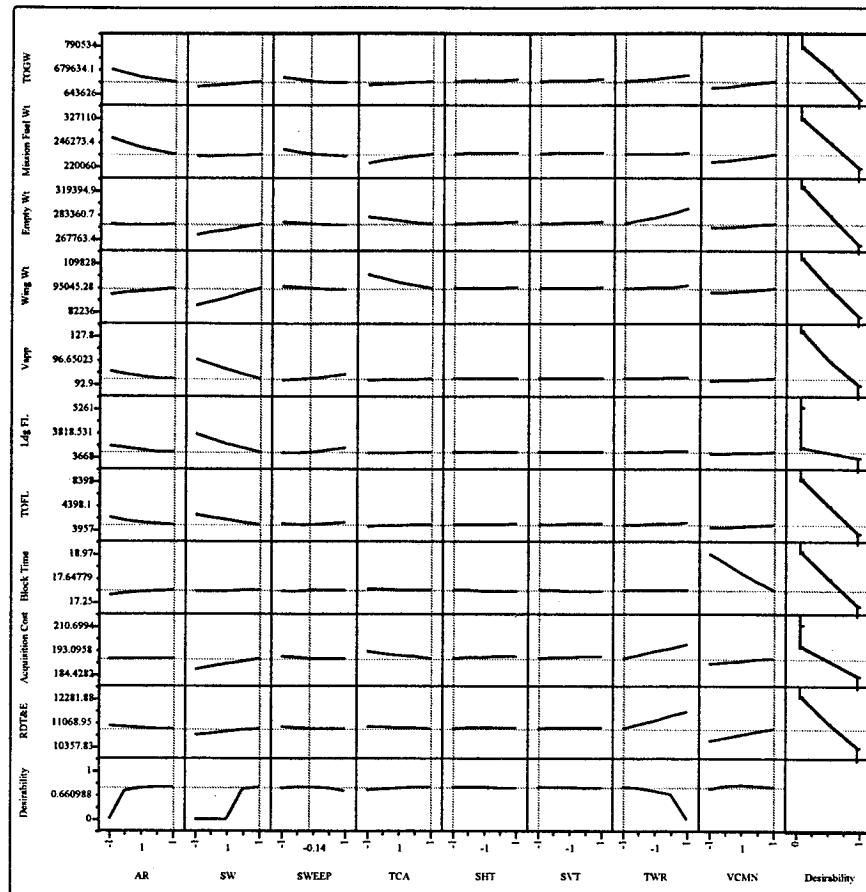
Figure 5.3-2. Adjusted Desirability Option in JMP

5.4. Analysis Tools and Methods

The execution of this study was dependent upon identifying aerodynamic analysis tools which could accurately capture the physics associated with a military or commercial transport aircraft cruising at high subsonic speeds. Furthermore, to incorporate the high number of executions required by the Response Surface methodology, automated analysis methods were a necessity. The following describes the analysis tools identified; four possibilities for execution; and the rational for the method utilized in this study; and the automation procedure for the analysis.

5.4.1. Analysis Tools

Various aerodynamic tools exist in industry which are capable of predicting the aerodynamics of subsonic transport aircraft. For an initial conceptual level trade study, an aerodynamic analysis based on potential flow theory is usually preferred. The tools have low fidelity but are time efficient. Typically, these codes solve the flow characteristics based on vortex-lattice paneling methods or grid structures, where the latter has higher fidelity but is more costly to execute. ASDL has many of these public domain codes which could be used for this study. These tools include AERO2S[21,22,23], BDAP[24,25,26], FPS3D[27], and VORLAX[28]. Each of these tools has a range of validity and application and requires different levels of user interaction and input development.

First, AERO2S is a subsonic 2-D vortex-lattice paneling method which is capable of analyzing lift induced drag of a wing and a second 2-D surface. AERO2S simulates 3-D configurations with an effective camber of the wing and second surface. A fuselage may be simulated with an effective camber. AERO2S can also handle leading and trailing edge flaps and is an industry standard for low speed and subsonic analysis. Paneling of the surfaces is achieved by specifying a number of spanwise slices along with panel element aspect ratios.

BDAP is a linearized method typically used for supersonic design and analysis. Yet, BDAP contains a module which will determine the subsonic profile drag characteristics of arbitrary shapes using turbulent flat plate theory.

FPS3D is a 3-D full potential flow solver. It requires an unstructured grid rather than panels. FPS3D is capable of accurately and efficiently predicting the aerodynamics of complete aircraft configurations at subsonic, transonic, and supersonic speeds. For the grid generation, FPS3D requires unstructured grids generated from FELISA[29].

VORLAX is a vortex lattice code that is capable of both supersonic and subsonic analysis. VORLAX can analyze both asymmetric and symmetric designs, making it applicable to unconventional geometries. Fusiform bodies may be modeled, as well as thickness as biplanar patches. VORVIEW[30] is a graphical preprocessor to VORLAX. It enables the user to use a hermite file as the geometry definition input. VORVIEW then automatically slices the planform view of the model and creates the VORLAX input file. For vertical paneling, the slices must be added manually.

5.4.2. Possible Analysis Methods

Based on the definition of the analysis codes above, three methods for execution were explored. Each method begins with the geometric modeler, RAM, then proceeds to analyze the configuration aerodynamics, and then feeds FLOPS the drag polars needed to size the aircraft. The methods are:

1. RAM→BDAP→AERO2S→FLOPS

2. RAM→VORVIEW→VORLAX→FLOPS

3. RAM→FELISA→FPS3D→FLOPS

Each one of these methods were investigated to determine if the actual analysis codes, i.e. AERO2S, VORLAX, and FPS3D, would efficiently capture the compressibility effects of high subsonic flight. This criteria was a necessity for this study since a primary objective was to increase the cruise Mach number for military and commercial compatibility. The baseline GRA was modeled in RAM and then analyzed by AERO2S for various Mach numbers. Figure 5.4-1 shows the drag polars generated by AERO2S for the GRA baseline. As compared to the actual drag polars in Figure 3.3.1-3, AERO2S is not capturing the drag rise effect of increasing Mach number. This inability to model the physics automatically eliminated AERO2S as the subsonic cruise analysis codes. VORLAX was also eliminated due to the fact that the analysis is based on theories similar to AERO2S. It does not capture compressibility effects.
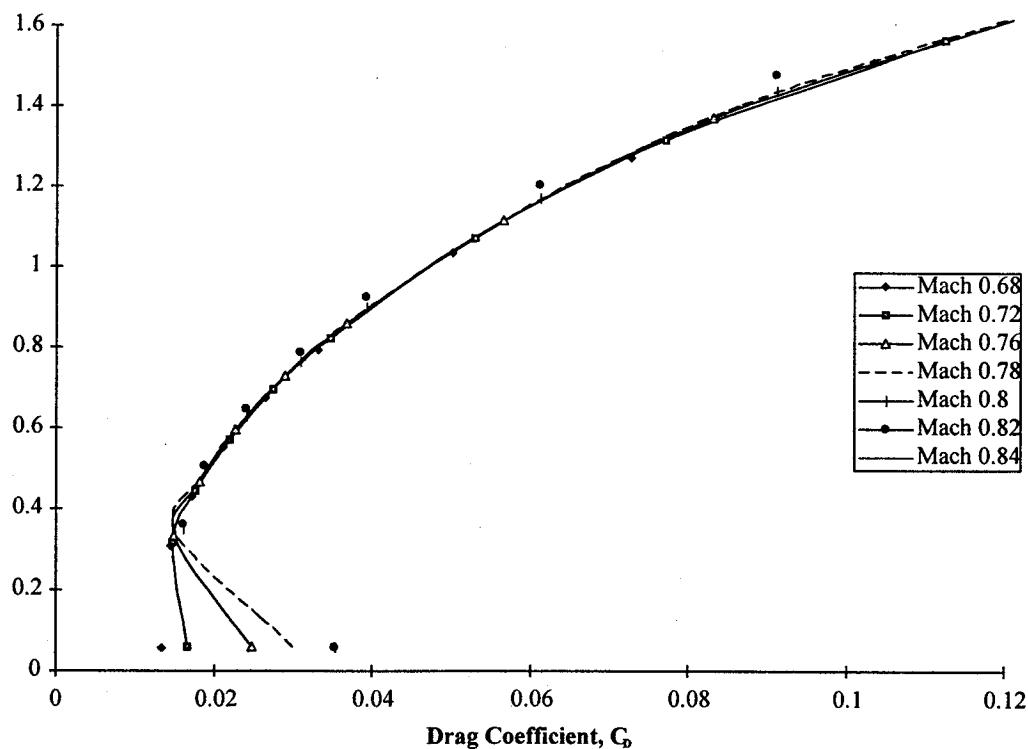
Figure 5.4-1. AERO2S Analysis of GRA Baseline

The only remaining option was method 3, RAM→FELISA→FPS3D→FLOPS. As stated previously, FPS3D could capture the compressibility effects desired. Even though the analysis would be more costly in set-up and execution time, the results would be valid. Yet, at the time of this study, the grid generation code, FELISA, required by FPS3D, had not been acquired. Consequently, FPS3D was also eliminated.

One final option was left to be explored and was considered as a last resort. Within FLOPS, there exists a module capable of analyzing the aerodynamics of a configuration. The analysis module is based on empirical relationships developed from current subsonic transport aircraft. It computes profile drag, induced drag, and accounts for compressibility based on a regression analysis of current transport performance data. As seen in Figure 5.4-2, FLOPS adequately captured the compressibility effects for the baseline configuration. At Mach 0.84, the drag rise effect is evident whereas it is non-existent in Figure 5.4-1. As compared to the actual drag polars in Figure 3.3.1-3, FLOPS over predicted the total drag. For example, the actual maximum Lift-to-Drag ratio at a cruise speed of Mach 0.78 is 24.7 and FLOPS predicted a value of 22.9 which was 7.2% less than the actual.
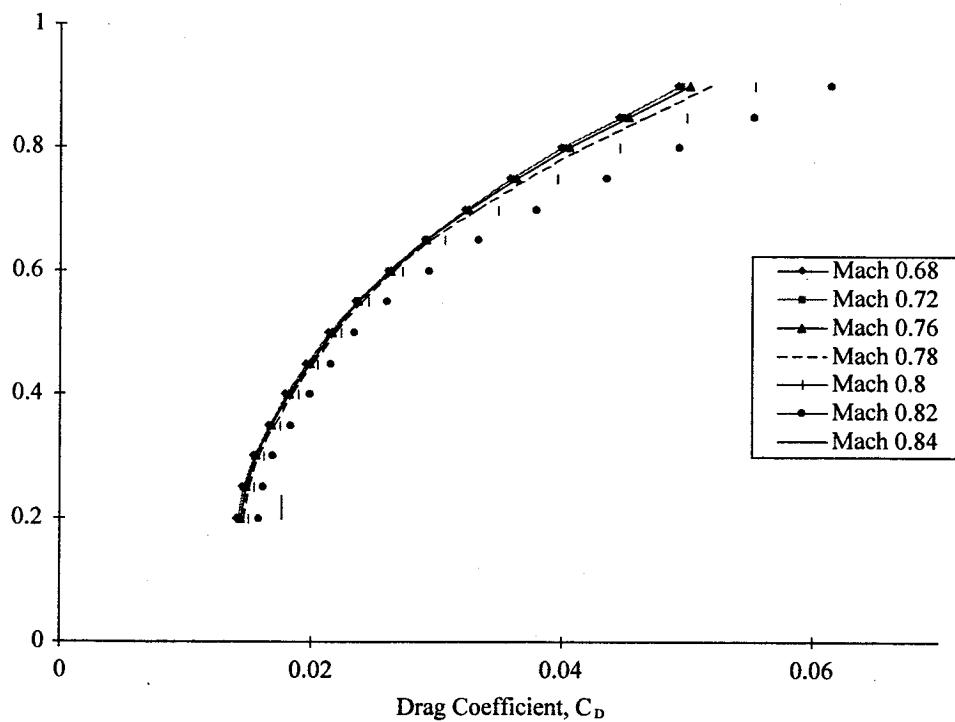
Figure 5.4-2. FLOPS Drag Polar Analysis of GRA Baseline

As a consequence of the investigation of the possible methods for execution, the internal aerodynamic analysis within FLOPS was the only alternative. This result simplified the overall analysis. This simplification had positive and negative aspects. Considering that the objectives of this study are system level metrics resulting from a FLOPS analysis, the amount of automation

required is reduced. Yet, the study gravitates from accurate representation to one of approximation.

Upon further investigation of the capabilities of the FLOPS aerodynamics module, it was discovered that the low speed characteristics (TOFL, LdgFL, and VAPP) were dependent upon externally provided drag polars. The actual aerodynamic analysis performed on a configuration did not translate to the take-off and landing module which calculates TOFL, LdgFL, and VAPP. This information would have to be provided from an external source. AERO2S and BDAP were utilized for this purpose. Eventhough AERO2S can not predict high subsonic Mach numbers, it has been proven to be quite valid for low speed analysis. Since AERO2S only predicts induced drag, the profile drag ability of BDAP was used. The combination of these two codes would allow for accurate low speed aerodynamic information for the determination of take-off and landing performance metrics.

5.4.3. Technology Infusion Modeling

After establishing the method of analysis, it was necessary to identify if various possibilities for technology infusion could be modeled or simulated. Based on the Mobility Volume of the New World Vistas[3], three areas of new technologies were identified. Those areas include: improved propulsive efficiency (i.e., post-IHPTET), better aerodynamic performance via advanced wing design and innovative configurations, and light-weight, low cost advanced materials. Each of the three technologies could be simulated within FLOPS/ALCCA[20,31]. The post-IHPTET engines could be simulated by using an external engine deck which reflected the thrust and fuel flows of the advanced technology. The aerodynamic performance could be enhanced by including flow control techniques, such as Hybrid Laminar Flow Control (HLFC). The HLFC technology can be simulated in FLOPS by assuming a percent of laminar flow over various surfaces, such as the wing, empennage, nacelles, and fuselage. And finally, the light weight advanced materials can also be fabricated through constant factors which reduce the calculated component weights.

As stated previously, the GRA baseline included an IHPTET engine and laminar flow over the wing and empennage. In addition, the GRA baseline described in Ref 18 correlated the component weights through the use of weight factors. Therefore, to simulate the various technologies state above, the original GRA baseline was stripped of these technologies to yield a new "clean" baseline configuration. Each technology that was added yielded a deviation from that baseline. Hence, the effects of adding new technologies could be quantified from an aircraft performance and system level metrics benefit point of view. As a summary, Table 5.4-1 defines the three technologies added to the "clean" version of the GRA and eight configurations resulted. Furthermore, the aerodynamic analysis was performed on each of the eight configurations, and optimal solutions were obtained for each. These different configurations provided the feasible space for the DMLA study.

Table 5.4-1. GRA Configurations Definition based on Technology Infusion

Configuration	Technologies Applied
1	Baseline with IHPTET engine
2	IHPTET engine and HLFC
3	IHPTET engine and composites
4	IHPTET engine, HLFC, and composites
5	Baseline with post-IHPTET engine
6	post-IHPTET engine and HLF
7	post-IHPTET engine and composites
8	post-IHPTET engine, HLFC, and composites

The technologies stated in Table 5.4-1 were simulated in FLOPS/ALCCA through various parameters. Table 5.4-2 lists the various parameters used and the values for the baseline and the technology. The introduction of the post-IHPTET engines was straight forward. In lieu of looking for the IHPTET engine deck, the post-IHPTET engine deck was inserted. The component weighting factors for the original GRA composed of increasing the weights calculated by FLOPS so as to correlate data. Therefore, composite use was simulated by allowing FLOPS to calculate the weights with no factors included. When the four factors shown (FRFU, FRHT, FRVT, and FRNU) were set at values of 1.0, the TOGW could be reduced by as much as 150,000 lbs. This was felt to be a reasonable reduction in weight through use of composites. The HLFC simulation required more variables. The HLFC technology values shown are indicative of the original GRA baseline before all technologies were removed, except for the C_{Lmax} at take-off and landing. As noted in the discussion of the internal aerodynamic ability of FLOPS, low-speed aerodynamic information is required from an off-line analysis. The tools utilized for this purpose, AERO2S and BDAP, cannot simulate the effect of HLFC. Therefore, a simulation of HLFC came through the effect of increasing the maximum lift obtainable through controlling the boundary layer. Since no data for HLFC impact on maximum lift was available, an analogy was made to the effects of circulation control on wing surfaces at take-off and landing. Various studies have shown that C_{Lmax} at take-off and landing can be increased by as much as 30% for circulation control airfoils. An assumption was made that the NLF characteristics of the HLFC wing could improve the C_{Lmax} at take-off and landing. Hence ,the C_{Lmax} at take-off and landing were increased by a conservative amount 11% and 14.8%, respectively.

The HLFC modeling in FLOPS was not completely accurate. Any time that blowing or suction is applied to a surface, the energy required to perform such a task is extracted from the engine. For this study, an external engine deck was used, not the internal capabilities of FLOPS. Therefore, the degradation in engine performance due to compressor bleed to operate HLFC was not modeled. An attempt was made to internally generate an engine, but the attempt failed to accurately model the IHPTET and post-IHPTET technologies.

Table 5.4-2. Technology Simulation in FLOPS

Technology	FLOPS Variables	Baseline Value	Technology Value
post-IHPTET engine	External file name	IHPTET engine deck	Post-IHPTET engine deck
Composites			
Fuselage	FRFU	1.175	1.0
Horizontal Tail	FRHT	1.16	1.0
Vertical Tail	FRVT	1.26	1.0
Nacelles	FRNA	1.06	1.0
Laminar Flow			
Wing	TRUW	Not used	35.0%
	TRLW	Not used	20.0%
Horizontal Tail	TRUH	Not used	40.0%
	TRLH	Not used	40.0%
Vertical Tail	TRUV	Not used	40.0%
	TRLV	Not used	40.0%
Nacelles	TRUN	Not used	50.0%
	TRLN	Not used	50.0%
Landing $C_{L_{max}}$	CLLDM	2.7	3.1
Take-off $C_{L_{max}}$	CLTOM	1.89	2.1

5.4.4. Aerodynamic Analysis Automation

Statistical approaches, such as the Response Surface Methodology (RSM)[16,32] and Design of Experiments (DoE)[14,23] are incorporated into this study and described in detail in Ref. 12. The use of RSM entails a great deal of up-front analysis in the form of repetitive execution of analysis codes to gather enough data for RSE generation. Generation of RSEs for the objective functions can require thousands of analysis cases, necessitating an automated process for aerodynamic analysis and RSE generation. Two main objectives of automation are to remove the possibility of human error during execution and allow execution by a single user. The process must perform the following generic functions:

- Generate the DoE table

- Create the cases needed for that DoE

- Execute the analysis code of interest

- Extract information from the code output needed for RSE generation and put it into a usable format for the statistical package, JMP

A series of UNIX and tk/tcl[34] shell scripts and FORTRAN programs were developed to accomplish these goals. The overall process includes two main programs: a top level DoE

execution script with a Graphic User Interface, called STARS, and the external analysis of the low speed characteristics, called RUN, utilized by the top level. The top level script, STARS, was written originally for the Life Cycle Cost class (AE4353) at Georgia Tech in the fall of 1995. It has been expanded since then to incorporate other features, such as modularity, robustness, and expandability. Each one of these two programs will be described in detail so that they may be utilized in the future by others. Furthermore, all of the scripts and files utilized for this study are contained in alphabetical order in **Appendix A.**

STARS: Top Level Automation

As stated previously, STARS was developed by this author, in collaboration Dr. Mark Hale, to automate the DoE case execution process. It was expanded for the purposes of this study to include the capability of including the low speed aerodynamics needed for the GRA analysis in FLOPS and execution of FLOPS/ALCCA. Within the STARS shell script, there are three sub-components of the main script: the pre-processor, the processor, and the post-processor; shown in Figure 5.4-3.
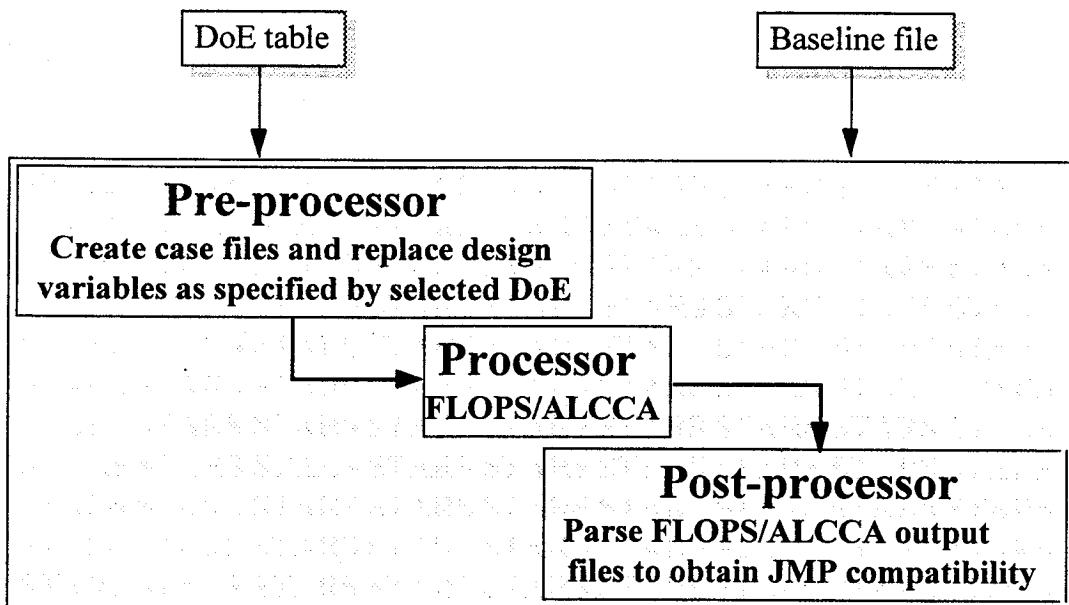
Figure 5.4-3. STARS Script - Top Level Automation

The main script sets up the GUI and allows the user to select from a library of DoE designs. This library is identical to those DoEs as defined in the JMP options. The library of designs is divided into screening tests and RSEs. The screening test options allow for three to thirty one variables, and the response surface options include three to eleven variables. Each option has different DoEs, such as Central Composite, Box-Behnken, and Fractional Factorial. Depending on the response surface design chosen, an appropriate RSE alpha must be inserted, or if a face-centered design is desired, an alpha value of 1.0 is used. This alpha value corresponds to the star point, i.e., axial point, locations defined in JMP. In addition, the user must define the baseline file name where the default is 'baseline'. In this study, a special option was inserted; if the

baseline file name was 'kirby', then STARS utilized other options to incorporate the low speed information.

After the user selects a design, the appropriate DoE table appears. The user then inputs the variables, associated namelist for those variables, a minimum and maximum value, and any prefix required. It should be noted that STARS is namelist driven. The baseline file which is manipulated must have a namelist format. This is required so that STARS can associate a variable name with a given location in the baseline file.

Next, the user clicks the button 'Create Case Files From Above Variables'. STARS copies the baseline file to the DoE determined case files. These files are called 'case#', where the '#' corresponds to the number of cases required by that particular DoE. The GUI used for the generation of the response surfaces is shown in Figure 5.4-4.
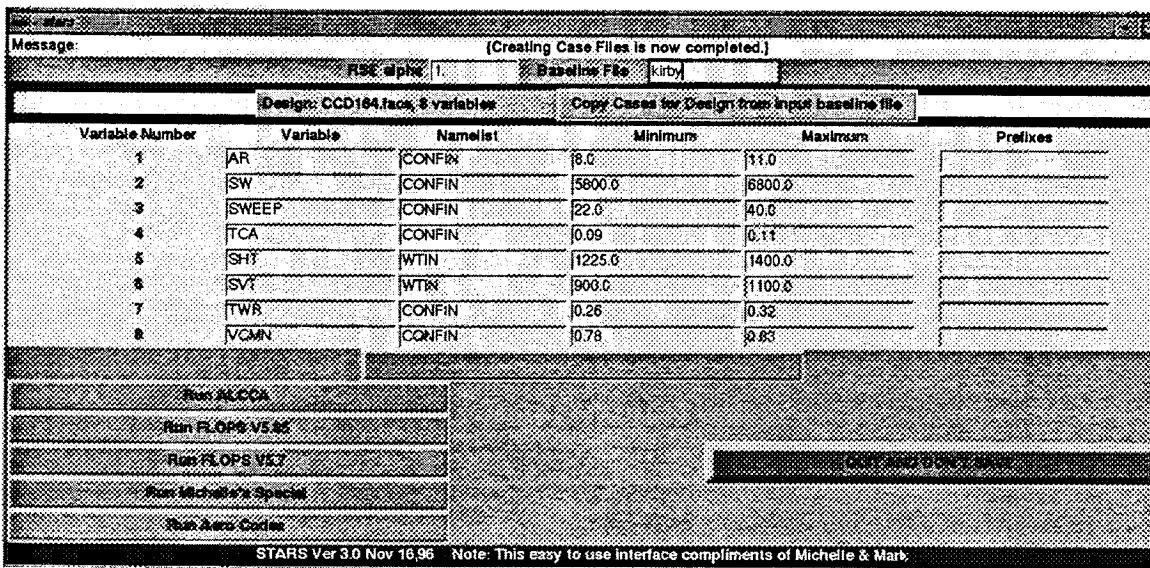
Figure 5.4-4. STARS GUI

Once the variables have been entered, the first subcomponent of STARS is performed. The file 'CreateFiles.tcl' is executed and replaces the design variables entered in the GUI. As stated previously, a special option was inserted for this study. If the baseline file was 'kirby', the low speed information obtained through the shell script 'RUN' was inserted to the appropriate case file. Next, a button labeled 'Run Michelle's Special' is then enabled. The user then clicks on this button, and the processor takes over.

The main processor is contained in a file called 'Kirby.tcl'. This file controls the execution of the FLOPS/ALCCA. Once FLOPS/ALCCA is complete, the post-processor assumes command. The post-processor parses the output from FLOPS/ALCCA into useful information for JMP. The output file, 'case.out#', is parsed for ten different pieces of information: TOGW, empty weight, mission fuel weight, wing weight, LdgFL, VAPP, TOFL, acquisition price, and RDT&E costs. At this point, the process is complete.

It should be noted that the memory required for complete execution can be expensive. As an example, if the user wants to generate an RSE for eight variables, and is using a face-centered Central Composite Design consisting of 145 cases, the total memory requirements is approximately 145*216 KB, as shown in Table 5.4-3. Hence, 31420 KB or 31.4 MB of storage is required. In addition, execution time requires approximately two minutes, which includes creating the files. For the DoE stated, the total CPU time was approximately five hours on an IBM RS-6000.

Table 5.4-3. Memory Requirements per Case

File	Memory (KB)
low#	0.165
case#	6.256
case#.out	210
Miscellaneous	10
Total	226.42

RUN: Low Speed Automation

The low speed information required by FLOPS for this study was established by linking several codes together - RAM, AERO2S, and BDAP through tk/tcl shell script and a FORTRAN code. The geometric modeler, RAM, was utilized for the geometry definition and visualization. AERO2S and BDAP were employed for analyzing the lift induced drag and profile drag, respectively.

As with any analysis code, an input geometry is required along with various analysis control switches and flight conditions. Hence, the first step for any conversion is identifying what one wants and what one has available. AERO2S and BDAP require similar definitions with slight deviations. The geometric modeler, RAM, writes a hermite file for the geometry definition. This hermite file is divided based on the number of component sections of the geometry. Each component is defined by X, Y, and Z coordinates for various cross-sections. This definition is not compatible with the analysis codes. Therefore, a script was written to extract the various component information and convert it into a definition compatible with AERO2S and BDAP. Figure 5.4-5 show the flow of logic for the shell script, RUN. Each portion of the flow chart shown will be described in detail. Please refer to **Appendix A** for detailed file descriptions.
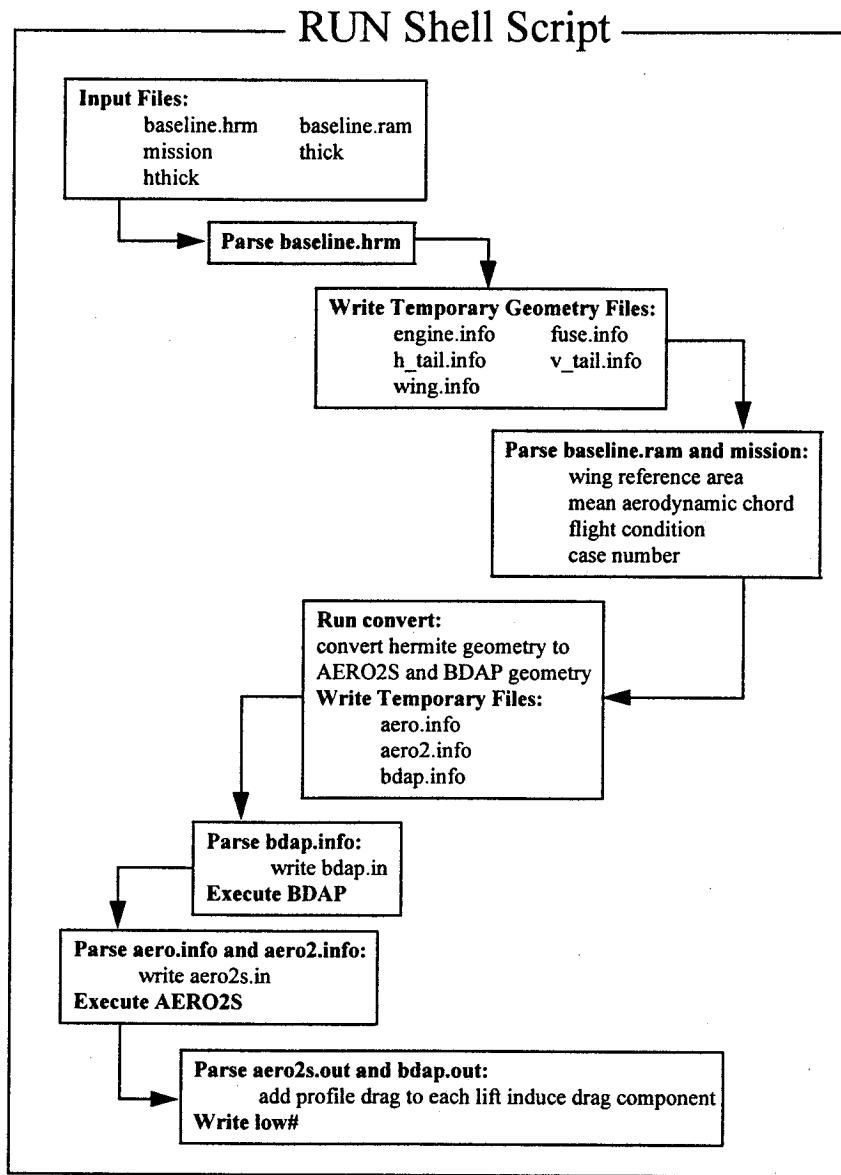
Figure 5.4-5. Flow Path of Logic for Low Speed Shell Script

First, RUN requires five input files: 'baseline.hrm', 'baseline.ram', 'mission', 'thick', and 'hthick'. The ".hrm" file, as stated previously, contains the component geometry definition. The '.ram' file contains geometric parameters such as mean aerodynamic chord and wing reference area. The two files, 'thick' and 'hthick', contain the description of the wing and horizontal tail airfoils. The airfoils are defined by the maximum thickness-to-chord (t/c) ratio, the maximum t/c location in percent chord, and the leading edge radius. And finally, the mission file contains the case number under consideration and the flight condition to be analyzed (i.e. Mach number and altitude).

The first step performed by RUN is extracting the component information from the RAM hermite file. Considering that BDAP and AERO2S only need half of the configuration (if it is symmetric), only the right hand side of the geometry was extracted. The fuselage definition was

withdrawn from the hermite file. This definition was defined by 25 cross-sections and 33 points per cross-section. Next, the engine coordinates were gathered and defined by 16 cross-section and 21 points per cross-section. The wing, horizontal tail, and vertical tail were defined by two cross-sections (root and tip) and 23 points per cross-section. Each one of these 5 components was written to an appropriate file: 'fuse.info', 'engine.info', 'wing.info', 'v_tail.info', and 'h_tail.info'.

Next, the FORTRAN code 'convert' was executed to convert the above definitions to those compatible for AERO2S and BDAP. The code 'convert' was created in such a fashion so that each step taken in the code corresponded to the information needed to define the BDAP input and then the AERO2S input. Please refer to References 22 and 25 for input descriptions.

'Convert' reads the component geometry files created by 'RUN' and stores the X, Y, and Z information into appropriately defined arrays. The fuselage is the first component to undergo conversion. As required by BDAP, only 20 cross-section (specified by x-location) can be used to define the fuselage and each cross-section must be defined by Y and Z locations from bottom to top. Since there were 25 cross-sections from the hermite file, 5 had to be removed. These 5 were chosen such that the curvature description of the fuselage would not be lost. As seen in Figure 5.4-6, the 5 points removed from the fuselage allowed for a linear interpolation between cross-sections and smoothness of curvature was retained. Only 20 points per cross-section were used for the BDAP input. The arrays of X, Y, and Z coordinates were reformatted and written to the file 'bdap.info'.
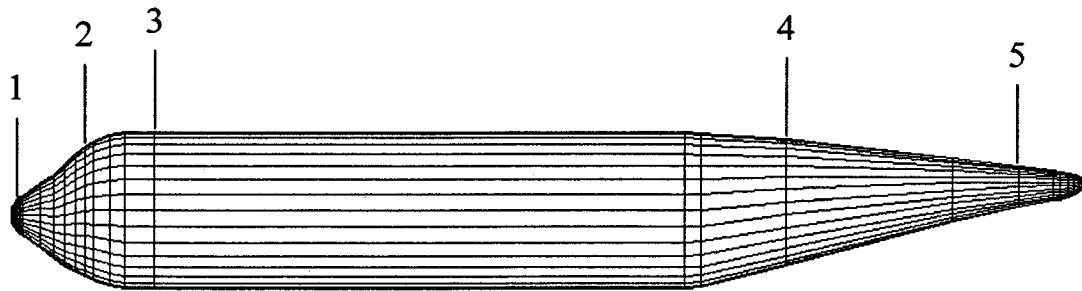
Figure 5.4-6. Fuselage Cross-Section Exclusion

Following the conversion of the fuselage, the wing definition was converted. This component created the most difficulty due to the definition provided by the hermite file, and the definition required by BDAP and AERO2S. Many assumptions were made regarding the conversion of the geometry and will be described in detail. As in the case with the fuselage, the wing geometry file was read and stored in various arrays. As shown in Figure 5.4-7, the definition of the wing was defined by 22 points on the root and tip of the wing. Note, this definition also applied to the horizontal and vertical tail. This definition was not sufficient for BDAP or AERO2S. Actually, the more accurate the solution obtained from these codes is directly proportional to the amount of spanwise station definitions of the wing. Typically, 20 stations is an adequate amount. Hence this original description would need to be expanded from 2 stations to 20.
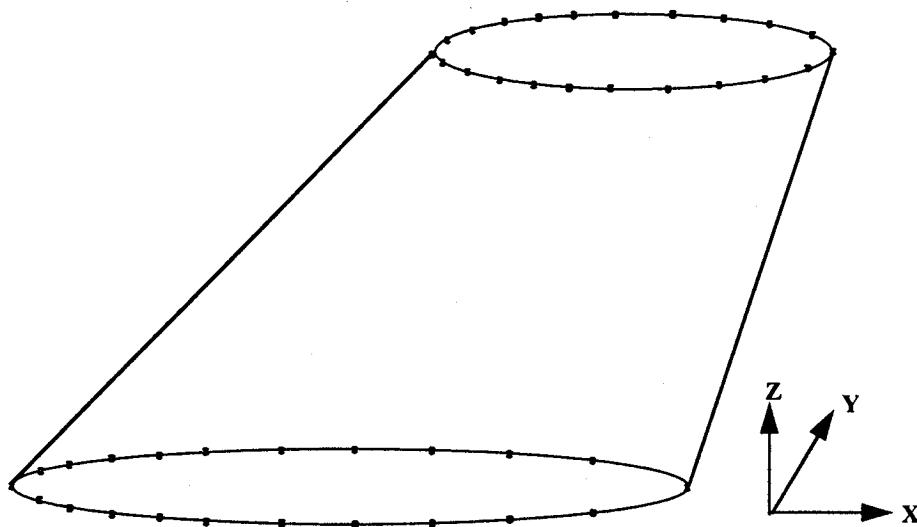
Figure 5.4-7. Hermite Definition of Wing Surface

Furthermore, both AERO2S and BDAP require specific information regarding the airfoil description. BDAP needs z-ordinates and half-thicknesses for at least 10 points per section if a symmetric airfoil is not used; and AERO2S requires z-ordinates. As a result, the wing was divided into 20, equally spaced, spanwise stations. From these stations, four sets of arrays were defined to describe the wing. These arrays are described in Table 5.4-4. The arrays were determined based on linear interpolation from root to tip.

Table 5.4-4. Wing Array Definition

	Array	Description
Wing Leading Edge	WXLE(I)	I = 1,...,20 increasing from root to tip
	WYLE(I)	" "
	WZLE(I)	" "
Wing Trailing Edge	WXTE(I)	I = 1,...,20 increasing from root to tip
	WYTE(I)	" "
	WZTE(I)	" "
Wing Upper Surface	WXUP(I,J)	I = 1,...,20 increasing from LE to TE J = 1,...,10 increasing from LE to TE
	WYUP(I,J)	" "
	WZUP(I,J)	" "
Wing Lower Surface	WXUP(I,J)	I = 1,...,20 increasing from LE to TE J = 1,...,10 increasing from LE to TE
	WYUP(I,J)	" "
	WZUP(I,J)	" "

The above definition of the wing was not sufficient. This was due to the fact that the z-ordinates defining the upper and lower surfaces of each airfoil were not in the same plane. For example, consider Figure 5.4-8 and the airfoil cross-section shown. BDAP requires that the z-ordinates and half t/c associated at those points be defined in the same cross-section cut planes. The only way to achieve a correct definition was to represent the upper and lower surfaces of the wing by curve-fit equations[35]. Therefore, at a given x/c percent location, the z-ordinate and half t/c would be nothing more than a difference of equations representing the upper and lower surfaces.
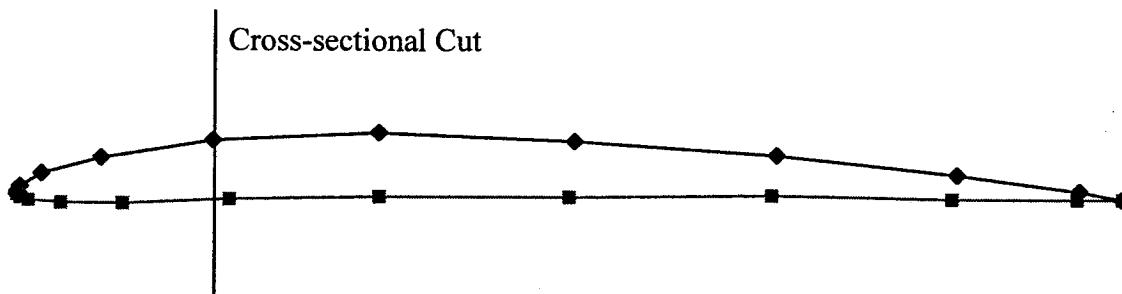
Figure 5.4-8. Cross-sectional Cut of Wing Airfoil

Initially, a quadratic equation representing the entire upper surface was assumed to be adequate. Yet, when the curve-fit approximation was compared to the actual airfoil, the representation was horrendous. As a result, three curve-fits were needed for accurate representation. A cubic curve-fit was needed at the first four point at the leading edge of the airfoil to capture the drastically changing leading edge radius. This cubic equation was valid for

approximately 12% of the airfoil chord length. Two quadratic equations were used for the middle and trailing edges respectively.

Once the coefficients of these polynomials were determined, the z-ordinates and half t/c ratios could be established. Twelve points were assumed to model the airfoils and were defined leading to trailing edge and from the root to tip. These points were defined at 20 stations for the following percent chords: 0, 0.5, 0.75, 2.5, 7.5, 20.0, 30.0, 50.0, 70.0, 85.0, 95.0, and 100. Based on the curve-fit equations and the fixed percent chords, the wing airfoil spanwise stations were defined. Once again, these definitions were formatted and written to the 'bdap.info' file.

The next component definition required by BDAP was the engine. The 'engine.info' file was read and coordinates stored in arrays. BDAP wants the engine defined as different x-stations with a centerline point and a radius corresponding to that point. This was a straight forward calculation based on the midpoint y-location, and the difference of two distantly space z-coordinates. The values were reformatted and written to 'bdap.info'.

The vertical and horizontal tails were straight forward since symmetric airfoils were used. For both components, 12 points were used to define the root and tip airfoils. Since both components used symmetric airfoils, only the half t/c was needed for definition. These values were written to 'bdap.info'.

Ensuing the development of the BDAP geometry conversion, the conversion of AERO2S was quite simple. AERO2S requires leading and trailing edge locations for both the wing and horizontal tail. Based on those locations, a cambered surface is defined between the leading and trailing edges. This is how a fuselage can be simulated; define the nose and tail x-location, and then define the camber as the z-displacement of the fuselage centerline. For this study, 20 spanwise station were used for the geometry definition for AERO2S. The arrays defined for BDAP were reformatted for compatibility with AERO2S. The information was written to two files, 'aero.info' and 'aero2.info'.

The final task of the "convert" code was to determine the Reynolds number. This was achieved by extracting a subroutine in FLOPS which calculates the Reynolds number based on the mean aerodynamic chord and the flight condition. The result was used for AERO2S.

At this time, the shell script RUN resumes command. It extracts the information from 'bdap.info', writes a header containing the control switches for BDAP, inserts the 'bdap.info', inserts more control switches, and then executes BDAP. Next, 'RUN' performs the same steps on 'aero.info' and 'aero2.info' to build the AERO2S input file. 'RUN' then executes AERO2S.

Once the low speed analysis is complete, RUN parses the output files from both codes. From BDAP, the profile drag is extracted; and from AERO2S, the induced drag and lift for ten angles of attack are extracted. RUN then adds the profile drag constant to each term of the induced drag and writes the results to the file 'low#', where the '#' indicates which case is being executed. At this time, the low speed automation shell script is finished.

The file 'low#' is the file needed by STARS and FLOPS. Depending on the case file being created by STARS, the corresponding 'low#' is opened and inserted into the namelist TOLIN in the 'case#' file. To reiterate the importance of the shell script created for the low speed analysis, the rational of the study must be recalled. Three of the system-level metrics under consideration for this study were VAPP, LdgFL, and TOFL. Within FLOPS, these metrics are determined separately from the sizing module. The only information needed are the aircraft weights and low speed drag polars. Therefore, to accurately capture the impact of varying geometry on those three metrics, the correct drag polars had to be generated.

5.5. Identifying Important Design Variables

With the analysis tools identified and the automation process created, the actual aerodynamic study could begin. The reader should recall the purpose of the study: identify a feasible design space of GRA configurations which could be considered for the DMLA initiative.

The first step in any performance investigation is the identification of all pertinent geometric design parameters which could influence the aerodynamic characteristics of the GRA. Figure 5.5-1 depicts the majority of the contributors in a cause and effect diagram. Most of these parameters are inputs to RAM and FLOPS and may be selected for the aerodynamic study. The Ishikawa[36] diagram displayed presents the various design variables which affect the overall system metrics.
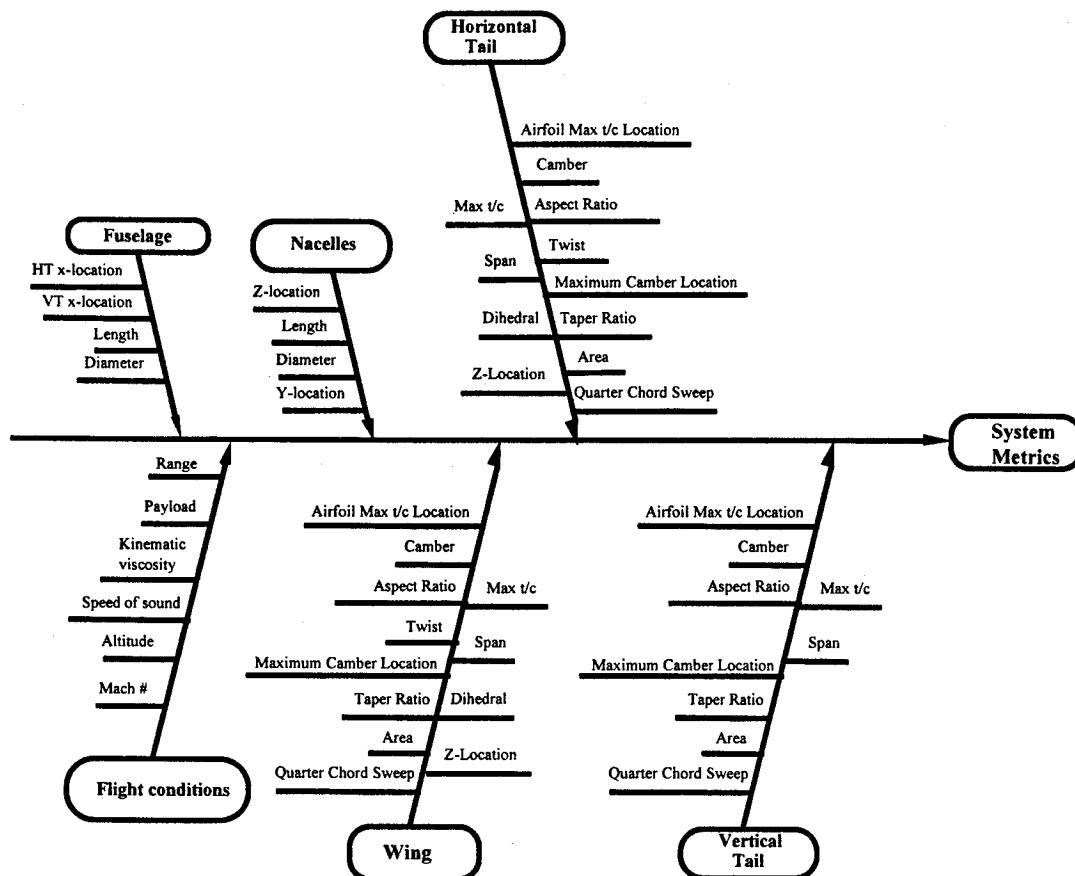
Figure 5.5-1. GRA Aerodynamic Ishikawa Diagram

Many of the variables identified in Figure 5.5-1 were not options for this study. This study focused on opening the design space on a fixed configuration. Therefore, mission parameters, such as payload and range, were fixed. In addition, based on the baseline definition of the GRA, the fuselage length was fixed at 163.5 ft, with a maximum width of 22.7 ft and height of 20.0 ft. Furthermore, an investigation was performed on the actual aerodynamic calculations within FLOPS. It was discovered that the dihedral of the wing and horizontal tail do not affect the results; nor does the actual z-displacement, i.e., the aircraft could be high or low wing, or T-tail or conventional, and the results would not deviate. This could be a source for inaccurate results due to the fact the a high wing has more interference drag from the fuselage and wing interaction. In addition, some of the variables defined are redundant. If the wing aspect ratio (AR) and span are specified, the wing area, SW, can be calculated. Therefore, one of those had to be eliminated as a possibility. Furthermore, based on the baseline aircraft definition, no twist was assumed.

After a down-selection exercise, 17 variables were identified as significant. Through a brainstorming exercise, the ranges for these significant variables were established. Table 5.5-1 presents the ranges for the 17 most significant variables selected in terms of their extreme settings, i.e. minimum and maximum values. These values were established based on an analysis of the VLT baseline. The desire was to capture the geometry of the VLT and, with hope, capture an optimal configuration in those ranges. Also, the cruise Mach number range was dependent on the type of technology added to a configuration.

The cruise Mach number range was dependent upon the level of technology applied to a configuration. For the baseline aircraft and the configurations with composites, the Mach number varied between 0.74 and 0.78. The HLFC technology allowed the cruise Mach number range to increase to 0.78 to 0.83. This range was also applied to the configurations including composites and HLFC. The ranges for the Mach number were determined based on a trial-and-error procedure. Each configuration was sized at various Mach numbers. The highest Mach number at which the configuration could still fly the mission became the upper limit.

Table 5.5-1. Significant Design Variables

Parameter	Minimum	Maximum	Units	Symbol
Wing Aspect Ratio	8.0	11.0		AR
Wing Taper Ratio	0.21	0.28		TR
Wing Reference Area	5800.0	6800.0	ft²	SW
Wing Quarter Chord Sweep	22.0	40.0	degrees	SWEEP
Wing Maximum t/c ratio	0.09	0.11	% chord	TCA
Horizontal Tail Aspect Ratio	3.6	4.2		ARHT
Horizontal Tail Taper Ratio	0.25	0.45		TRHT
Horizontal Tail Reference Area	1225.0	1400.0	ft²	SHT
Horizontal Tail Quarter Chord Sweep	18.0	40.0	degrees	SWPHT
Horizontal Tail Maximum t/c ratio	0.08	0.10	% chord	TCHT
Vertical Tail Aspect Ratio	1.15	1.35		ARVT
Vertical Tail Taper Ratio	0.33	0.63		TRVT
Vertical Tail Reference Area	900.0	1100.0	ft²	SVT
Vertical Tail Quarter Chord Sweep	24.0	50.0	degrees	SWPVT
Vertical Tail Maximum t/c ratio	0.09	0.12	% chord	TCVT
Thrust-to-Weight Ratio	0.26	0.32		TWR
Cruise Mach Number*	0.74*	0.78*		VCMN

* Dependent upon level of technology applied.

These values were input into RAM for the low speed analysis and then into FLOPS/ALCCA in accordance with the DoE table for the screening, or Box-Behnken, format for the RSE. Each row in the DoE table yielded one FLOPS/ALCCA output, e.g., TOGW, based on the input variables and their assigned levels in that row. The ground rules and assumptions established for the aerodynamic analysis are presented in Table 5.5-2.

Table 5.5-2. GRA Aerodynamic Analysis Assumptions

Discipline	Assumptions
Aerodynamics	HLFC simulated by assuming a percent LF over a component surface
	HLFC increases C_{Lmax} at take-off and landing by 11% and 14.8%
	Elliptic lift distribution (Oswald efficiency = 1.0)
	Take-off and landing drag polars generated at an altitude of 1000 ft
	Cruise drag polars generated at an altitude of 45,000 ft
	Fixed airfoil camber and maximum thickness location
	No flaps utilized
Propulsion	No degradation in engine performance from use of HLFC
	Dimension fixed (diameter = 12.855 ft, length = 19.076 ft), thrust scaled
Stability and Control	Not taken into consideration, assumed a stability augmentation system is available
Geometry	Fixed fuselage length of 163.5 ft
	Fixed fuselage diameter and width of 24.7 ft and 20.0 ft
	High wing
	T-tail empennage
Mission	Fixed cruise altitude of 45,000 ft
	Fixed range of 8,000 nm
	Fixed payload of 150,000 lbs
Economics	Default values applied within ALCCA
	Aircraft unit price excludes RDT&E costs

The second step of the GRA aerodynamic study is the development of the equations for the metric responses in terms of geometric variables using the RSM. As previously shown, a three-level DoE for 17 variables requires too many runs to obtain an RSE in a reasonable amount of time. Therefore, based on the Pareto principle, a screening test was conducted using a two-level DoE linear model in order to identify which 7 or 8 of the 17 variables contributed to the responses. After obtaining the response outputs for all level combinations displayed in the DoE tables, an Analysis of Variance, or ANOVA[16], for the main model effects was performed on each configuration metric to determine which variables significantly contributed to the system metrics and constraints. The Pareto plots, Figure 5.5-2 to 5.5-5, displays these contributions for the baseline GRA. The Pareto charts were generated for each metric for each configuration, resulting in 40 Pareto charts. The objective of the Pareto charts was to identify if any of the main contributors were common among the configurations.

The bar chart depicted indicates the relative influence of each variable, while the solid curve represents the cumulative contribution to the response. The Summary of Fit, represented by the R^2 term, estimates the amount of variation in the response around the mean which is explained by the fitted model[33]. Since the "experiments" performed are computer simulations that are 100%

repeatable, fit error is only due to lack of model fit or model error, i.e., the curvature and correlation of parameters not accounted for in the model. As a general rule of thumb, a Summary of Fit, or R^2 value, greater than 80% represents a good model fit[33]. For this study, R^2 values of greater then 95% for all configuration metrics were achieved.

Term	Scaled Estimate
AR	-35989.703
TWR	29226.641
SWEEP	15988.359
TCA	-5405.797
VCMN	-4914.828
SVT	4409.047
SHT	4015.422
TRVT	3526.672
TRHT	3218.641
TCHT	1133.734
TCVT	1095.141
TR	845.328
SW	-412.203
ARVT	205.391
SWPVT	-179.203
ARHT	47.203
SWPHT	40.953
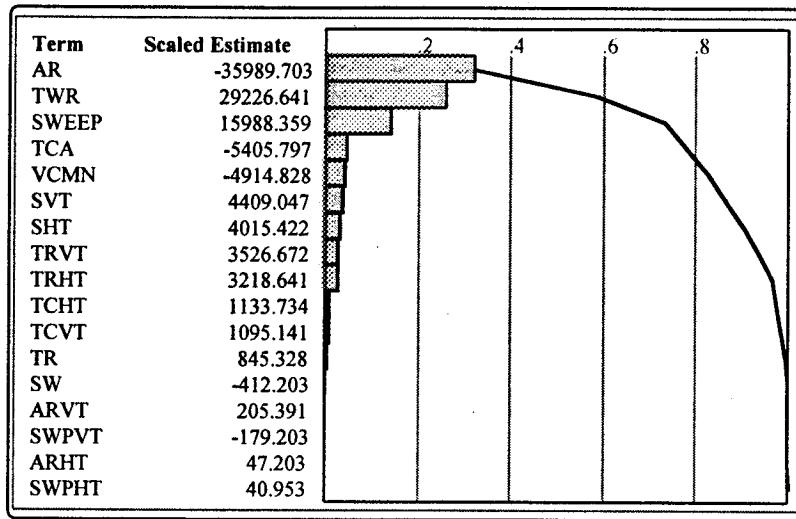
Figure 5.5-2. Pareto Chart for TOGW

As is seen in Figure 5.5-2 for the TOGW for configuration 1, approximately 90% of the response is captured by the first seven variables: wing aspect ratio, AR, thrust-to-weight ratio, TWR, wing quarter chord sweep, SWEEP, wing maximum thickness-to-chord ratio, TCA, cruise Mach number, VCMN, vertical tail area, SVT, and horizontal tail area, SHT. The remaining variables did not contribute significantly to the TOGW variation. In addition, the logic behind the significance of each variables is intuitive. One would expect these variables to be more important on a system level than variables such as the quarter chord sweep of the horizontal and vertical tails. Yet, variables such as the taper ratio of the vertical and horizontal tail appear to have more influence than the wing area. This is a misleading result. The rationale behind the importance of the taper ratios is that the range for these two variables were quite large as compared to the relatively small range of the wing area. As a result, the relative importance was magnified.

Term	Scaled Estimate	.2	.4	.6	.8
AR	-32813.094				
TWR	16437.938				
SWEEP	12812.156				
SW	-5085.687				
VCMN	-4090.281				
SVT	2842.156				
SHT	2528.156				
TRVT	1611.344				
TRHT	1367.625				
TCA	915.063				
TCHT	904.063				
TCVT	881.000				
TR	688.438				
SWPVT	-171.219				
ARVT	160.625				
ARHT	36.250				
SWPHT	33.969				
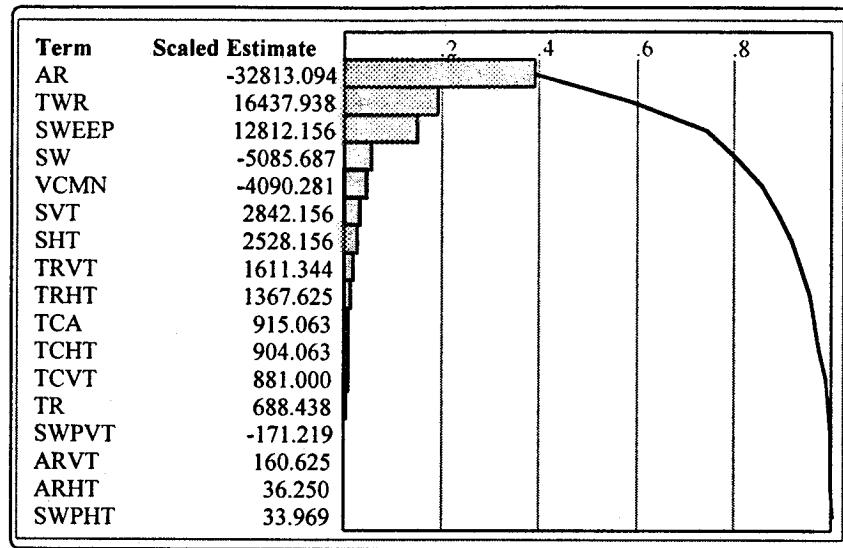
Figure 5.5-3. Pareto Chart for Mission Fuel Weight

In the Pareto chart for the mission fuel weight, the wing area becomes more of a contributor than in the TOGW chart and the TCA importance reduces. This is an expected result. As the surface area of the wing increases, so does the drag. Hence, more thrust is needed and thus more fuel. Yet the AR, TWR, SWEEP, SW, VCMN, and SVT were main contributors once again. For the constraint VAPP below, the wing geometry dominates, since VAPP is a direct function of how much lift the wing can produce. And as expected, the block time is dominated by the cruise Mach number, VCMN.

Term	Scaled Estimate	.2	.4	.6	.8
SW	-4.9171875				
AR	-2.6296875				
SWEEP	1.9328125				
TWR	1.7546875				
TCA	-0.3546875				
VCMN	-0.2890625				
SVT	0.2515625				
TRHT	0.2203125				
TRVT	0.1671875				
SWPHT	0.1328125				
SHT	0.0671875				
ARHT	-0.0546875				
ARVT	-0.0484375				
TCVT	0.0421875				
TCHT	0.0359375				
SWPVT	-0.0296875				
TR	-0.0015625				
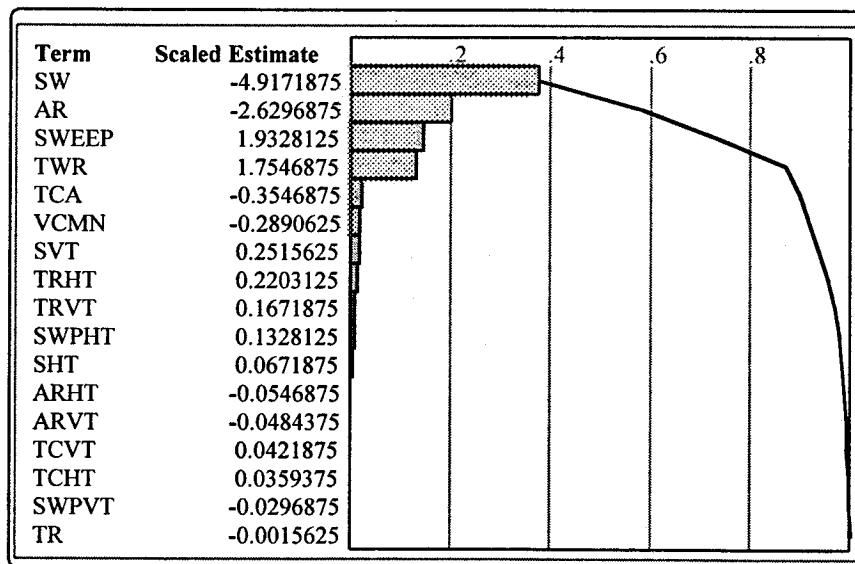
Figure 5.5-4. Pareto Chart for VAPP

Term	Scaled Estimate	.2	.4	.6	.8
VCMN	-0.4067187				
AR	0.17046875				
SW	0.07671875				
SWEEP	-0.0745312				
TWR	-0.0401562				
TCA	0.01609375				
TRVT	-0.0095312				
SVT	-0.0089062				
ARHT	-0.0067187				
SWPVT	0.00578125				
TR	-0.0057812				
SHT	-0.0051562				
TRHT	-0.0045312				
TCVT	-0.0032812				
SWPHT	0.00046875				
ARVT	-0.0001562				
TCHT	-0.0001562				
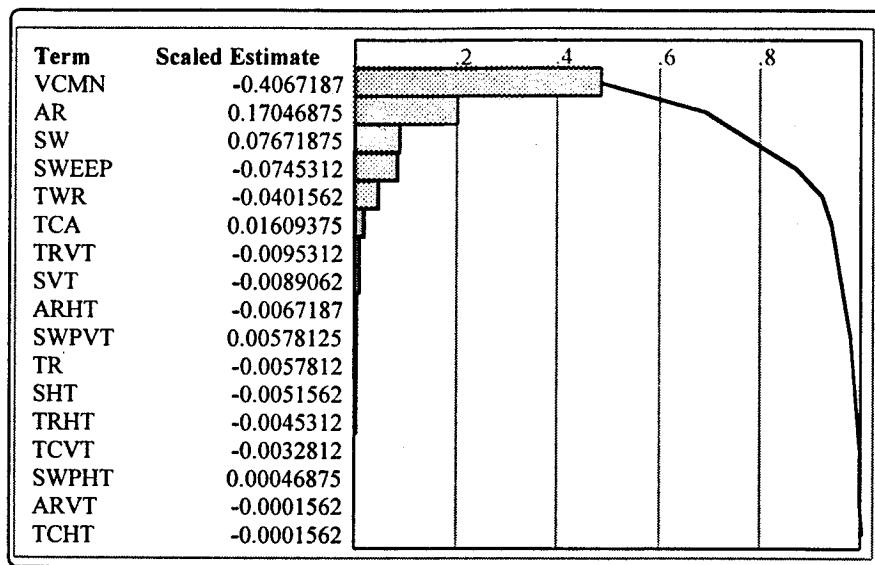
Figure 5.5-5. Pareto Chart for Block Time

As stated previously, a Pareto chart was generated for each configuration metric. Each of these 40 Pareto charts were analyzed and a great deal of commonality was established. In most instances, eight variables constituted approximately 90% of the response. Those variables were AR, SW, SWEEP, TCA, SHT, SVT, TWR, and VCMN. These eight main contributors would constitute the variables used for generating the RSEs for the system-level metrics and constraints. The remaining variables, which contributed very little to the responses, were fixed for the remainder of the study. The fixed values were determined via a prediction profile chart. The prediction profile chart shows the prediction traces for each variable; where the prediction trace is defined as the predicted response in which one variable is changed while the others are held constant. The prediction profile for configuration 1 is shown in Figure 5.5-6. The objective functions and constraints are listed on the left, and the design variables at the bottom. Since one of the main objectives of this study was to minimize the system-level metrics subject to four constraints, the design variables which did not contribute significantly were fixed in the direction which would meet said objectives. The values of the fixed variables are listed in Table 5.5-3. Upon further review of the prediction profiles for each configuration, it was discovered that the fixed variables settings in Table 5.5-3 allowed for an increase of 0.1 in VCMN for configurations 2, 5, 6, and 7.

Figure 5.5-6. Prediction Profile for Configuration 1
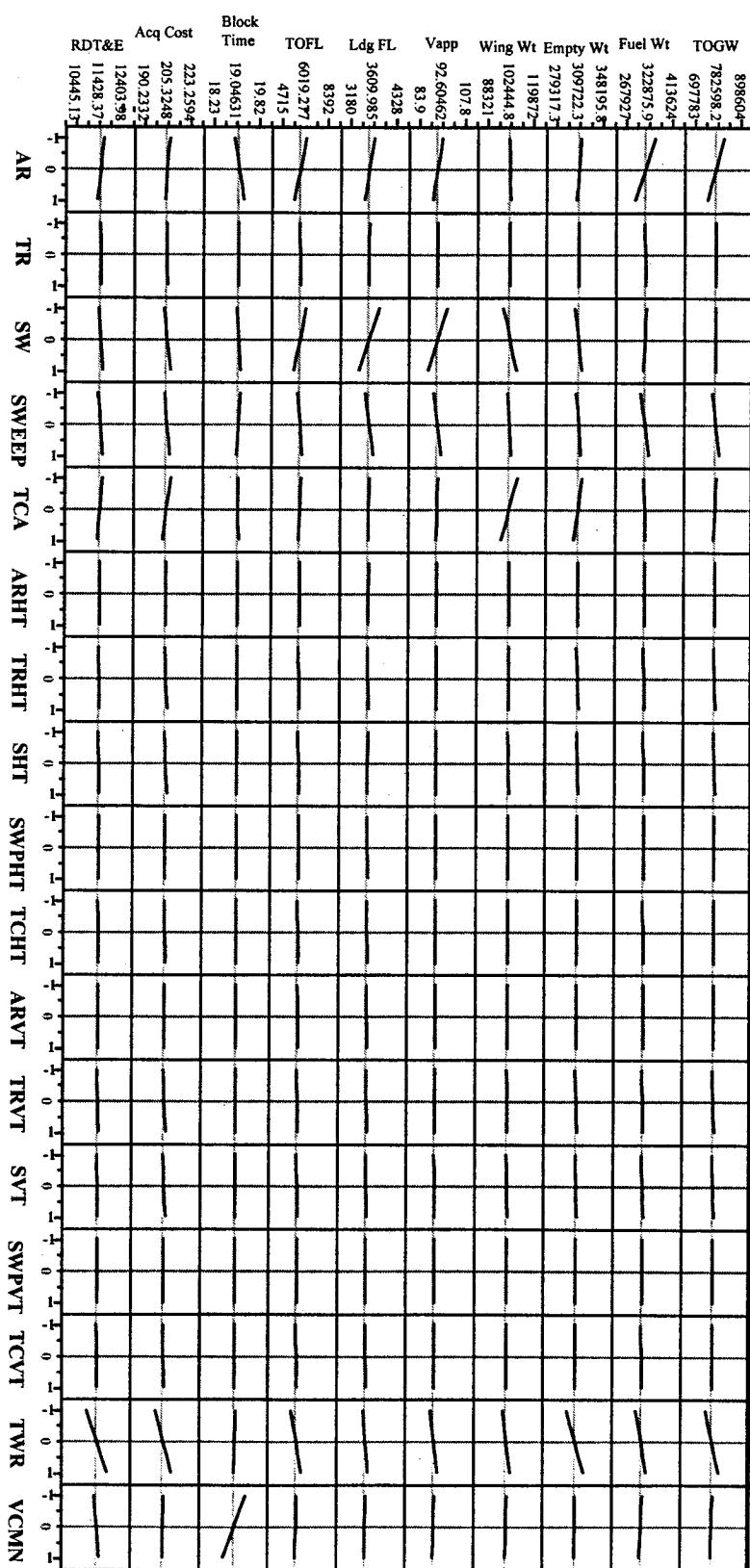
Table 5.5-3. Fixed Design Variables

Design Variable	Fixed Value
Wing Taper Ratio	0.245
Horizontal Tail Aspect Ratio	3.9
Horizontal Tail Taper Ratio	0.25
Horizontal Tail Quarter Chord Sweep	29 deg
Horizontal Tail Maximum Thickness-to-Chord Ratio	0.08
Vertical Tail Aspect Ratio	1.15
Vertical Tail Taper Ratio	0.33
Vertical Tail Quarter Chord Sweep	37 deg
Vertical Tail Maximum Thickness-to-Chord Ratio	0.09

5.6. Response Surface Equation Generation

The eight independent variables defined above were used to generated the RSEs for each configuration metric. The RSEs were generated using a face-centered Central Composite design. The Central Composite design is one of the more popular response surface designs. It combines a two-level fractional factorial with center points (all factor values are at the midrange value) and face-centered axial points (all factors set at midrange value and one factor set at an outer extreme). This design requires 145 cases to be analyzed. Due to the commonality of the design variables for each configuration, the generation of the RSEs for the ten responses could be done simultaneously for a given configuration; that is, one set of 145 cases for a given configuration would result in the 10 metric RSEs of interest. Therefore, a total of 1,160 cases were executed. The total CPU time required for this execution was on the order of 38.7 CPU hours.

Once the cases were generated, a Summary of Fit, R^2, analysis was checked to ensure that the model fit was acceptable. All 40 RSEs generated achieved an R^2 value of at least 98.9%. Since this R^2 value is close to one, it can be assumed that no higher interactions are significant to the responses; therefore, the quadratic representation of the responses was a sufficient estimate.

Furthermore, to ensure that the RSEs were accurately representing the analysis performed by FLOPS/ALCCA, a confirmation test was performed on each RSE. The confirmation test consists of setting the design variables of the RSE to some value and then executing the analysis code at those values and comparing the results. Table 5.6-1 shows the results of the confirmation tests. The negative sign implies that the RSE is under predicting the response compared to the FLOPS/ALCCA output. Typically, a confirmation deviation of less than 5% implies that the RSE is accurately representing the analysis tool. All RSEs listed in Table 5.6-1 were satisfying this criteria except for the VAPP, LdgFL, and TOFL in most instances. This is due to the fact that the low speed drag polars supplied to FLOPS/ALCCA were inserted from the AERO2S/BDAP analysis. There was some slight deviation of the configurations geometric definition in AERO2S and FLOPS. Since AERO2S is a 2-D analysis tool, the accurate prediction of the fuselage lift and drag is not fully realized. This created some deviation in what

FLOPS was expecting and what the analysis tool supplied. Hence, caution should be used when considering the RSEs for VAPP, LdgFL, and TOFL. There is approximately 7 to 8% error in the calculations.

Table 5.6-1. Confirmation of Response Surface Equations

Config.	1	2	3	4	5	6	7	8
RSE								
TOGW	-0.3	-0.7	-0.3	-0.3	-0.5	0.7	-1.8	-0.1
Mission Fuel Weight	-0.7	-1.5	-0.7	-0.8	-1.2	1.9	-4.7	-0.3
Empty Weight	3.2	2.9	3.5	3.5	3.1	3.6	3.4	-0.1
Wing Weight	0.2	-0.5	-0.2	-0.2	-0.4	0.5	-1.4	-0.1
VAPP	7.6	8.9	3.4	8.3	8.5	3.4	7.6	8.4
LdgFL	7.6	8.5	6.3	7.0	7.9	7.5	7.2	7.2
TOFL	0.5	12.1	-1.4	0.1	1.7	3.3	-0.8	1.9
Block Time	0.2	-0.2	-0.02	-0.04	-0.1	-0.1	0.5	0.01
Acquisition Cost	-0.1	-0.3	-0.1	-0.1	-0.2	0.05	-0.2	-0.1
RDT&E	-0.1	0.3	-0.1	-0.1	-0.2	-0.06	0.1	-0.1

6. Feasibility Study Results and Conclusions

The following discussions lead to the overall conclusion that the DMLA concept is well worth pursuing. As the results indicate, the DMLA aircraft are both technically and economically feasible, justifying the continuation of this study.

6.1. Aircraft Performance

At first glance, it could be said that the DMLA concept is technically feasible, given that FLOPS successfully sized the aircraft variants. Inspection of the sizing results validates this claim in the four-engine aircraft cases, yet disproves it for the twin-engine aircraft. As stated before, range and payload were fixed to the values specified by the mission profiles in all cases. For this preliminary design study, the performance metrics of cruise Mach number, fuel required, takeoff gross weight, and required engine thrust remained to gauge technical feasibility.

1. <u>Cruise Mach number</u>. As seen below in Figures 6.1-1 and 6.1-2, comparable performance was achieved between the specialized aircraft and their DMLA counterparts. In fact, greater cruise Mach numbers were achieved in the DMLA aircraft over the specialized aircraft. However, this was caused by somewhat excessive thrust growth from FLOPS when optimizing the DMLA VLT variants, as engine thrust was one of the variables for optimization. The resulting difficulties are considered below with evaluation of the engine thrust metric.
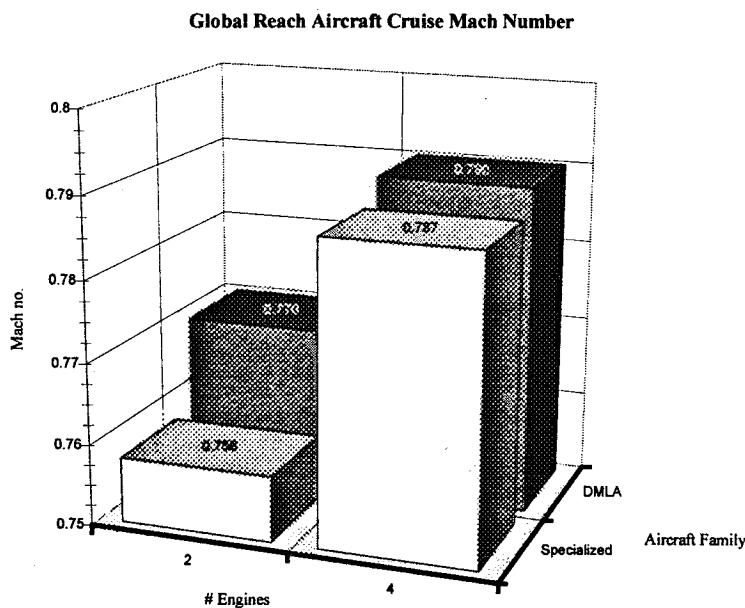
Figure 6.1-1. Cruise Mach Numbers for GRAs.
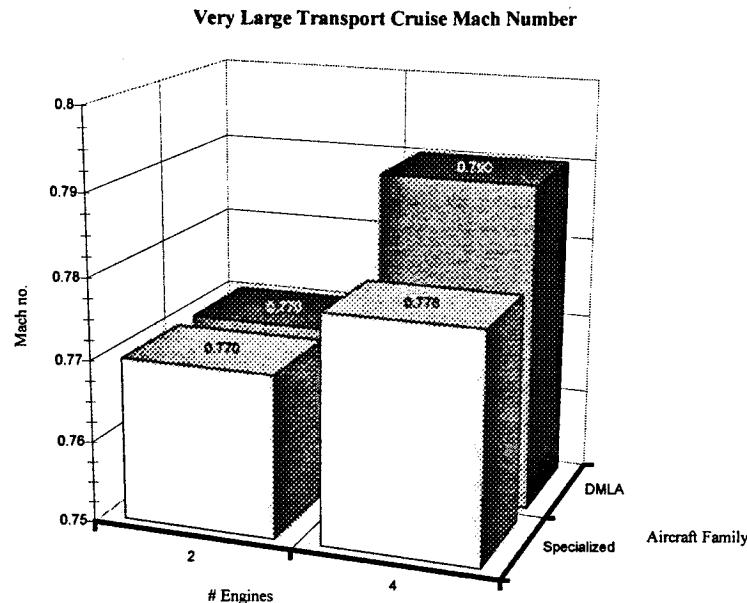
Figure 6.1-2. Cruise Mach Numbers for VLTs.

2. <u>Takeoff gross weight (TOGW)</u>. For both the GRA and VLT aircraft, the DMLA variants
   were heavier than their specialized-aircraft counterparts, as shown in Figures 6.1-3 and 6.1-4
   below. For the VLT aircraft, the weight increase was modest, due mostly to changes in fuel
   and engine weight. The increase in wing fuel capacity to accommodate the GRA mission fuel
   in the DMLA VLT variant caused this component weight increase; thrust growth (discussed
   later) increased the weight of the aircraft propulsion system. The actual fuel and engine
   weight results are provided in Table 6.1-1 below.

Table 6.1-1. VLT Fuel and Engine Weight Increases.

Quantity	Twin-Engine		Four-Engine	
	Specialized	DMLA	Specialized	DMLA
fuel weight (lb.)	474533	481017	456250	502767
engine weight (lb.)	40906	40980	61366	81873

In the GRA aircraft, conversion to the DMLA variants resulted in substantial weight gain.
This is most likely the result of the DMLA being designed for the VLT mission; since the
VLT variant is larger, its internal structure is heavier, making it overdesigned for the GRA
role. Furthermore, for each aircraft type, the larger gross weight increased the engine thrust,
resulting in a heavier engine, which further augmented aircraft gross weight. This becomes
evident upon inspection of Figure 6.1-7, shown later. Additionally, the GRA variants exceed
the weight limit specified in the <u>New World Vistas</u> by 8.75% in the twin-engine aircraft, and
15.96% in the four-engine aircraft. While these violations are tolerable, they raise doubt as to
the technical feasibility of the DMLA concept.

Global Reach Aircraft Takeoff Gross Weight
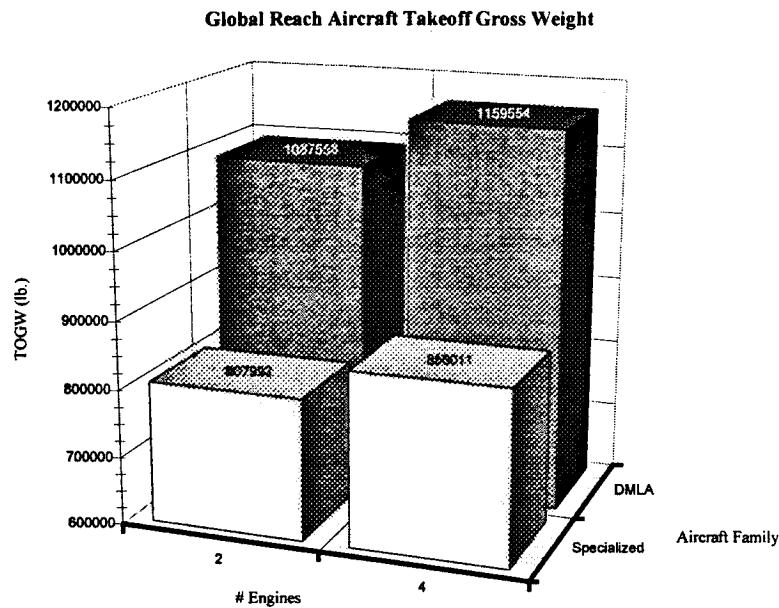
Figure 6.1-3. Takeoff Gross Weights for GRAs.

Very Large Transport Takeoff Gross Weight
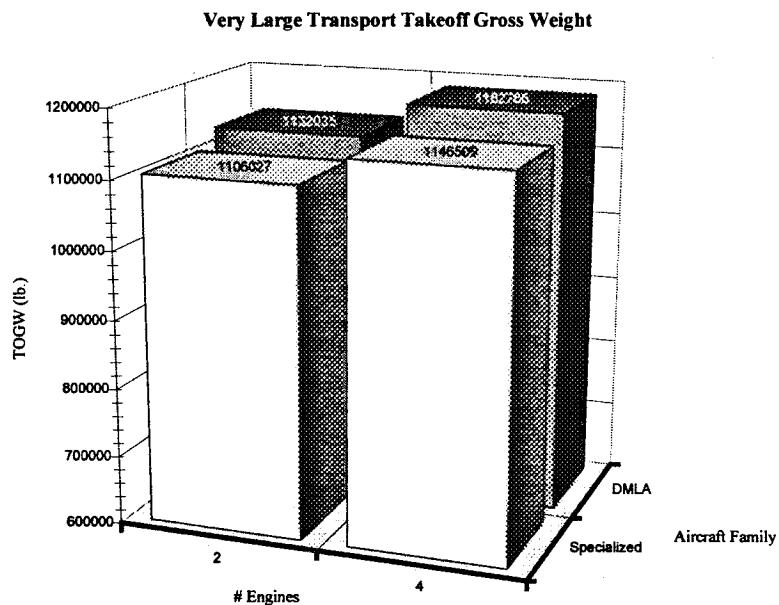
Figure 6.1-4. Takeoff Gross Weights for VLTs.

3. <u>Fuel required</u>. Considerable increases in fuel requirements occurred in creating the DMLA variants for the VLT and GRA missions, depicted below in Figures 6.1-5 and 6.1-6.

Global Reach Aircraft Required Fuel Amounts
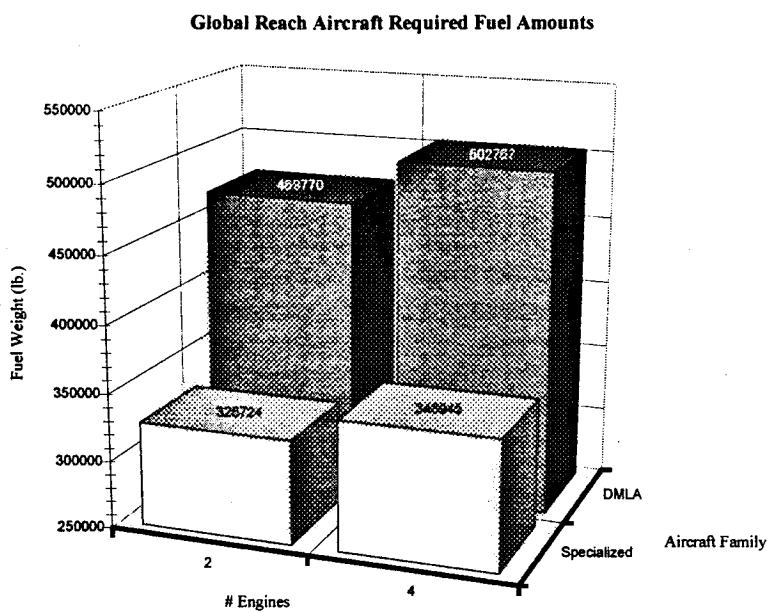
Figure 6.1-5. Fuel Requirements for GRAs.

Very Large Transport Required Fuel Amounts
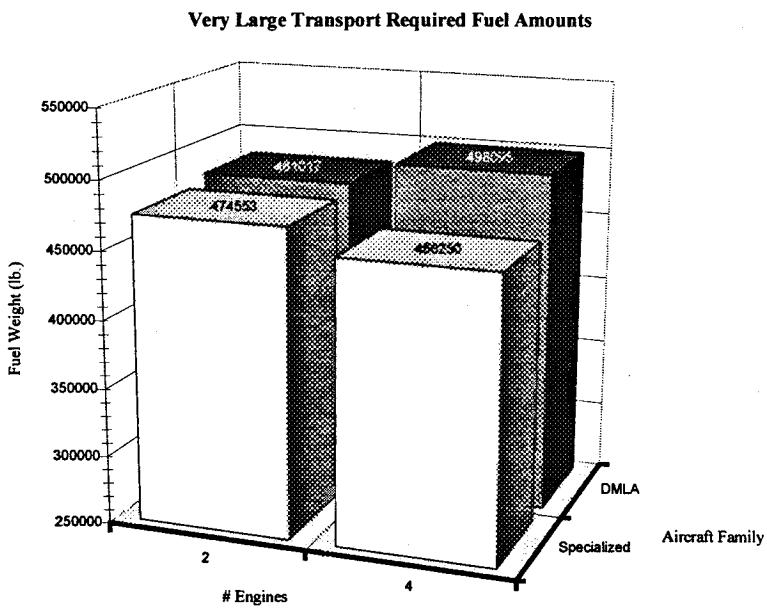
Figure 6.1-6. Fuel Requirements for VLTs.

These increases come from the engine thrust growth (described later); the more powerful engines of the DMLA variants consume more fuel than the specialized aircraft powerplants. This is further substantiated by Figure 6.1-7 below.

Fuel and Thrust Requirements vs. Takeoff Gross Weight
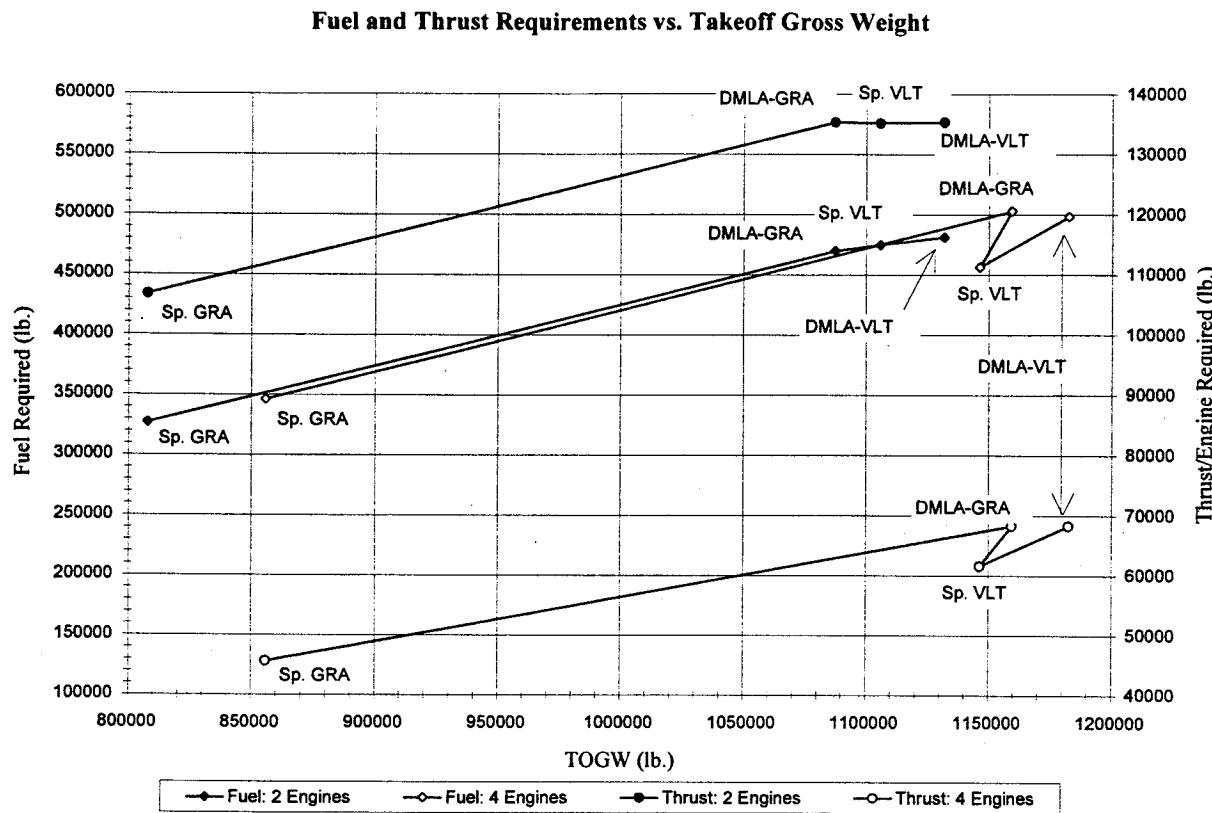
Figure 6.1-7. Fuel and Thrust Increasing with Takeoff Gross Weight.

4. <u>Required engine thrust</u>. As mentioned above, higher Mach numbers came about due to excessive thrust growth, illustrated in Figures 6.1-8 and 6.1-9 below. This creates the problem in the case of the twin-engine DMLA aircraft of extremely high-capacity engines, which are well beyond the capability of current propulsion technology, thereby violating the governing assumption in this study of purely conventional aircraft. It can thus be immediately concluded that *a conventional twin-engine DMLA is not technically feasible.* However, for the four-engine aircraft, the required engine thrust is well within current technological limits. That is, in all four-engine DMLA cases, the engines require much less than 98,000 lb. of sea-level-static thrust currently achieved by the Pratt & Whitney PW4098 powerplant for Boeing's high-capacity models of the 777 transport. (Pratt and Whitney).

Global Reach Aircraft Required Engine Thrust
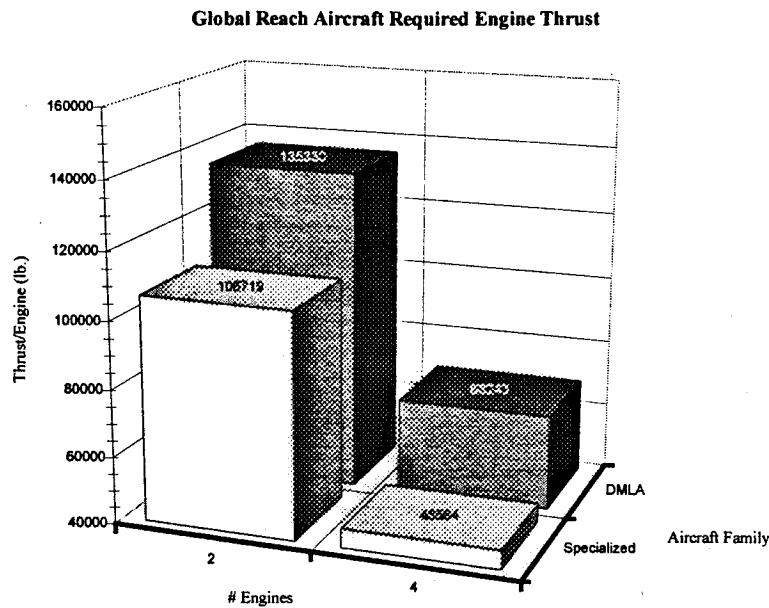
Figure 6.1-8. Required Engine Thrust for GRAs.

Very Large Transport Required Engine Thrust
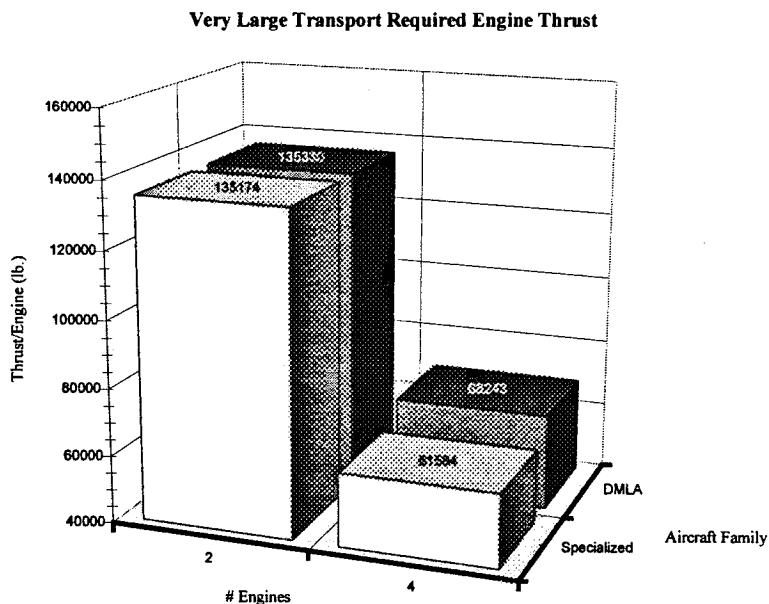
Figure 6.1-9. Required Engine Thrust for VLTs.

Several conclusions result from this analysis. First, the DMLA concept fails for the case of a conventional, twin-engine aircraft family. The engines required are too powerful to exist given current levels of propulsion technology. Aside from this condition, however, the second conclusion is that overall, the DMLA concept is technically feasible. A single DMLA has been successfully sized for both the GRA and VLT roles, yielding aircraft performance that is not unreasonable. The third conclusion, however, deals with the questionable validity of claiming technical feasibility, upon which the high DMLA gross weights cast doubt. DMLA aircraft are

heavier than their specialized counterparts, especially in the GRA role; in this case, the Air Force desire for a new transport weighing under one million pounds has not been satisfied (though not grossly unsatisfied, either). This, and the first statement, lead to the final conclusion: an infusion of new propulsive and structural technologies is clearly needed to completely prove technical feasibility of the DMLA in all cases considered in this study. (The issue of new technologies is addressed later in this report.)

6.2. Aircraft Economics

Inspection of the cost results presented in the following figures, for both the GRA and VLT aircraft, immediately demonstrates the economic feasibility of the DMLA concept.

The most obvious difference is the greatly reduced acquisition costs of the DMLA variants over their specialized counterparts, as illustrated below in Figures 6.2-1 and 6.2-2. The reasons for this lie at the heart of the DMLA concept's nature. First, the combination of a commercial system and a military transport into a single aircraft family also combines their production quantities - 900 DMLA, instead of 549 VLT aircraft and 351 GRA aircraft. This provides a greater number of units over which to distribute the research, development, test, and evaluation (RDT&E) costs that lead to the production of the first unit prototype, so the acquisition cost will be lower. Second, since only one system is designed and produced, then the manufacturer need only invest in one instance of RDT&E costs, instead of two (as would occur for two different, specialized systems). Lower RDT&E costs then, of course, translate into lower acquisition costs for the aircraft customers.
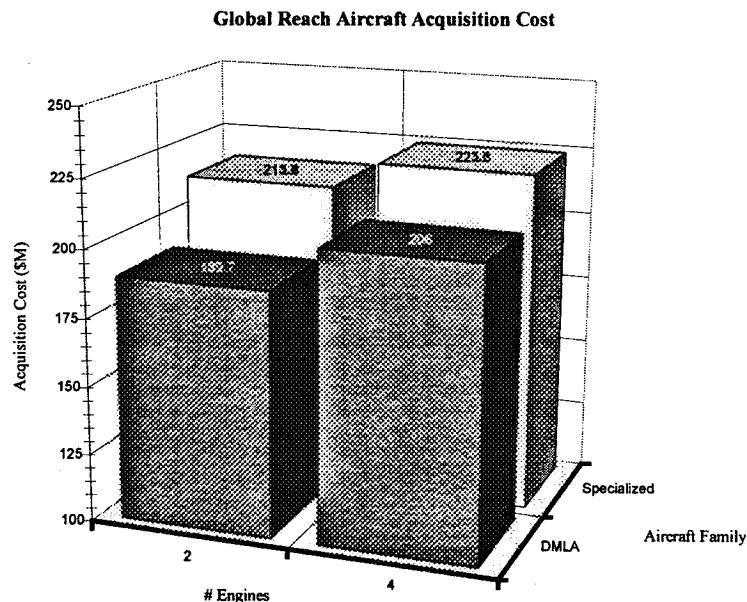
Figure 6.2-1. GRA Acquisition Costs.

Very Large Transport Acquisition Cost
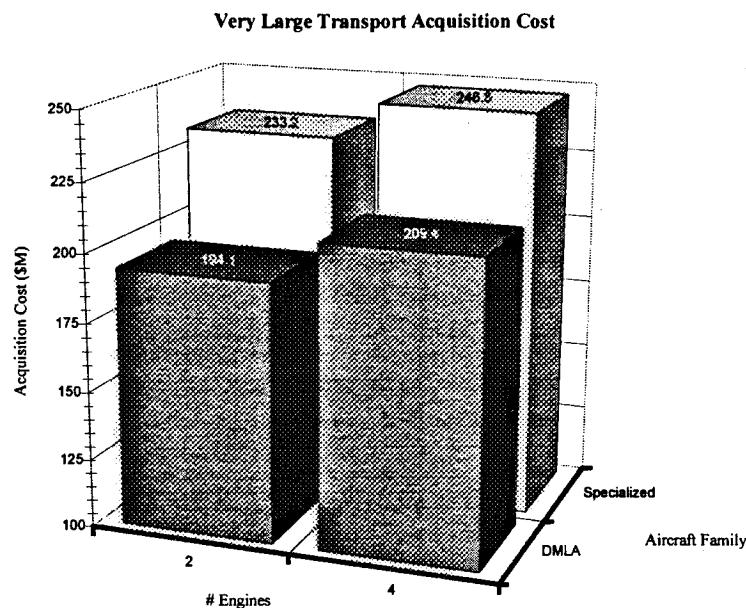
Figure 6.2-2. VLT Acquisition Costs.

Figures 6.2-3 and 6.2-4 depict a less obvious result: DOC for the VLT variant of the DMLA was less than for the specialized VLT, while these costs were higher for the GRA variant. The previously mentioned fact that the DMLA was optimized for the VLT role and subsequently overdesigned for the GRA role impacts economics as well. This characteristic of the DMLA yields an aircraft as complex as the specialized VLT, yet more complex than the specialized GRA. Thus, the GRA variant is more difficult to service and maintain, raising DOC. Furthermore, the larger production quantity realized in the DMLA program also yields a greater number of spares produced, reducing their cost and thus, the customer's maintenance costs. While this is overshadowed by the cost increases due to complexity in the GRA variant, this results in lower DOC for the VLT variant.

Very Large Transport Direct Operating Costs
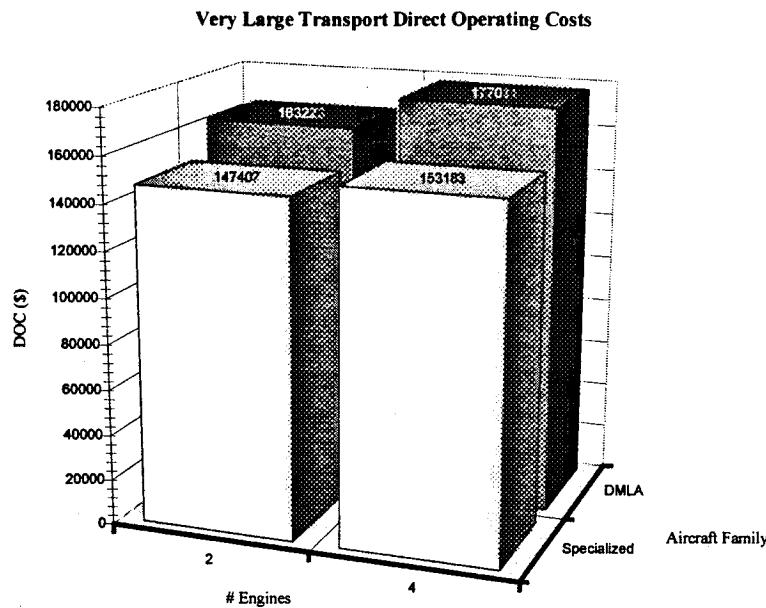
Figure 6.2-3. GRA Direct Operating Costs.

Very Large Transport Direct Operating Costs
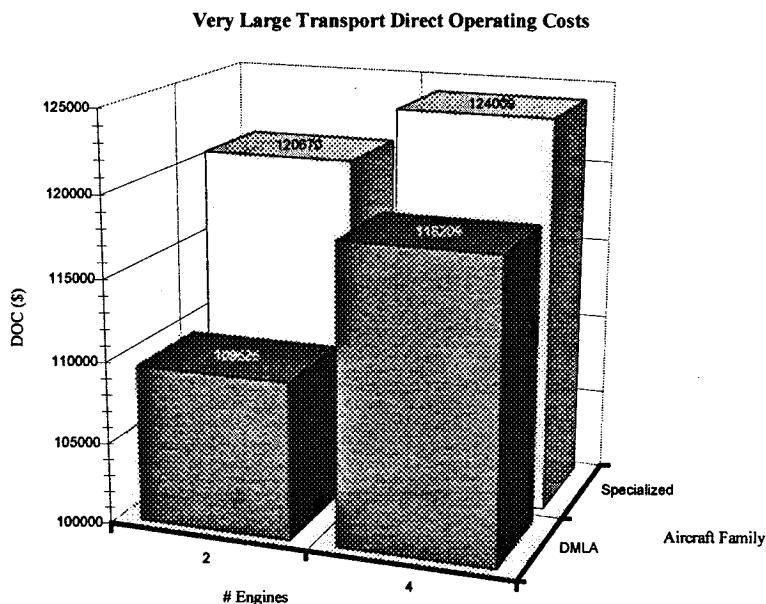
Figure 6.2-4. VLT Direct Operating Costs.

IOC for the specialized aircraft increased only slightly in the DMLA, as shown in Figures 6.2-5 and 6.2-6 below. The small increases stem from the greater gross weight of the DMLA, which would increase ground terminal fees. The increase is greater in the GRA aircraft, since the weight increase from the specialized GRA to the DMLA variant was larger than that for the VLT aircraft.

Global Reach Aircraft Indirect Operating Costs
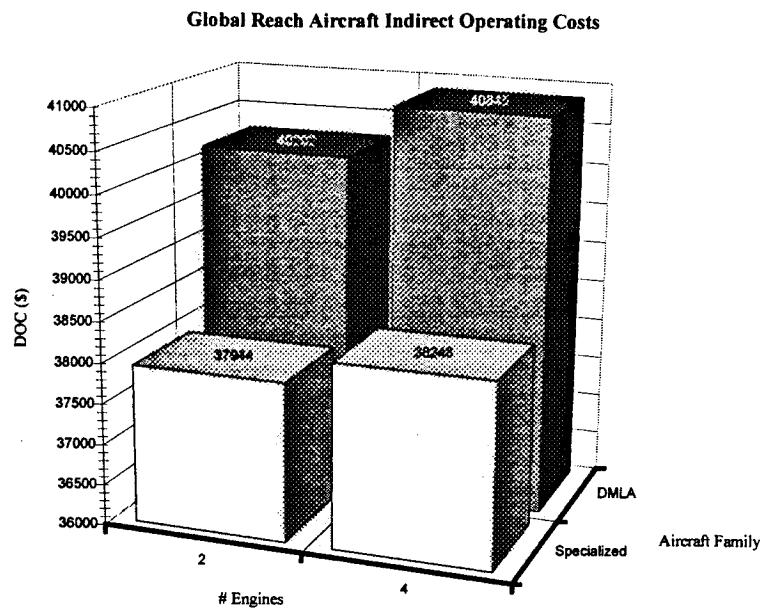
Figure 6.2-5. GRA Indirect Operating Costs.

Very Large Transport Indirect Operating Costs
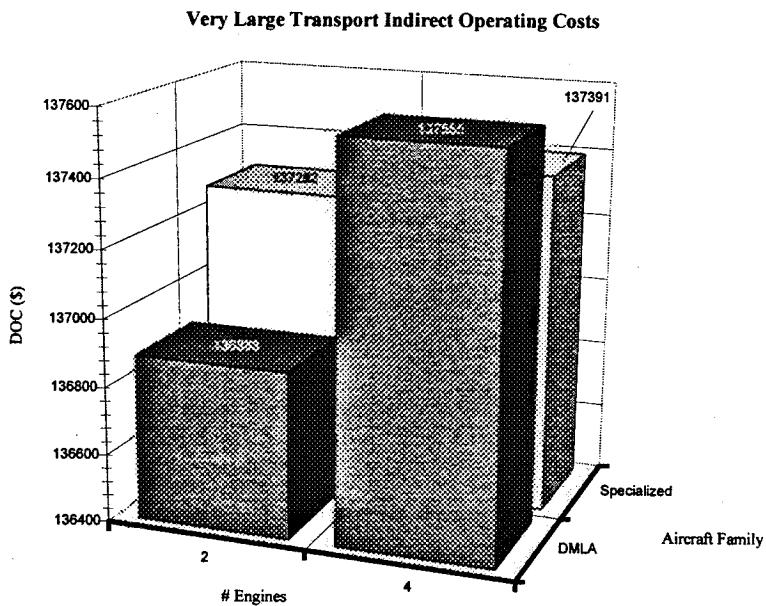
Figure 6.2-6. VLT Indirect Operating Costs.

Overall, however, the DMLA obtained lower total operating costs per trip. TOC, the sum of DOC and IOC, is shown below for all aircraft in Figures 6.2-7 and 6.2-8.

Global Reach Aircraft Total Operating Costs
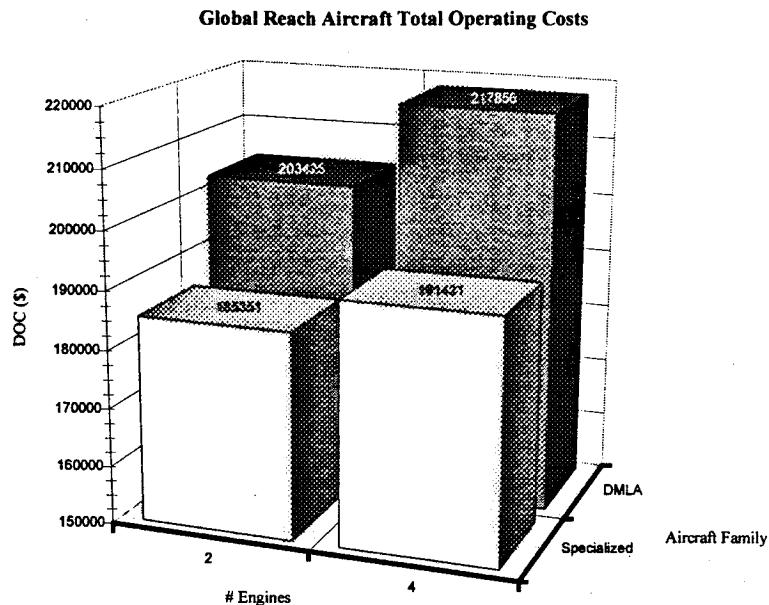
Figure 6.2-7. GRA Total Operating Costs.

Very Large Transport Total Operating Costs
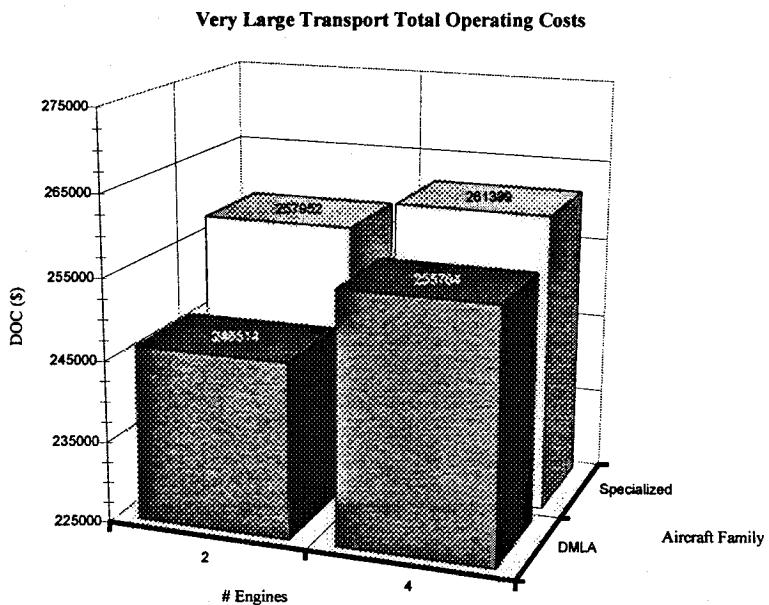
Figure 6.2-8. VLT Total Operating Costs.

Required average yield per revenue passenger-mile is strictly a commercial affordability metric. It is also the most inclusive, capturing the effects of nearly all contributors to airline economics; it is therefore a good measure of an aircraft's affordability. Figure 6.2-9 below depicts $/RPM for the VLT aircraft.

Very Large Transport
Required Avg. Yield per Revenue Passenger-Mile
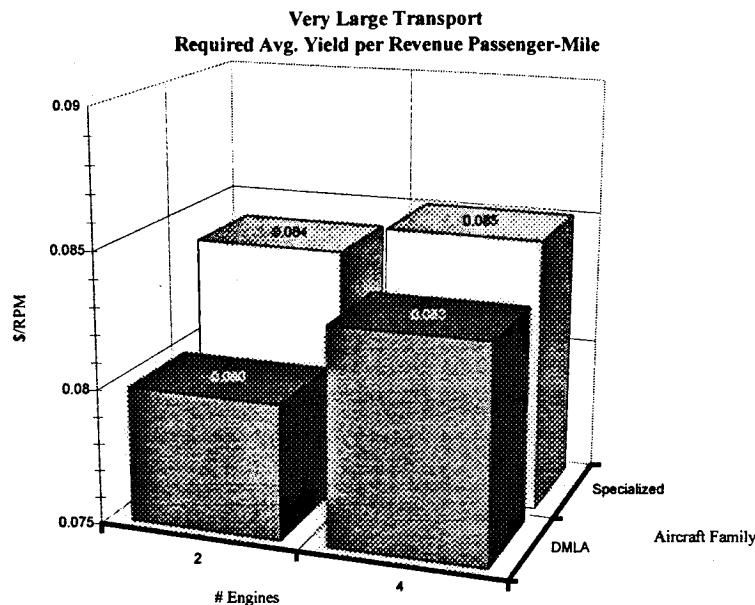
Figure 6.2-9. VLT Required Yields.

Not surprisingly, the VLT variant of the DMLA requires a lower yield than the specialized VLT. Since the acquisition costs and total operating costs per trip are less for the DMLA, then less revenue yield is required for the DMLA-operating airline to break even.

Clearly, DMLA aircraft are more affordable than their dedicated counterparts. The initial acquisition costs are substantially lower; operating cost savings are also realized. Thus, for both cases of aircraft (twin-engine and four-engine), the DMLA concept is economically feasible.

6.3. Effects of Two versus Four Engines

The above results lead to expected conclusions concerning the number of engines installed on the aircraft. Additionally, the following conclusions hold for each category of aircraft (specialized GRA, specialized VLT, GRA variant of DMLA, or VLT variant of DMLA).

From a performance point of view, the four-engine aircraft sized to higher cruise Mach numbers than the two-engine aircraft in each category. Since the powerplant used in the four-engine cases are lower-thrust engines, they possess greater thrust growth potential than the powerplants installed on the twin-engine aircraft. Thus, FLOPS could more easily scale the engines to higher thrust output in order to maximize the cruise Mach number. Also, the twin-engine aircraft emerged from optimization with lower gross weights than their four-engine corollaries. Although the individual engines are heavier in the twin-engine cases, their fewer numbers mean half the redundancy in terms of nacelles, fuel systems, etc. in comparison to the four-engine cases. Furthermore, the twin-engine aircraft required less mission fuel than the four-engine aircraft, no doubt due to the lower gross weight of each twin-engine aircraft.

In terms of cost, the twin-engine aircraft were undoubtedly more affordable in all respects than their four-engine counterparts; that is, ACQ, DOC, and IOC were consistently lower. Fewer engines installed translates into fewer engines for the customer to purchase, lowering aircraft acquisition cost. This also means fewer engines to maintain and fewer spares to purchase, thereby reducing direct operating costs. Additionally, fewer technicians are needed, decreasing indirect operating costs, which are further lowered by the lower terminal fees realized through the lighter twin-engine aircraft.

As a result, it can be concluded that in the scope of this research problem, the sole advantage of a four-engine aircraft is a higher cruising speed in comparison to a twin-engine aircraft. In all other respects, however, the twin-engine aircraft is consistently superior. Such aircraft are lighter, more fuel-efficient, and more affordable. Again, however, it should be noted that the results for the twin-engine aircraft are not easily accepted, as the thrust output of these engines is unrealistic by current technological standards.

7. Aerodynamic Results and Conclusions - Configuration Optimization

7.1. Maximizing the Desirability

Once the RSEs were generated for each configuration, the desirability option within JMP was utilized to determine the optimal settings of the design variables which would minimize the objectives subject to the constraints. This process was performed for all eight configurations. Figure 7.1-1 shows the desirability optimization for configuration 5. The optimization for all other configurations was identical. As described in the Methodology section of this report, if the Mach number was not maximized as a result of the desirability optimization, the Mach number was forced to a maximum and the degradation in the desirability noted. If the degradation in desirability was too large, say 10%, then VCMN would be reduced to approximately a 10% degradation level. All configurations except for 1 and 3 resulted in a slight degradation. The result of the desirability optimization for configurations 1 and 3 was that Mach number was maximized. The reduction of the maximum desirability of the other six configurations was between 0.7% and 7.3% The design variable values which maximized the desirability function are summarized in Table 7.1-1 for each configuration.
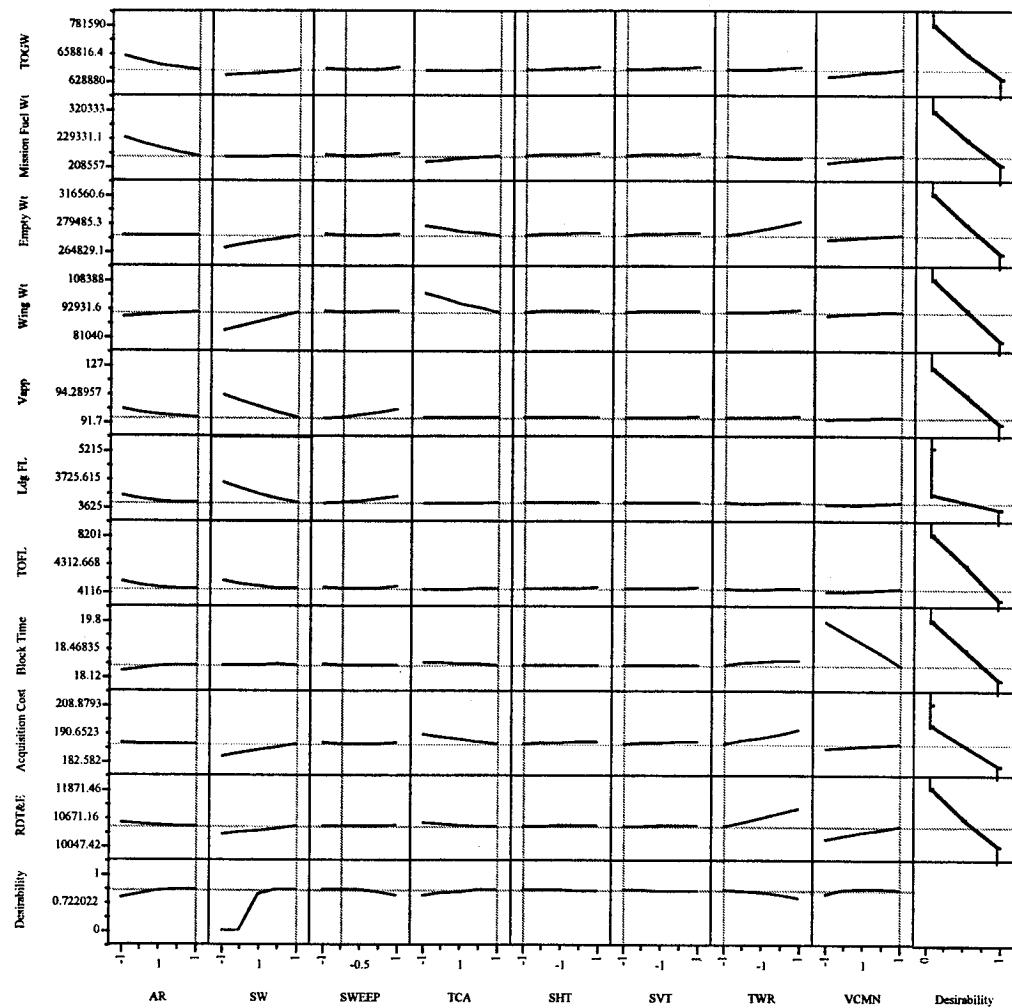
Figure 7.1-1. GRA Configuration 5 Desirability Optimization

Five of the design variable which maximized the desirability for each configuration had common values: wing AR of 11.0, wing TCA of 0.11%, SHT of 1225.0 ft², SVT of 900 ft², and a TWR of 0.26. This result shows commonality of the design space and a gravitation to a global optimum.

Table 7.1-1. Optimal Design Variable Settings for GRA Configurations

Config.	1	2	3	4	5	6	7	8
Design Variable Setting								
AR	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0
SW (ft^2)	6550	6800	6290	6310	6800	6290	6800	6165
SWEEP (deg)	26.3	27.67	24.88	27.13	26.5	31.0	26.59	30.01
TCA	0.11	0.11	0.11	0.11	0.11	0.11	0.11	0.11
SHT (ft^2)	1225	1225	1225	1225	1225	1225	1225	1225
SVT (ft^2)	900	900	900	900	900	900	900	900
TWR	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26
VCMN	0.78	0.84	0.78	0.83	0.79	0.84	0.80	0.83
Desirability	0.813	0.661	0.809	0.854	0.722	0.663	0.669	0.725
Desirability Reduction (%)	0	-5.16	0	-0.77	-1.16	-7.33	-6.6	-3.44

7.2. Optimal Configurations

The values of the design variables in resulted in the minimization of the objective functions, subject to all constraints. The values of the objectives and constraints are summarized in Table 7.2-1. The values shown were calculated by a FLOPS/ALCCA analysis of the optimal design variable settings.

The two primary study objectives were to minimize the objective functions (TOWG, Fuel weight, Empty weight, Wing weight, block time, and RDT&E costs) subject to four constraints (VAPP, LdgFL, TOFL, and Acquisition price) and to maximize the cruise Mach number. As can be seen in Table 7.2-1, all eight optimized configurations met the constraints of VAPP less than 150 knots, LdgFL less than 4,000 ft, and TOFL less than 10,000 ft. From Table 7.1-1, each configuration was able to achieve the maximum value of the cruise number range for the specified level of technology.

Table 7.2-1. Optimal Values of the System Metrics and Constraints

Config.	1	2	3	4	5	6	7	8
TOGW	704683	684428	673928	641648	662253	629538	664643	596609
Mission Fuel Weight (lbs)	269226	250026	260353	233305	232077	209465	250392	197423
Empty Weight (lbs)	276222	275195	254407	249185	270976	260919	255060	249186
Wing Weight (lbs)	95549	95553	90412	87402	93310	86047	93594	81851
VAPP (knots)	93.6	88.7	84.7	78.9	86.9	79.3	87.1	78.4
LdgFL (ft)	3708	3517	3251	3169	3451	3182	3458	3148
TOFL (ft)	4955	3922	4578	3473	4238	3372	4268	3153
Block Time (hrs)	18.86	17.69	18.86	17.88	18.49	17.5	18.08	17.72
Acquisition Price ($M,FY92)	193.63	193.65	181.02	178.68	191.05	186.15	181.56	173.82
RDT&E ($M,FY92)	10815.1	11106.7	10383.2	10492.1	10697.1	10710.1	10418.8	10215.0

    Configurations 6 and 8 achieve the lowest values of the TOGW and Mission fuel weight while achieving a cruise Mach number of 0.84 and 0.83, respectively. Moreover, the configurations which utilized HLFC tended to result in lower objective functions. Consider configuration 1 and 2, the impact of adding HLFC to the clean aircraft resulted in a 2.87% reduction in TOGW and a 7.13% reduction in mission fuel weight. Even though the wing weight and empty weight were comparable in magnitude, the drag reduction on the wing due to HLFC had a tremendous impact on the fuel required to complete the mission. This result could have a tremendous impact on the Direct Operating Costs (DOC) of the aircraft. This same trend is also evident between configuration 5 (post-IHPTET engine) and 6 (post-IHPTET engine with HLFC). Yet, the magnitude of the reduction was increased to 9.7% from the use of a more fuel efficient engine. Table 7.2-2 summarizes the percent improvement of the different configurations from the clean aircraft, configuration 1. Note, the positive values indicate an improvement of the objective function, i.e., a reduction in the objective function.

Table 7.2-2. Percent Reduction in Objective Functions Due to Infusion of Advanced Technologies

Config.	1	2	3	4	5	6	7	8
TOGW	0	2.87	4.36	8.95	6.02	10.66	5.68	15.34
Mission Fuel Weight	0	7.13	3.30	13.34	13.8	22.2	7.00	26.67
Empty Weight	0	0.37	7.90	9.79	1.9	5.54	7.66	9.79
Wing Weight	0	0.0	5.38	8.53	2.34	9.94	2.05	14.34
VAPP	0	5.24	9.51	15.71	7.16	15.28	6.94	16.24
LdgFL	0	5.15	12.32	14.54	6.93	14.19	6.74	15.1
TOFL	0	20.85	7.61	29.91	14.47	31.95	13.86	36.37
Block Time	0	6.2	0.0	5.20	1.96	7.21	4.14	6.04
Acquisition Cost ($M,FY92)	0	-0.01	6.51	7.72	1.33	3.86	6.23	10.23
RDT&E ($M,FY92)	0	-2.7	3.99	2.99	1.09	0.97	3.66	5.55

7.3. Optimal Configurations Performance and Weight Breakdown

The eight optimal configurations defined above were further investigated to determine the detailed component weight breakdown and the performance characteristics. The weights breakdown are summarized in Table 7.3-1 to 7.3-8.

Table 7.3-1. Optimal Configuration 1 Component Weight Breakdown

Component	Weight (lbs)
Structure	207461
Wing	95549
Horizontal Tail	8347
Vertical Tail	6167
Fuselage	75012
Landing Gear	15453
Engine Nacelles and Pylons	6932
Propulsion	43797
Engines	38476
Fuel Systems	4883
Miscellaneous Systems	438
System and Equipment	24963
Surface Controls	5323
Auxiliary Power Systems	1017
Instruments	454
Hydraulic and Pneumatic Equipment	2683
Electrical System	3114
Avionics	3500
Furnishings and Equipment	5448
Air Conditioning	2936
Anti-Icing	488
Empty Weight	276222
Operating Equipment	9195
Operating Weight	285417
Cargo	150000
Zero Fuel Weight	435417
Fuel Weight	269266
Take-off Gross Weight	704683

Table 7.3-2. Optimal Configuration 2 Component Weight Breakdown

Component	Weight (lbs)
Structure	**207096**
Wing	95553
Horizontal Tail	8298
Vertical Tail	6114
Fuselage	75012
Landing Gear	15453
Engine Nacelles and Pylons	6666
Propulsion	42543
Engines	37222
Fuel Systems	4883
Miscellaneous Systems	438
System and Equipment	25555
Surface Controls	5605
Auxiliary Power Systems	1017
Instruments	454
Hydraulic and Pneumatic Equipment	2783
Electrical System	3114
Avionics	3500
Furnishings and Equipment	5448
Air Conditioning	3137
Anti-Icing	498
Empty Weight	**275195**
Operating Equipment	9207
Operating Weight	**284402**
Cargo	150000
Zero Fuel Weight	**434402**
Fuel Weight	250026
Take-off Gross Weight	**684428**

Table 7.3-3. Optimal Configuration 3 Component Weight Breakdown

Component	Weight (lbs)	
Structure	**187821**	
Wing		90412
Horizontal Tail		7132
Vertical Tail		4830
Fuselage		63840
Landing Gear		15453
Engine Nacelles and Pylons		6155
Propulsion	41872	
Engines		36551
Fuel Systems		4883
Miscellaneous Systems		438
System and Equipment	24714	
Surface Controls		5121
Auxiliary Power Systems		1017
Instruments		454
Hydraulic and Pneumatic Equipment		2649
Electrical System		3114
Avionics		3500
Furnishings and Equipment		5448
Air Conditioning		2936
Anti-Icing		475
Empty Weight	**254407**	
Operating Equipment	9168	
Operating Weight	**263575**	
Cargo	150000	
Zero Fuel Weight	**413575**	
Fuel Weight	260353	
Take-off Gross Weight	**673928**	

Table 7.3-4. Optimal Configuration 4 Component Weight Breakdown

Component	Weight (lbs)
Structure	**184274**
Wing	87402
Horizontal Tail	7062
Vertical Tail	4759
Fuselage	63840
Landing Gear	15453
Engine Nacelles and Pylons	5758
Propulsion	39871
Engines	34550
Fuel Systems	4883
Miscellaneous Systems	438
System and Equipment	25039
Surface Controls	5217
Auxiliary Power Systems	1017
Instruments	454
Hydraulic and Pneumatic Equipment	2706
Electrical System	3114
Avionics	3500
Furnishings and Equipment	5448
Air Conditioning	3103
Anti-Icing	480
Empty Weight	**249185**
Operating Equipment	9158
Operating Weight	**258343**
Cargo	150000
Zero Fuel Weight	**408343**
Fuel Weight	233305
Take-off Gross Weight	**641648**

Table 7.3-5. Optimal Configuration 5 Component Weight Breakdown

Component	Weight (lbs)
Structure	**204495**
Wing	93310
Horizontal Tail	8244
Vertical Tail	6054
Fuselage	75012
Landing Gear	15453
Engine Nacelles and Pylons	6422
Propulsion	41387
Engines	36066
Fuel Systems	4883
Miscellaneous Systems	438
System and Equipment	25094
Surface Controls	5372
Auxiliary Power Systems	1017
Instruments	454
Hydraulic and Pneumatic Equipment	2728
Electrical System	3114
Avionics	3500
Furnishings and Equipment	5448
Air Conditioning	2970
Anti-Icing	493
Empty Weight	**270976**
Operating Equipment	9201
Operating Weight	**280177**
Cargo	150000
Zero Fuel Weight	**430177**
Fuel Weight	232076
Take-off Gross Weight	**662253**

Table 7.3-6. Optimal Configuration 6 Component Weight Breakdown

Component	Weight (lbs)
Structure	**196606**
Wing	86047
Horizontal Tail	8161
Vertical Tail	5962
Fuselage	75012
Landing Gear	15453
Engine Nacelles and Pylons	5970
Propulsion	39231
Engines	33910
Fuel Systems	4883
Miscellaneous Systems	438
System and Equipment	25083
Surface Controls	5208
Auxiliary Power Systems	1017
Instruments	454
Hydraulic and Pneumatic Equipment	2714
Electrical System	3114
Avionics	3500
Furnishings and Equipment	5448
Air Conditioning	3137
Anti-Icing	491
Empty Weight	**260919**
Operating Equipment	9154
Operating Weight	**270073**
Cargo	150000
Zero Fuel Weight	**420073**
Fuel Weight	209465
Take-off Gross Weight	**629538**

Table 7.3-7. Optimal Configuration 7 Component Weight Breakdown

Component	Weight (lbs)
Structure	**190475**
Wing	93594
Horizontal Tail	7112
Vertical Tail	4810
Fuselage	63840
Landing Gear	15453
Engine Nacelles and Pylons	5667
Propulsion	39406
Engines	34085
Fuel Systems	4883
Miscellaneous Systems	438
System and Equipment	25178
Surface Controls	5414
Auxiliary Power Systems	1017
Instruments	454
Hydraulic and Pneumatic Equipment	2739
Electrical System	3114
Avionics	3500
Furnishings and Equipment	5448
Air Conditioning	3003
Anti-Icing	490
Empty Weight	**255060**
Operating Equipment	9190
Operating Weight	**264250**
Cargo	15000
Zero Fuel Weight	**414250**
Fuel Weight	250392
Take-off Gross Weight	**664643**

Table 7.3-8. Optimal Configuration 8 Component Weight Breakdown

Component	Weight (lbs)	
Structure	178000	
Wing		81851
Horizontal Tail		6960
Vertical Tail		4656
Fuselage		63840
Landing Gear		15453
Engine Nacelles and Pylons		5240
Propulsion	37221	
Engines		31900
Fuel Systems		4883
Miscellaneous Systems		438
System and Equipment	24831	
Surface Controls		5026
Auxiliary Power Systems		1017
Instruments		454
Hydraulic and Pneumatic Equipment		2687
Electrical System		3114
Avionics		3500
Furnishings and Equipment		5448
Air Conditioning		3103
Anti-Icing		482
Empty Weight	240052	
Operating Equipment	9134	
Operating Weight	249186	
Cargo	150000	
Zero Fuel Weight	399186	
Fuel Weight	197423	
Take-off Gross Weight	596609	

From the component weight breakdowns above, main differences were identified. The most prominent difference was the fuel weight for all configurations except for 3 and 7 where the structural weight reduced. The reduction in mission fuel required is attributed directly to the technology infused to the configuration. As stated previously, the reduction in fuel translates to a reduction in DOC. This is an attractive feature regardless of military or commercial use.

The aerodynamic performance characteristics of each configuration are summarized in the form of drag polars in Figure 7.3-1 (configurations 1 to 4) and Figure 7.3-2 (configurations 5 to 8). As a result of the commonality of 5 design variable values, the drag polars are quite similar. The slight differences are coming through the variations in SW and VCMN. In Figure 7.3-1, configurations 1 and 3 had higher drag at lower lift coefficients. Consequently, the two aircraft tended to fly at a higher C_L but with a reduced maximum L/D ratio.
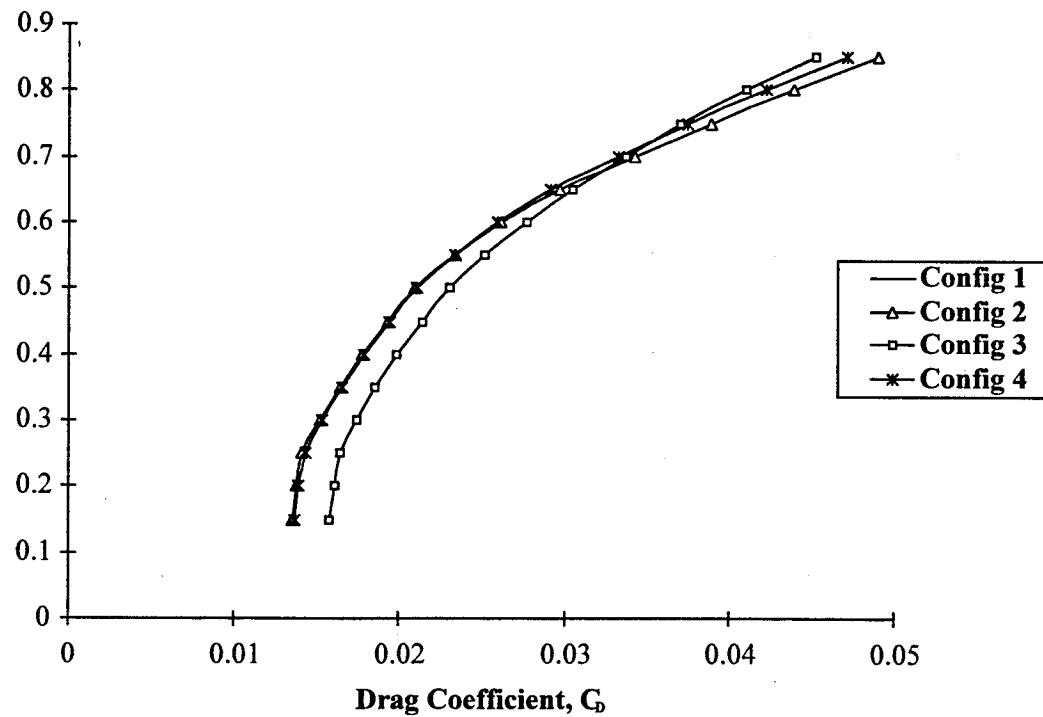
Figure 7.3-1. Optimal GRA Configurations (1 to 4) Drag Polars

In Figure 7.3-2, configurations 5, 6, and 8 resulted in higher lift at lower drag. Actually, configurations 6 and 8 produced the highest L/D ratio and cruised at the highest Mach numbers, 0.84 and 0.83, respectively
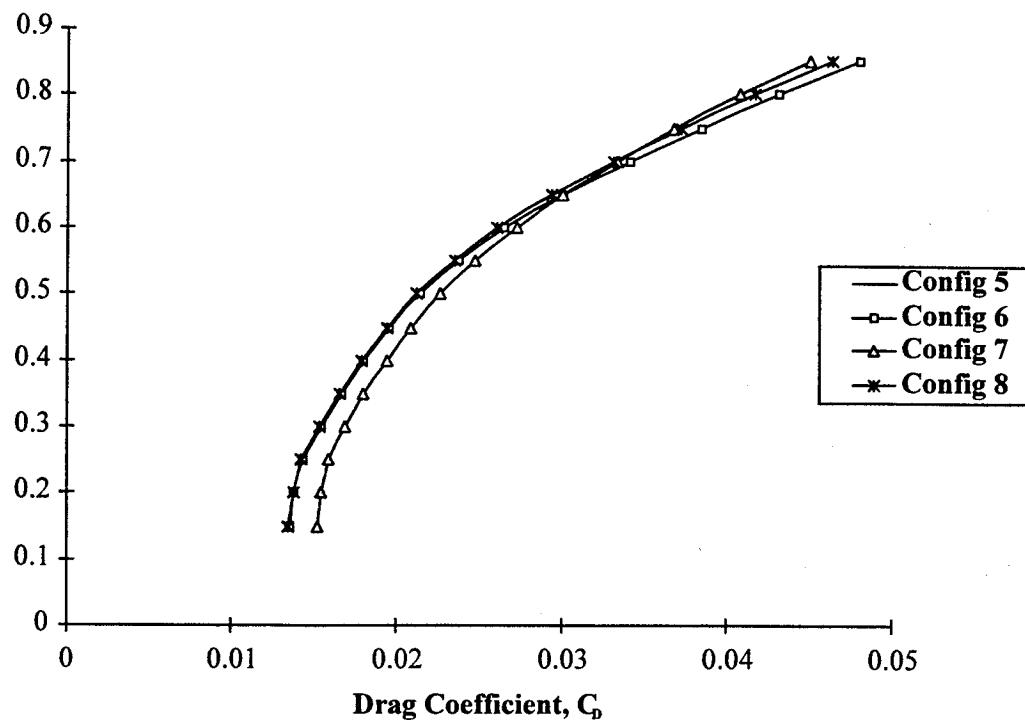
Figure 7.3-2. Optimal GRA Configurations (5 to 8) Drag Polars

From an aerodynamics point of view, a maximum value of L/D is desired. As is evident from Figure 7.3-3 and 7.3-4, configurations 2, 4, 6, and 8 resulted in the highest values. All four of these aircraft employed boundary layer control through the use of HLFC technology. The percent improvement from the original GRA baseline L/D is summarized in Table 7.3-9. The most drastic improvement was form configuration 2 with a 3.97% increase, and configuration 4 pulled a close second. These two aircraft surpassed 6 and 8 due to the slightly lower wing quarter chord sweep. Furthermore, the maximum desirability value for 2 and 4 were higher as compared to 6 and 8. Hence, the degradation in the desirability function for configurations 6 and 8 directly translated to a reduction in the maximum obtainable L/D ratio. It was felt that this loss was insignificant as compared to the benefits of reduced fuel weight through the addition of the post-IHPTET engine technology. In other words, the trade-off of loss in maximum performance for lower fuel weight seemed inconsequential.
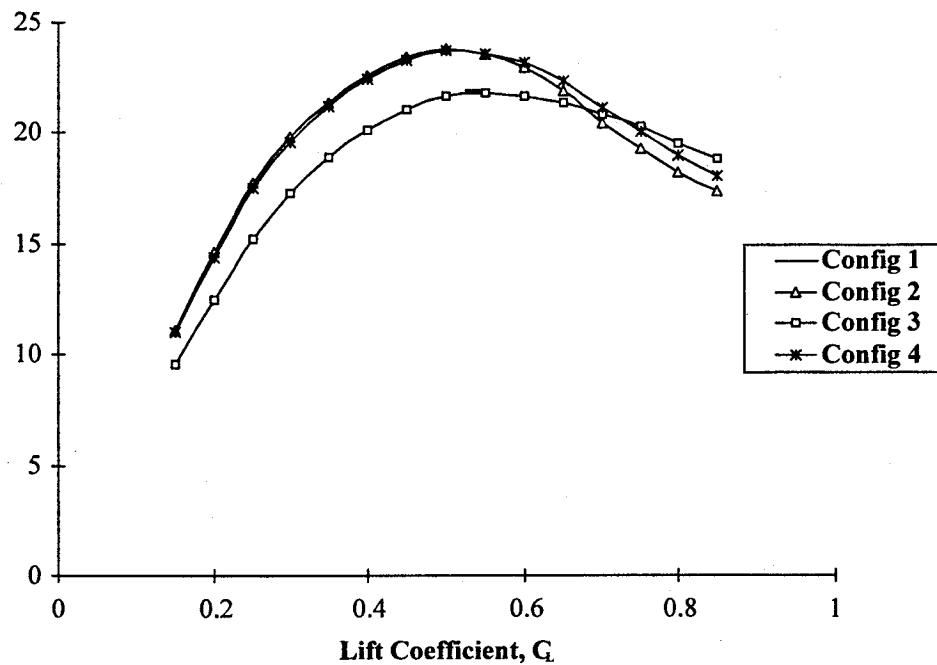
Figure 7.3-3. Optimal GRA Configurations (1 to 4) Lift-to-Drag Ratios
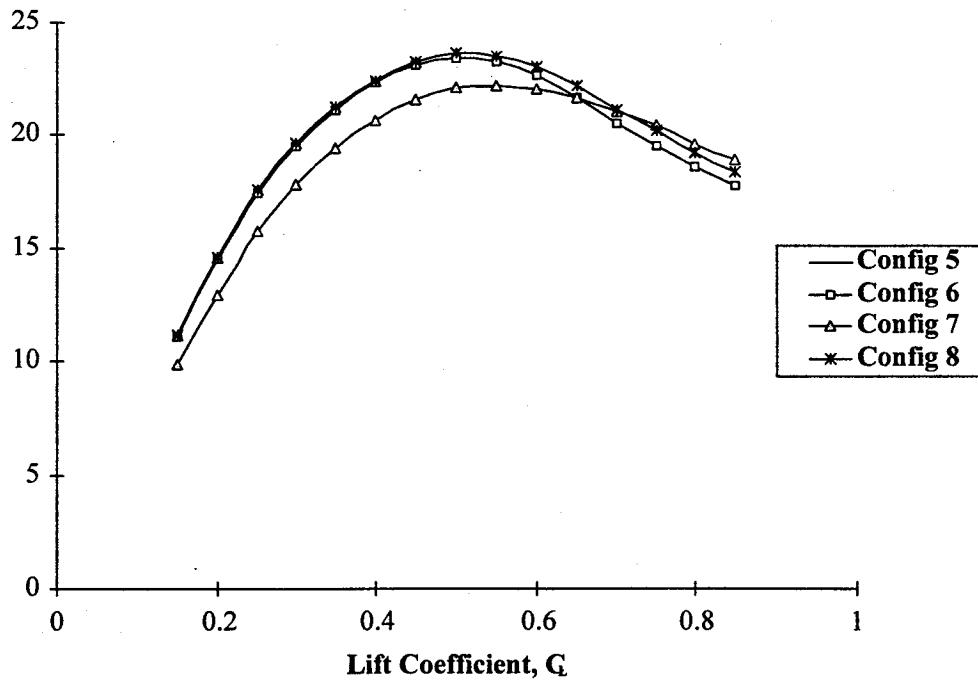
Figure 7.3-4. Optimal GRA Configuration (5 to 8) Lift-to-Drag Ratios

Table 7.3-9. Maximum Cruise Lift-to-Drag Ratios

Configuration	Maximum Cruise L/D	D% FLOPS Baseline
1	21.95	-4.15
2	23.81	3.97
3	21.81	-4.76
4	23.7	3.49
5	22.19	-3.1
6	23.43	2.31
7	22.2	-3.06
8	23.61	3.10

The low speed performance of each configuration was also investigated. The drag polars displayed in Figure 7.3-5 and 7.3-6 were generated by the low speed shell script 'RUN'. Typically for landing, an aircraft needs high L/D, but also a high drag to reduce the VAPP and LdgFL. Configurations 3, 4, 6, and 8 achieve this and have the lowest values of VAPP and LdgFL. For take-off, the goal is to maximize the amount of lift the wing can produce. Configurations 2, 4, 6, and 8 had the lowest TOFL. This would seem contrary to the LdgFL results except for the fact that the maximum C_L obtained by these configurations was enhanced to a value 2.1 to simulate the effects of HLFC technology.
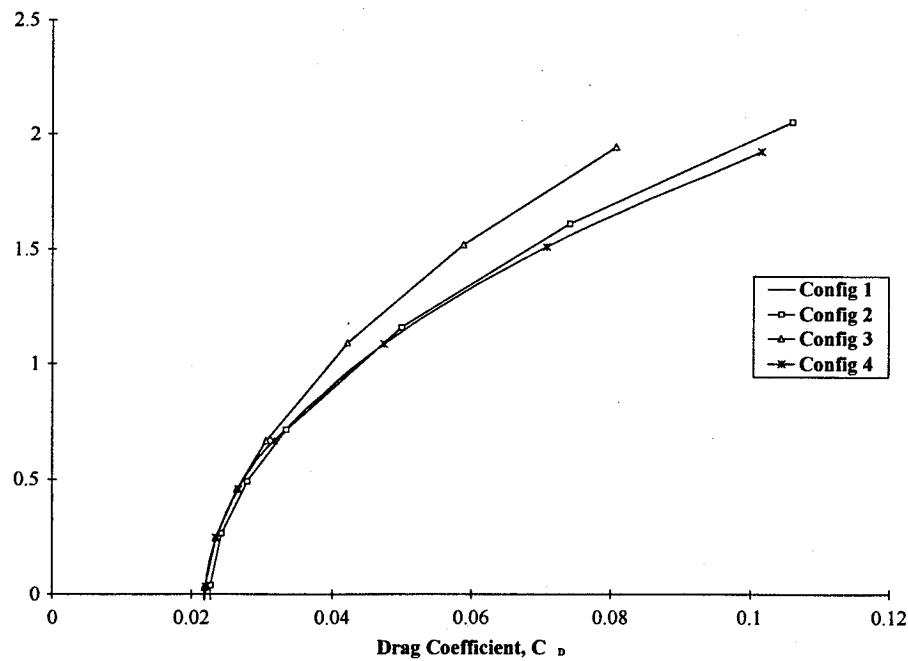
Figure 7.3-5. Optimal GRA Configurations (1 to 4) Low Speed Drag Polars
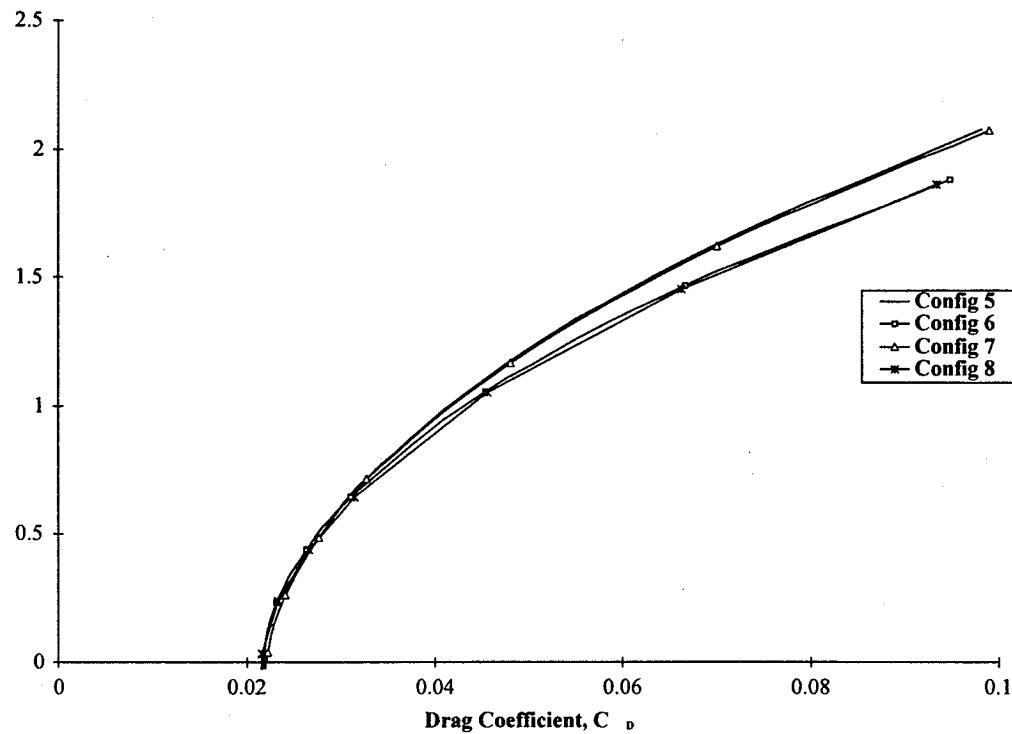
Figure 7.3-6. Optimal GRA Configurations (5 to 8) Low Speed Drag Polars

Finally, a system level performance characteristic of interest is the payload versus range capability. Figure 7.3-7 shows the eight configurations and the maximum obtainable range for a ferry mission. Configuration 8 had the longest range with a capability of 17,682 nm. Configuration 6 was the next farthest with 17,143 nm. These aircraft achieved these distances due to low TOWG and advanced technology engines, i.e. post-IHPTET. These engines provided a significant reduction in SFC and thus allow farther distances for a constant fuel amount. The values of are tabulated in Table 7.3-10.
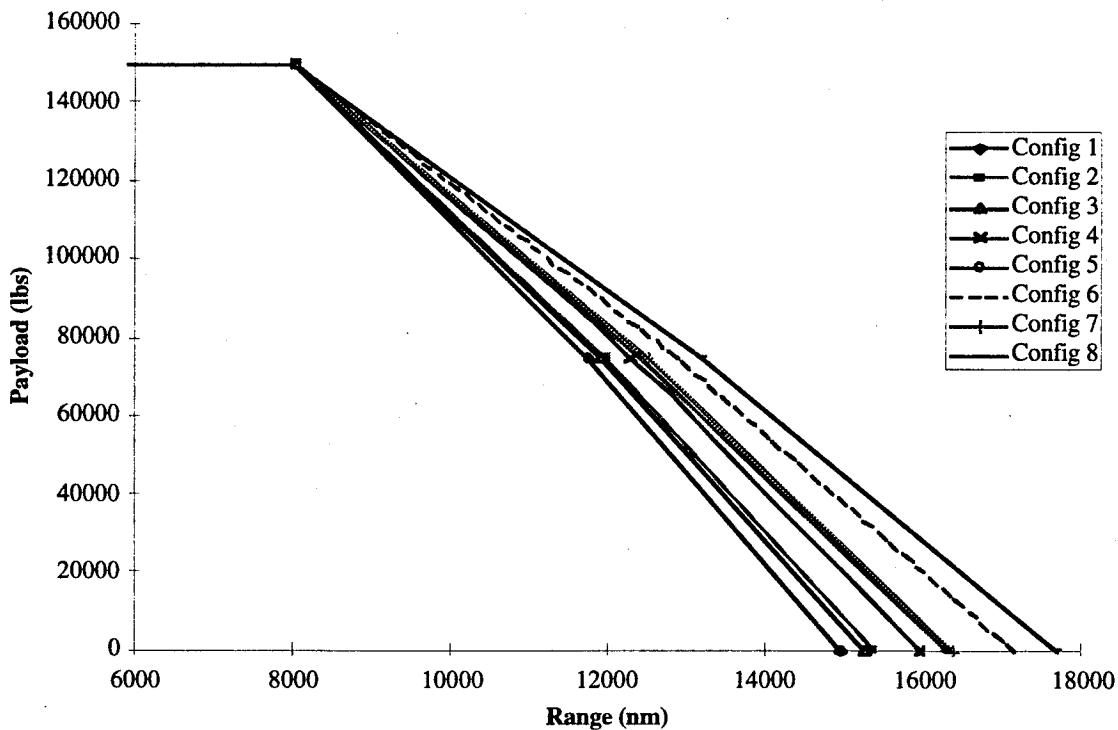
Figure 7.3-7. Optimal GRA Configurations Payload vs. Range

Table 7.3-10. Configuration Range Capability for a Ferry Mission

Configuration	Ferry Range (nm)
1	14944
2	15353
3	15238
4	15945
5	16301
6	17143
7	16370
8	17682

As a result of this study, it would appear as if configurations 6 and 8 were the superior aircraft with respect to the eight optimal configurations identified. Both aircraft had the lowest TOGW at 629,358 lbs. and 596,609 lbs. respectively. These low values were a consequence of the reduced mission fuel requirements from the infusion of the post-IHPTET engines and HLFC technologies. In addition, both aircraft complied with all four constraints and achieved a maximum cruise Mach number of 0.84 and 0.83. Even though the desirability function values were not at the maximum levels, the overall performance characteristics and system-level metric values are evidence to the excellence of these two aircraft Each aircraft could fly a substantial length for a ferry mission. And as a result of the lower mission fuel requirements, the needed DOC will be reduced.

As a comparison of the optimal configurations to the original GRA baseline, Figures 7.3-8 and 7.3-9 show three views the original and the optimal configuration 8. There is a slight discrepancy in the scale of the two views so that both attributes may be seen. All eight configurations made some deviations to the original baseline. For example, the AR increased from 10.7 to 11 and the SW decreased from 6800 ft^2 to 6165 ft^2 for configuration 8. Also, the wing quarter chord sweep increased from 18 deg to 30.01 deg. The thickness-to-chord ratio did remain constant.
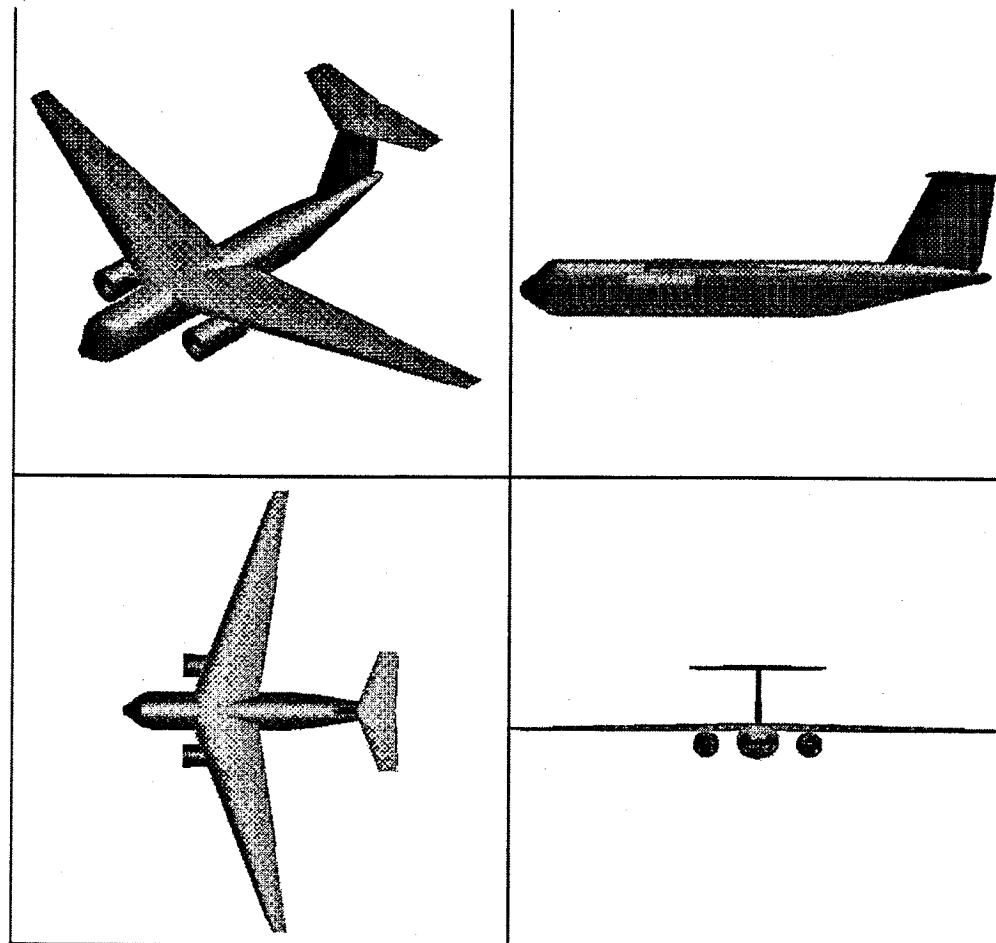
Figure 7.3-8. Original GRA Baseline
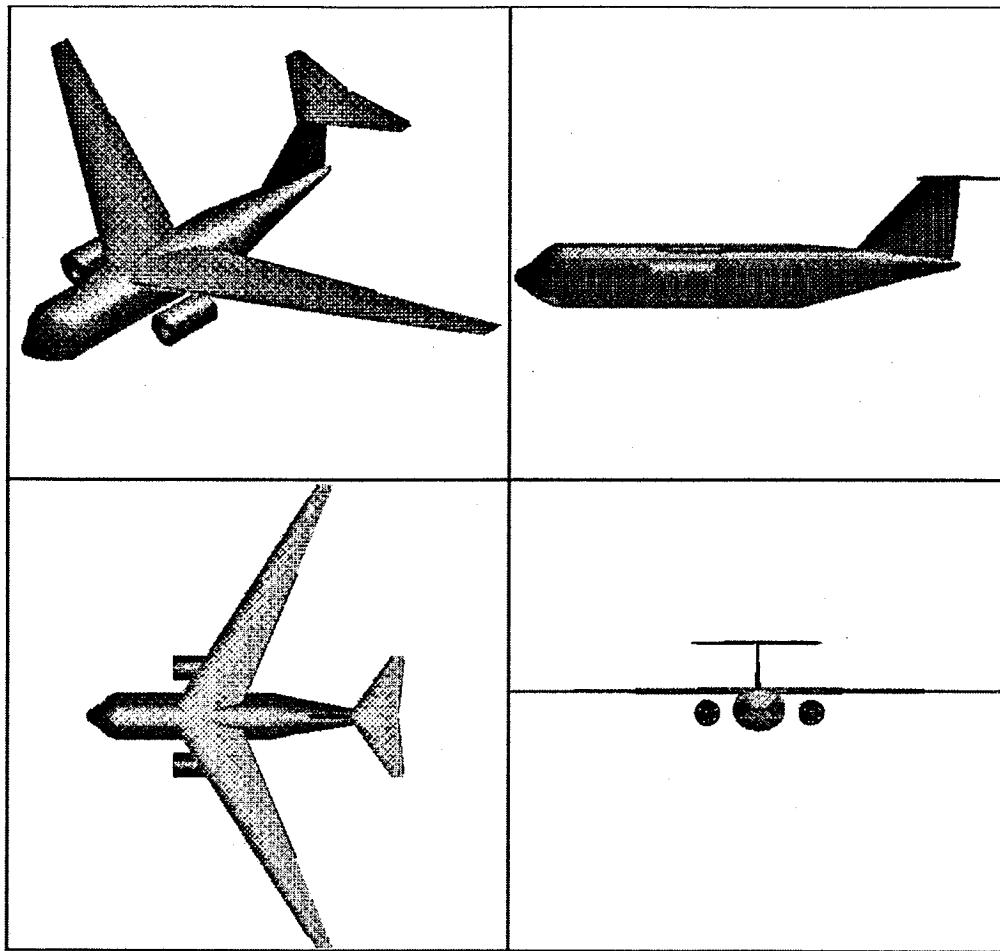
Figure 7.3-9. Optimal GRA Configuration 8

7.4. Conclusions

The conclusions reached from this study were that a design space exists where eight different feasible configurations meet the GRA requirements and may be considered for the proposed DMLA. These configurations included a clean baseline aircraft (no technologies), with HLFC, with composites, with composites and HLFC, with post-IHPTET engines, with post-IHPTET engines and HLFC, with post-IHPTET engines and composites, and with post-IHPTET engines, HLFC, and composites. The design space was achieved by deviating the geometric characteristics and evaluating performance characteristics and six system level metrics subject to four constraints. The eight main contributing variables included wing AR, wing SW, wing quarter chord sweep, wing TCA, SHT, SVT, TWR, and cruise Mach number. Each design variable was subject to optimization and an optimal aircraft for each configuration was established. The optimal settings of the design variables produced a great deal of commonality. Actually, five optimal design variable setting were the same for all aircraft: AR of 11.0, TCA of 0.11%, SHT of 1225.0 ft^2, SVT of 900.0 ft^2, and TWR of 0.26. The remaining variables, SW, SWEEP, and VCMN, varied between 6290-6800 ft^2, 24.88-30.01 degrees, and 0.78-0.84.

Comparing the optimal geometric characteristics to the original baseline, Table 7.4-1, one can see that only one of the original geometric characteristics that was under investigation remained the same through the optimization procedure. The wing thickness-to-chord ratio stayed constant at 11%.

Table 7.4-1. Deviation of Optimal Geometric Characteristics from the GRA Baseline

Component	Parameter	Baseline GRA Value	Eight Optimal Configurations
Wing			
	Aspect Ratio	10.7	11.0
	Taper Ratio	0.21	0.245
	Area	6815 ft²	6290 - 6800 ft²
	Span	270 ft	263 - 273.5 ft
	Quarter Chord Sweep	18°	24.88° - 30.01°
	Thickness-to-chord Ratio	11%	11%
Horizontal Tail			
	Aspect Ratio	4.07	3.9
	Taper Ratio	0.45	0.25
	Area	1300.77 ft²	1225.0 ft²
	Span	72.77 ft	69.12 ft
	Quarter Chord Sweep	18°	29°
	Thickness-to-chord Ratio	10%	8%
Vertical Tail			
	Aspect Ratio	1.27	1.15
	Taper Ratio	0.63	0.33
	Area	1037.81 ft²	900.0 ft²
	Span	36.24 ft	32.2 ft
	Quarter Chord Sweep	24.4°	37°
	Thickness-to-chord Ratio	12%	9%

The optimal aircraft met or exceeded the original baseline performance characteristics with respect to all evaluation criteria. The original baseline GRA TOGW was 854,872 lbs. The range of TOGW of the eight optimal configurations was 704,683 lbs for the clean aircraft to 596,609 lbs for the configuration with post-IHPTET engines, HLFC, and composites. Hence, the geometric optimization provide to be a valuable assessment.

Furthermore, a primary aspiration of this study was to assess the impact of an increase in cruise Mach number on the performance and hence size of the aircraft. If the Mach number could be increased from 0.78 to 0.82-0.85, the aircraft could possibly be compatible with commercial ventures. This goal was achieved by four configurations: 2 (0.84), 4 (0.83), 6 (0.84), and 8 (0.83). Each one of these configurations employed HLFC technology. Configurations 2

and 4 utilized IHPTET engines, whereas 6 and 8 used post-IHPTET engines. Configurations 2 and 4 appear to be the superior designs from the eight optimal configurations identified. Both out-perform the other six aircraft with respect to an overall system view point.

8. Directions for Improvement

The results obtained in this research problem demonstrate the feasibility of the DMLA concept, and the economic analysis yielded especially promising results. However, the specialized aircraft generally provided superior performance, allowing much room for improvement. The economics may stand some improvement as well. In either case, the DMLA may be improved by redesign through more radical geometric configurations, as well through the infusion of new technologies.

8.1. Assessment of Advanced Configurations

Aside from the conventional wing-body-empennage configurations studied here, the possibility exists to design a DMLA with alternative geometric configurations. The New World Vistas encourages development of the following concepts. While intended for design of a GRA, they are largely applicable to a VLT aircraft, and therefore a DMLA.

8.1.1. Stealth Transport
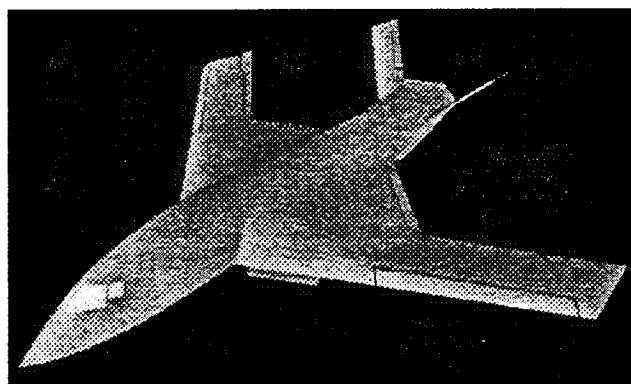
Figure 8.1-1. Concept for Stealthy Military Transport.

The survivability of a GRA could be greatly improved by applying current stealth fighter technology to a transport aircraft. This includes stealthy geometric shaping, composite structures usage, and radar-absorbent materials application. Operations in higher-risk environments could then be conducted more confidently (Mobility 22-23). It remains to be seen, however, if these concepts would hinder VLT performance if applied to a DMLA.

8.1.2. Twin Fuselages
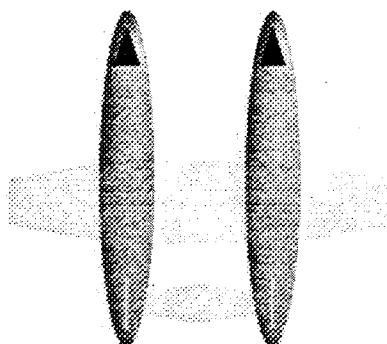
Figure 8.1-2. Twin-Fuselage Aircraft.

Lift-induced drag decreases with increased wing aspect ratio; however, the enormous wing spans that result from a high aspect ratio wing presents a difficult structural challenge. One solution is to use twin side-by-side fuselages, separated by a straight wing. The wing aspect ratio could increase from current values between 7 and 9 to as high as 12, without the accompanying structural weight penalties (Mobility 23-24).

8.1.3. Strut-Braced Wing
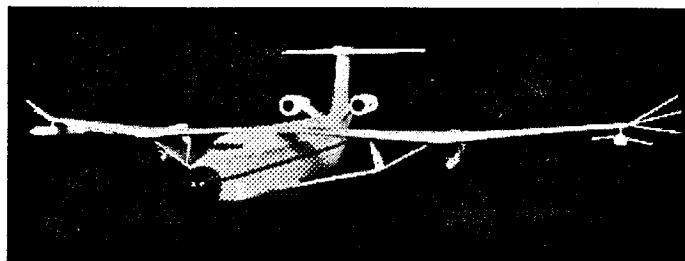
Figure 8.1-3. Strut-Braced Wing Aircraft.

Another concept for achieving higher lift-to-drag through increases in wing aspect ratio is the implementation of strut-braced wings. Wing struts are used to brace the large-span wings; the weight of the added span is thus partially supported by the struts. This not only reduces the weight of the wing-body fairing, it also limits wing bending due to weight (Aircraft 6).

8.1.4. Blended Wing-Body
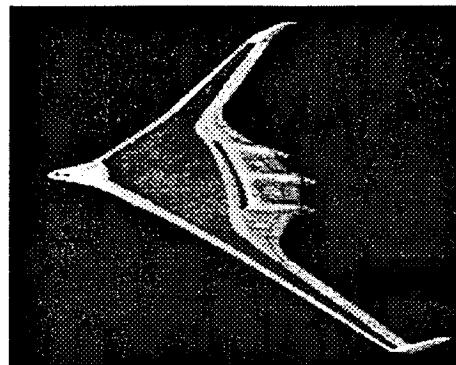
Figure 8.1-4. Blended Wing-Body Aircraft.

Instead of increases in aspect ratio, higher lift-to-drag may be realized by eliminating one sizable drag component. The blended wing body blends the fuselage into the wings, reducing fuselage drag and improving overall lift-to-drag ratio. Furthermore, the blended design eliminates the need for concentrated wing-body fairings (Aircraft 6).

8.2. Assessment of New Technologies

A conventional baseline aircraft, such as those developed in this study, may stand to benefit from the infusion of new, enabling technologies. Review of previous GRA and VLT work, the recommendations of the New World Vistas, and research of other publicly available literature revealed the following technological candidates and their associated benefits.

8.2.1. Aerodynamics

The following aerodynamic technologies could improve the lift-to-drag ratio and increase the cruise Mach number of a DMLA.

- alteration of aircraft geometry to the advanced configurations depicted above
- flow control, via
  - wing riblets (Aircraft 11)
  - micro-vortex generators (Aircraft 11)
  - NLFC (Global 3)
  - HLFC (Kirby 11)
- circulation control (based on ASDL studies)
- supercritical airfoils (Kirby 11)

Boeing predicts that laminar flow control methods are "...the aerodynamic concept offering the greatest potential for conserving fuel... Later results indicated that LFC can provide large reduction in fuel usage, and lower gross weights." However, Boeing also cautions that "...the life-cycle costs were found to be very dependent on airplane utilization, on technology complexity costs, and on LFC total systems weight." (Application 3)

8.2.2. Propulsion

The following propulsive technologies could decrease engine thrust-to-weight and specific fuel consumption of a DMLA powerplant, as well as increase affordability.

- low cost-of-ownership engines (currently under study by General Electric)
- post-IHPTET engine technologies (the LMASC GRA uses IHPTET powerplants, as specified on page four of their design report)
- lower sfc via
  - high-temperature engine materials and structures (Aircraft 11)
  - high bypass ratio (Aircraft 11)

8.2.3. Structures

The DMLA structural weight could be decreased by implementing the following technologies.

- advanced airframe materials, such as
  - composites (Aircraft 11)
  - advanced metallics (Aircraft 11)
  - high-temperature, light weight materials (Aircraft 11)
- adaptive structures and smart materials
  - active load/thermal control (Aircraft 11)
  - active flexible wing (Technologies 1)

8.2.4. Controls

The handling qualities and flight performance of a DMLA may benefit from these innovations in flight control technologies.

- integrated control system architecture (encompassing avionics, engines, subsystems) (Aircraft 11)
- thrust vectoring (Technologies 3)
- fiber-optic control signals (Aircraft 11)
- electric actuators (Aircraft 11)

Taken together, these last two technologies are termed "fly by light, power by wire" (Aircraft 11); Lockheed recommended their use to advance GRA development (Technologies 3).

8.2.5. Miscellaneous

Additional advancements include the following:

- innovative methods for design and manufacture, to lower cost (Mobility 13) (This is the heart of the ASDL design methodology, and is briefly discussed in the next chapter.)
- mission adaptive wing (Technologies 3)
- advanced avionics (Technologies 3)

9. Remaining Issues and Future Work

It cannot be stressed enough that this study is only the first step in the entire two-year research effort. Many assumptions were made in this study to simplify the analysis and quickly arrive at the conclusion as to the DMLA concept's feasibility. Often, these assumptions prevented certain issues from becoming addressed. In a few cases, the assumptions made were erroneous, and must be corrected in the future. Finally, the overall method was simplified to such a degree as to yield suboptimum results.

9.1. RSM Implementation

Response Surface Methodology (RSM) is based on a statistical approach to building and rapidly assessing empirical models. By careful design and analysis of experiments or simulations, the methodology seeks to relate and identify the relative contributions of the various input variables to the system response. In most cases, the behavior of a measured or computed response is governed by certain laws which can be approximated by a deterministic relationship between the response and a set of design variables. Usually the exact relationship between this response and the design variables is either too complex or unknown, and an empirical approach is necessary to determine it. This relationship is known as a Response Surface Equation, or RSE. Once this equation is constructed, it can be used in lieu of more sophisticated, time consuming codes to predict and optimize the response of a sub-system or the entire system. The "optimal" settings for the design variables are identified by finding the maximum or minimum of this equation, and a confirmation case is run using the actual simulation code to verify the results.

Future work will implement RSM as a powerful tool to quickly and accurately provide results, allowing for the circumvention of the difficulties encountered in this study and the greater ease of conducting additional studies. Using RSM, future studies will no longer need to use the FLOPS internal optimizer, thereby avoiding all of its pitfalls. FLOPS would be run multiple times in an analysis-only mode, with each run containing a different combination of variable settings as specified by a given Design of Experiments (DoE). These runs would provide results that eventually would lead to the creation of RSEs. RSEs would become models of the performance and economics responses reviewed here. Not only could the RSEs be optimized to determine the truly "optimum" performance possible in the aircraft of this study, they also would eliminate the need for manual iteration steps, such as those needed to develop the GRA variants of the DMLA. For example, an equation for wing fuel volume as a function of wing geometry variables could be generated. In parallel with another equation for performance (e.g., Mach number) including the geometry terms of the wing fuel equation, the optimum-performing configuration containing enough wing fuel volume could be quickly determined off-line.

RSM allows for additional studies to be performed as well. These studies are outlined in the next two sections.

9.2. Additional Design Issues

This study included no treatment of numerous design specifics, nor were trade studies conducted as yet in this research effort. The system-level analyses performed to complete this research problem contained conceptual-level work only; for progression to further preliminary-level studies, the following issues must be addressed and resolved.

First, the DMLA was concluded to require a high wing, without little assessment of the impact of this decision. The Federal Aviation Administration (FAA) has yet to certify a high-wing transport aircraft; they may develop neither the means nor the desire to certify the VLT variant of a DMLA. A high wing allows for no over-wing exits, nor could the wing hold the fuselage above the water level in the event of a forced water landing. Furthermore, placing the wing above the fuselage eliminates the noise-shielding effects of the wing - the engines are thus placed in a direct line with the fuselage. Depending on the size and placement of the wing-body fairing, the passenger cabin may also be affected. Additionally, for compliance with military rules, the DMLA wing structural load factor is 2.25, far less than the 3.75 value typically employed on commercial transports. Not only does this again raise the specter of FAA certification, it also brings to light the limitations now placed on the maneuverability of the VLT variant.

Second, if a twin-engine DMLA becomes possible, as this study indicates, the engines required would have extremely high thrust output. Immediately, consideration of aircraft performance following the loss of an engine presents itself. Could a DMLA continue flying on one functioning powerplant? If so, what magnitude of off-axis thrust yawing would result? How does this affect engine placement and flight control designs? These are all questions that will need to be answered in future studies. Superseding these considerations must be a quantification of the reliability of a twin-engine versus four-engine DMLA.

Third, as stated before, several subtleties of producing GRA and VLT variants of a DMLA were not considered. The acquisition cost results are surely optimistic, in light of this fact. The extra cost involved with GRA variant development from a DMLA design for a VLT must be accounted for. The effects of production line fuselage plug installment on manufacturing costs must be modeled as well.

Fourth, cost modeling must be expanded to include a full economic uncertainty analysis, as part of the ASDL Robust Design Methodology. RSM comes to bear in this case, especially in light of the uncontrollable and unpredictable nature of the economic variables that come into play, which were assumed fixed in this study. RSEs must be generated that include these "noise" factors, and they must be treated probabilistically in order to fully ascertain economic viability and benefit of the DMLA. In addition, for the GRA, this work must be completed using TCM instead of ALCCA.

An additional note to the cost modeling problem involves what would be real-world financing of the DMLA program. If industry finances the DMLA program using private funding,

developing it first as a VLT and then creating a GRA variant, the military's acquisition cost would drop significantly, as this customer normally funds the entire research and development program itself. On the other hand, if developed first as a GRA, then as a VLT, the commercial customer stands to benefit from lower acquisition costs, benefiting from an aircraft whose development costs were absorbed by the military. Ways to model either scenario must be defined, and the results quantified.

Fifth, trade studies must now be performed to define the performance envelope of a DMLA. Completion of this research problem largely entailed a purely preliminary- and system-level analysis, which allowed for no deviations of aircraft capability. To assess the DMLA concept's technical feasibility, it first remained to be seen if the originally specified missions could be completed. With this now demonstrated for a DMLA, this study has provided a starting point for future work, which must now include an assessment of wing loading versus thrust-to-weight, payload capacity versus range, and payload capacity versus fuel required. Furthermore, the effects of takeoff and landing field lengths on aircraft sizing must be quantified, especially in the face of the constraints on these quantities. Additionally, the effects of flights resulting in off-design ranges on landing field length must be assessed. An additional cost-related trade study presents itself with the expected availability of General Electric's work into low cost-of-ownership engines. Aircraft performance and cost with these engines versus more "conventional" powerplants can be compared. Here, RSM becomes an extremely useful tool. If the aircraft are modeled mathematically, using RSEs containing terms of performance and cost, then trade studies become a simple matter of setting the RSE terms at different values and recording the results. Manually executing multiple instances of FLOPS or other design tools to generate the same results becomes unnecessary.

Finally, numerous design specifics must now be considered. In the course of this study, a number of military transport design details were overlooked. Ground cargo handling alone provides a number of design constraints. A GRA must be compatible with present Army and Marine Corps material handling systems, including fork lifts and transport vehicles (C-141 1). Also, especially large aircraft (such as the C-5, and the DMLA) require landing gear with a pneumatic "kneeling" capability to lower the main cargo deck within reach of ground handling equipment (C-5 1). Additionally, accommodations must be provided for both the flight crew and embarked personnel, including galleys, lavatories, and rest bunks (C-5 2). Future design work must account for each of these details.

Some of these specifics give rise to similar considerations of a VLT variant. Cargo and baggage handling within the cavernous cargo hold of a DMLA will certainly be difficult; it is unlikely new ground equipment to service such an aircraft will be developed anytime soon. Thus, a VLT variant would require not only several more cargo doors than found on current transports, but some form of internal handling capability may also be required. Airport compatibility is another problem. The large number of passengers embarked on a VLT aircraft would certainly necessitate an airport capable of loading and unloading both cabin decks simultaneously. Even if some airports expanded to include this capability, the VLT aircraft may operate to airports that cannot do this. Thus, the VLT may again require some internal passenger loading capability. Again, future work must include consideration of these points.

The capabilities needed to fulfill the expected requirements of a GRA must be modeled as well. The <u>New World Vistas</u> define four different transport missions: personnel airlift, rapid cargo/troop deployment, aeromedical evacuation, and humanitarian aid-related duties. For a single aircraft to fulfill these different roles, its payload bay must be modular and easily reconfigurable, as in the C-141 (C-141 1). As mentioned previously, many of these missions may require deployment into hostile areas with diminishing fighter escorts, to airfields with limited ground support equipment. Thus, aircraft must be able to withstand battle damage and continue operations; the wings of the C-17, for example, can withstand small arms fire - at the cost of a wing weight equal to one-third the total aircraft empty weight (Air 122). A GRA must also contain FEBA equipment for self reliability, such as the self-inflating undercarriage and internal cargo unloading systems found on the C-5 (Wilkinson 38). Such airfields may be short and unimproved, requiring STOL-capable high-lift devices and hardened landing gear (Air 120). Survivability is as much a matter of electronic as physical warfare; the *Vistas* thus specify that future aircraft must contain improved ECM systems, and the means to resist enemy jamming.

9.3. Aerodynamic Investigation Recommendations

Even though all of the eight configurations were identified as superior to the original baseline, only one achieved this goal without the addition of advanced technologies; that aircraft being the clean configuration. The evaluation of these aircraft was considered from the benefit point of view of adding new technologies. Yet, there is also risk associated with each one and it must be quantified based on readiness and confidence. A designer must consider if those technologies will be ready for widespread application by the time of the aircraft's introduction to service. In addition, the designer must also be confident that those technologies are proven and mature. For example, if a failure were to occur to the HLFC during cruise, what would be the degradation in performance. Quantifying the risk associated with these technologies will be one focus of future research.

In addition, the aerodynamic analysis was executed with analysis tools based on empirical relationships. As stated previously, this over-simplified the analysis from one of accuracy to one of approximation. There were two main reasons for this method. First, the analysis tools AERO2S and VORLAX are incapable of analyzing compressibility effects. This inhibited the use of these codes due to the fact that the aircraft was to be analyzed in a region of compressibility. Second, the grid generator required by the full potential flow solver, FPS3D, had not arrived from NASA. At the conclusion of this study, the software had arrived and was being put in place. This method of execution should be applied and compared to the results of this study. This would also allow validation of the internal aerodynamic capabilities of FLOPS.

Finally, one of the main objectives of the study was to establish a feasible design space of GRA configurations. This design space would hopefully capture the needs of the VLT mission requirements. To respond to this need, the future work should identify which aircraft is more critical with respect to volumetric requirements; that is, does the VLT or the GRA require more internal volume to complete its given mission. Once that critical volume is established, then both aircraft baselines need to be readjusted to meet those volumetric requirements. For example, if the VLT turns out to be more critical, then the GRA fuselage dimensions must be adjusted. Furthermore, to attempt to maintain commonality on the manufacturer's production line, could the GRA be converted from the VLT by inserting plus forward and aft of the wing? Also, the more geometric commonalities that exist between the two aircraft further enhance the feasibility of the DMLA. These items will be the thrust of future efforts on the GRA and DMLA.

9.4. Further Studies

Following is a road map for the remainder of the DMLA research effort, due for completion by August 15, 1998. Except for obvious circumstance, this work will be accomplished for four-engine, conventional aircraft.

January 15 through August 15, 1997 (end of first year)

1. Definition of the overall evaluation criterion (OEC) for GRA and VLT aircraft. Also define OEC for a DMLA, capturing its specific metrics.

2. Resolution of above issues; modeling of solutions and capabilities.

3. Implementation of ASDL Robust Design Methodology to development of specialized GRA and VLT aircraft. This work includes the results of a study performed by Michelle Kirby in parallel with this research problem. Ms. Kirby implemented RSM to expand a GRA point design into a design space capturing the geometry of the VLT. Also, her study included a detailed aerodynamic analysis of the DMLA problem, resulting in the identification of the most critical aerodynamic design variables to overall aircraft performance.

   The resulting GRA and VLT are optimized in terms of the OEC developed in (1), and incorporate the results of (2). This work includes the economic uncertainty analysis, outlined above.

   The work performed in this step includes the trade studies outlined in the previous section. Based on the conclusion of this study, all work performed in this time period considers four-engine aircraft only.

4. Utilization of results in (2) and experience from this study to develop a VLT variant of a DMLA. Apply RSM as described in section 8.2 above to derive a GRA variant.

5. Development and OEC optimization of DMLA aircraft in (4) using Robust Design Methodology, following procedure in (2).

6. Comparison of results of (2) and (5) to demonstrate capability of DMLA. Use results to identify areas of improvement via new technologies or configuration advancements.

August 15 through December 15, 1997

1. Risk vs. benefit assessment of technology candidates identified in this study and at end of first year design work.

2. Forecasting of GRA and VLT needs to the year 2025. Examine maturity of technological candidates in (1) over the 1997-2025 time frame.

3. Detailed structural analysis of GRA, VLT, and DMLA aircraft following the work performed for DMLA aerodynamics by Ms. Kirby.

January 15 through August 15, 1998 (conclusion of research effort)

1. Application of technologies identified (in work completed to this point) to the DMLA developed by this time. Resulting aircraft to be modeled as outlined above. Results should be compared to quantify any benefit in terms of OEC.

2. If possible, configuration alternatives should also be investigated. This includes modeling and preliminary analysis, to compare against all other DMLA aircraft.

Acknowledgments

REFERENCES

1. Gething, M.J., "The Lifeblood of Global Mobility", Defence, March 1993.

2. Wilkinson, S. "Big", Air & Space, August 1989.

3. New World Vistas Mobility Volume. United States Air Force Scientific Advisory Board, Wright-Patterson Air Force Base; Dayton, OH, December 15, 1995.

4. "C-5A/B Galaxy". USAF Fact Sheet 92-35; http://www.dtic.mil/airforcelink/pa/factsheets/ C_5A_B_Galaxy.html, October 1992.

5. "C-141B Starlifter". USAF Fact Sheet 92-19; http://www.dtic.mil/airforcelink/pa/factsheets/ C_141B_Starlifter.html, October 1992.

6. "C-17 Globemaster III". USAF Fact Sheet 95-06; http://www.dtic.mil/airforcelink/pa/ factsheets/ C_17_GLobemaster_III.html, July 1995.

7. High-tech lifter from Long Beach", Air International, March 1993.

8. "Global Range Aircraft", Lockheed-Martin Aeronautical Systems, May 5,1996.

9. "Global Range Aircraft Mission Description", Lockheed-Martin Aeronautical Systems, September 5,1996.

10. "Preliminary Assessment of the Economic Viability of a Family of Vary Large Transport Configurations", Presented at the 1st World Aviation Conference, Los Angeles, CA, October, 22-24, 1996, AIAA 95-5516.

11. "Specifications", Aviation Week & Space Technology, March 18, 1991.

12. Mavris, D.N., Brewer, J.T., Bandte, O., "A Method for the Identification and Assessment of Critical Technologies Needed for an Economically Viable HSCT", 1st AIAA Aircraft Engineering, Technology, and Operations Congress, Los Angeles, CA, September, 1995, AIAA-95-3887.

13. Montgomery, D.C., Design and Analysis of Experiments, 3rd Edition, John Wiley & Sons, Inc., New York, NY, 1976.

14. Box, G.E.P., Hunter, W.G., Hunter, J.S., Statistics for Experimenters, An Introduction to Design, Data Analysis, and Model Building, John Wiley & Sons, Inc., New York, NY, 1978.

15. SAS Institute Inc., JMP, Computer Program and User's Manual, Cary, NC, 1994.

16. Deiter, G.E., Engineering Design, A Materials and Processing Approach, 2nd Edition, McGraw-Hill, Inc., New York, NY, 1991.

17. Derringer, D., and Suich, R., "Simultaneous Optimization of Several Response Variables," Journal of Quality Technology, 12:4, 214-219, 1980.

18. "Notional Aircraft Design Definition Study", Final Report under contract number F33615-94-C-3010, Aerospace Systems Design Laboratory, October 1996.

19. "RAM, Rapid Aircraft Modeler";http://fornax.arc.nasa.gov:9999/ram.html.

20. McCullers, L.A., Flight Optimization System, Computer Program and Users Guide, Version 5.85, NASA Langley Research Center, Hampton, VA, December 1994.

21. Carlson, H.W. and Mann, M.J. "Survey and Analysis of Research on Supersonic Drag-Due-to-Lift Minimization with Recommendations forWing Design," NASA-TP-3202, Sept. 1992.

22. Carlson, H.W., Darden, C.M. and Mann, M.J. "Validation of a Complete Code for Analysis of Subsonic Aerodynamic Performance of Wings with Flaps in Combination with a Canard or Horizontal Tail and an Application to Optimization," NASA-TP-2961, Jan. 1990.

23. Carlson, H.W. and Darden, C.M. "Validation of a Pair of Computer Codes for Estimation and Optimization of Subsonic Aerodynamic Performance of Simple Hinged-Flap Systems for Thin Swept Wings," NASA-TP-2828, Nov. 1988.

24. Middleton, W.D. and Landry, J.L. "A System for Aerodynamic Design and Analysis of Supersonic Aircraft, Part I: General Description and Theoretical Development," NASA-CR-3351, 1980.

25. Middleton, W.D., Landry, J.L., and Coleman, R.L. "A System for Aerodynamic Design and Analysis of Supersonic Aircraft, Part II: User's Manual," NASA-CR-3352, 1980.

26. Middleton, W.D., Landry, J.L., and Coleman, R.L. "A System for Aerodynamic Design and Analysis of Supersonic Aircraft, Part III Computer Program Description," NASA-CR-2717, July 1976.

27. Kinney, D.J., "FPS3D, User's Guide Version 3.0," NASA Ames, Sept. 1996.

28. Miranda, L.R., Elliot, W.M., and Baker, W.M., "A Generalized Vortex Lattice Method for Subsonic and Supersonic Flow Applications," NASA CR 2865, December 1977.

29. Pieró, J., Peraire, J., and Morgan, K., "FELISA System, Version 1.1, Reference Manual Part 2 - User Manual," January 27, 1995.

30. Birckelbaw, L.G., McNeill, W.E., and Wardwell, D.A., "Aerodynamics Model for a Generic ASTOVL Lift-Fan Aircraft," NASA TM-110347, April 1995.

31. Galloway, T.L., Mavris, D.N., "Aircraft Life Cycle Cost Analysis (ALCCA) Program," NASA Ames RC, Moffet Field, CA, September 1993.

32. Box, G.E.P., Draper, N.R., Empirical Model-Building and Response Surfaces, John Wiley & Sons, Inc., New York, NY, 1978.

33. Haaland, P.D., Experimental Design in Biotechnology, Marcel Dekker, Inc., New York, NY, 1989.

34. OusterHout, J.K., Tcl and the TK Toolkit, Addison-Wesley, Reading, MA, 1994.

35. Vanderplaats, G.N., Numerical Optimization Techniques for Engineering Design: With Applications, McGraw-Hill, New York, NY, 1984.

36. Fabrycky, W.J., Blanchard, B.S., Life-Cycle Cost and Economic Analysis, Prentice Hall, Englewood Cliffs, NJ, 1991.

37. "Advanced Technologies Applicable to Global Range Aircraft". Lockheed-Martin Aeronautical Systems, May 5, 1996.

38. Hayden, William T., Garcia, Elena and Arndt, Till. "Economic Uncertainty Analysis of the Boeing 747-400." Georgia Institute of Technology; Atlanta, GA, December 19, 1996

39. Kulfan, Robert M. and Howard, Weston M. "Applications of Advanced Aerodynamic Concepts to Large Subsonic Transport Airplanes". Boeing CAG, Seattle, WA; Nov 17, 1975.

40. Kulfan, Robert M. and Vachal, John D. "Wing Planform Geometry Effects on Large Subsonic Military Planes". Boeing CAG, Seattle, WA; Feb. 1978.

41. Lenorvitz, J.M. "Airbus Survey Confirms Requirement for Very Large Transport Aircraft." Aviation Week & Space Technology, October 28, 1991.

42. Marx, William and Mavris, Dimitri. "ALCCA User Notes." Aerospace Systems Design Laboratory; Atlanta, GA.

43. Mecham, M. and McKenna, J.T. "Cost, not Size, to Drive Success of Superjumbo." Aviation Week & Space Technology, November 12, 1994.

44. McCullers, L.A. "FLOPS User's Guide." NASA Langley Research Center; Hampton, VA, December 14, 1994.

45. New World Vistas Aircraft and Propulsion Volume. United States Air Force Scientific Advisory Board, Wright-Patterson Air Force Base; Dayton, OH, December 15, 1995.

46. New World Vistas Summary Volume. United States Air Force Scientific Advisory Board, Wright-Patterson Air Force Base; Dayton, OH, December 15, 1995.

47. Osburg, Jan. "Tailored Cost Model for Military Aircraft." Aerospace Systems Design Laboratory; Atlanta, GA, September 24, 1996.

48. "Product News." Pratt & Whitney; East Hartford, CT, April 1994.

APPENDIX
A

Top Level Shell Scripts

STARS

STARS is the top-level shell script which generates the GUI, copies the case file from the library of various design; and calls all other procedures which perform the actual creation of the case files, execution of the analysis program, and extraction of response surface data. This script is written in tk/tcl and executes on an IBM RS6000.

```
#!/usr/local/bin/wishx -f

#       To make stars compatible with gypsy use the following command
#               in the first line of this code
#
#               #!/usr/X11R6/bin/wishx -f

#       To make stars compatible with RS6000 use the following command
#               in the first line of this code
#
#               #!/usr/local/bin/wishx -f


#
#       Set the precision for expr calculations
#

global tcl_precision
        set tcl_precision 8
        set default_baseline_filename "baseline"


#       Prevents echo of info to screen

proc debug {args} {


#       To use debug commands, uncomment the following line
#       eval $args

}


#       Withdraw the "." window and set toplevel frames and windows

wm withdraw .
toplevel .stars
toplevel .results
toplevel .data

set Number_of_Sets 1
#
#       This is if I am using my account   - comment in or out
#
#       setting a hard coded path for the designs
```

```
set Design_Hard_Coded_Path /home/gypsy5/gt0038c/image/designs

#       setting a hard coded path for the procedures

set Procedure_Hard_Coded_Path /home/gypsy5/gt0038c/image/proc


#
#       This is if I am using asdl stars version  - comment in or out
#
#set Design_Hard_Coded_Path /home/gypsy5/asdl/defaults/AIX/stars/designs
#set Procedure_Hard_Coded_Path /home/gypsy5/asdl/defaults/AIX/stars/proc



#       I am telling stars to open the procedures in proc directory
#               to be used when called

foreach tclfile [glob $Procedure_Hard_Coded_Path/*.tcl] {

        source $tclfile

}



#set Design BBD
#set DesignType uniform

#
#       This is setting up the top message frame
#
frame  .stars.m

        label .stars.m.1 -text "Message:" -width 10 -anchor w -relief flat -bg white
        label .stars.m.2  -relief flat -bg white   -anchor w
        pack .stars.m.1 .stars.m.2 -side left -fill x -expand 1
pack .stars.m -side top -anchor w -fill x


#       setting up a field to put messages on the frame


proc Message {args} {
        eval .stars.m.2 conf -text \"$args\"
        update
}


frame .stars.m1 -bg burlywood

        label .stars.m1.alphalabel -text "RSE alpha" -width 10 -bg burlywood

        entry .stars.m1.alpha -relief sunken -width 10
                .stars.m1.alpha insert 0 1.

        label .stars.m1.baselabel -text "Baseline File" -width 15 -bg burlywood

        entry .stars.m1.basefile -relief sunken -width 15 \
                -textvar baseline_filename
                .stars.m1.basefile insert 0 $default_baseline_filename

        label .stars.m1.spacer -width 40 -bg burlywood

        pack .stars.m1.spacer .stars.m1.alphalabel .stars.m1.alpha .stars.m1.baselabel \
                .stars.m1.basefile -side left -fill x
```

```
pack .stars.ml -side top -fill x -anchor center
#

frame .stars.design -bg navy -border 5

        menubutton .stars.design.test -text "Select a Design:" \
                -bg grey -activebackground LightCoral \
                -relief raised -width 35 -menu .stars.design.test.menu -anchor w

            menu .stars.design.test.menu  -bg grey

#         setting up drop menu for screening DoE files

                        .stars.design.test.menu add cascade -label "Screening" \
                             -menu .stars.design.test.menu.screening


#         setting up drop menu for RSE DoE files

                        .stars.design.test.menu add cascade -label "RSE" \
                             -menu .stars.design.test.menu.rse

                    menu .stars.design.test.menu.screening  -bg grey
                    menu .stars.design.test.menu.rse  -bg grey

        button .stars.design.generate -relief raised -text \
                "Copy Cases for Design from input baseline file" -command GenerateCases \
                -bg grey -activebackground plum1 -width 40


        label .stars.design.spacer -width 29
        label .stars.design.spacer2 -width 29

        pack .stars.design.spacer .stars.design.test .stars.design.generate
.stars.design.spacer2 -side left




pack .stars.design -anchor w


#
#         This is setting up the drop menu for the Screening directory
#

        foreach file [lsort [glob -nocomplain $Design_Hard_Coded_Path/Screening/*]] {

                set number [file tail $file]
                .stars.design.test.menu.screening add cascade -label $number \
                        -menu .stars.design.test.menu.screening.$number
                    menu .stars.design.test.menu.screening.$number -bg plum1

                    foreach file2 [lsort [glob -nocomplain $file/*]] {

                            set type [file tail $file2 ]
                            .stars.design.test.menu.screening.$number add command -label
$type \
                                    -command "global Design DesignType Number_of_Sets
SorR;set Number_of_Sets $number; set Design [file rootname $type]; set DesignType [file
extension $type];.stars.design.test conf -text \"Design: $type, $number
variables\";FillSets;set SorR Screening"

                            debug puts stdout " $number [file rootname $type] [file extension
$type]"
```

```
                        }

            }

#
#        This is setting up the drop menu for the RSE directory
#

        foreach file [lsort [glob -nocomplain $Design_Hard_Coded_Path/RSE/*]] {

                set number [file tail $file]
                .stars.design.test.menu.rse add cascade -label $number \
                        -menu .stars.design.test.menu.rse.$number
                        menu .stars.design.test.menu.rse.$number -bg plum1

                foreach file2 [lsort [glob -nocomplain $file/*]] {

                        set type [file tail $file2 ]
                        .stars.design.test.menu.rse.$number add command -label $type
\
                                -command "global Design DesignType Number_of_Sets
SorR;set Number_of_Sets $number; set Design [file rootname $type]; set DesignType [file
extension $type];.stars.design.test conf -text \"Design: $type, $number
variables\";FillSets;set SorR RSE"

                        debug puts stdout " $number [file rootname $type] [file extension
$type]"

                }

        }




#      Create the frame for the window

frame .stars.header

        label .stars.header.number -text "Variable Number" -width 31
        label .stars.header.variable -text "Variable" -width 10
        label .stars.header.name -text Namelist -width 27
        label .stars.header.low -text Minimum -width 20
        label .stars.header.high -text Maximum -width 20
        label .stars.header.prefix -text "Prefixes" -width 25

        pack .stars.header.number .stars.header.variable .stars.header.name \
                .stars.header.low .stars.header.high \
                .stars.header.prefix -side left


pack .stars.header -side top -anchor w



#       Look for the DoE that user has chosen by looking for the DoE setup file in
directory designs


proc GenerateCases {} {

        global Number_of_Cases Design_Hard_Coded_Path SorR Design DesignType
Number_of_Sets Matrix baseline_filename
```

```
        if ![file exists $Design_Hard_Coded_Path/$SorR/$Number_of_Sets/$Design$DesignType]
{Message "This design/variable combination does not exist.";return}


        set file [open $Design_Hard_Coded_Path/$SorR/$Number_of_Sets/$Design$DesignType
"r"]


        set Number_of_Cases 0
        while ![eof $file] {


                set line [gets $file]
                if [eof $file] break
                incr Number_of_Cases 1

                set Matrix($Number_of_Cases) \{$line\}

        puts stdout $line

        }

        close $file

#       Generate copies of the baseline file to be manipulated later

                for { set i 1 } { $i <= $Number_of_Cases } {incr i 1 } {
                        Message "Copying case file case $i from baseline file. Please
Wait."
                        exec cp $baseline_filename case$i
                        update
                }

                        Message "Done creating $Number_of_Cases case files of the DoE."


                        .stars.bottom.create conf -state normal

}

proc CheckCases {} {
        global Number_of_Cases
                if { $Number_of_Cases > 400 } {
                        set Number_of_Cases 64
                }
}


proc FillSets {} {

        global Number_of_Sets


        foreach set [winfo children .stars] {
                if { $set == ".stars.header" || $set == ".stars.sets" || $set ==
".stars.cases" || $set == ".stars.m" || $set == ".stars.design" || $set == ".stars.m1" }
{continue}
                destroy $set
        }


        for {set i 1} { $i <= $Number_of_Sets} { incr i 1} {


                frame .stars.set$i
```

```
                    label .stars.set$i.label$i -text "$i" -width 20
                    entry .stars.set$i.variable$i -width 20 -relief sunken -textvar
Variable$i

                    entry .stars.set$i.namelist$i -width 20 -relief sunken -textvar
Namelist$i

                    entry .stars.set$i.minimum$i -width 20 -relief sunken -textvar
Minimum$i

                    entry .stars.set$i.maximum$i -width 20 -relief sunken -textvar
Maximum$i


                    label .stars.set$i.spacer$i -width 2
                    entry .stars.set$i.leader$i -width 20 -relief sunken -textvar
Leader$i


                    bind .stars.set$i.variable$i <Tab> "focus .stars.set$i.namelist$i"
                    bind .stars.set$i.namelist$i <Tab> "focus .stars.set$i.minimum$i"
                    bind .stars.set$i.minimum$i <Tab> "focus .stars.set$i.maximum$i"
                    bind .stars.set$i.maximum$i <Tab> "focus .stars.set$i.variable$i"



                    pack .stars.set$i.label$i .stars.set$i.variable$i
.stars.set$i.namelist$i \
                            .stars.set$i.minimum$i .stars.set$i.maximum$i \
                            .stars.set$i.spacer$i .stars.set$i.leader$i -side left



            pack .stars.set$i -side top
            update


        }


        frame .stars.bottom -bg burlywood

                button .stars.bottom.create -relief raised -text \
                        "Create Case Files From Above Variables" -command
".stars.bottom.create conf -state disabled;CreateFiles" \
                        -bg grey -activebackground LightCoral -width 50 -state disabled
                frame .stars.bottom.frame -bg lightgrey

                button .stars.bottom.frame.run -relief raised -text \
                        "Run ALCCA" -command ".stars.bottom.frame.run conf -state disabled;
RunALCCA" \
                        -bg grey -activebackground LightCoral -width 50 -state disabled

                    button .stars.bottom.frame.flops585 -relief raised -text \
                            "Run FLOPS V5.85" -command ".stars.bottom.frame.flops585 conf -
state disabled; Runflops585" \
                            -bg grey -activebackground LightCoral -width 50 -state disabled

                    button .stars.bottom.frame.flops57 -relief raised -text \
                            "Run FLOPS V5.7" -command ".stars.bottom.frame.flops75 conf -state
disabled; Runflops57" \
                            -bg grey -activebackground LightCoral -width 50 -state disabled

                button .stars.bottom.frame.kirby -relief raised -text \
                        "Run Michelle's Special" -command ".stars.bottom.frame.kirby conf -
state disabled; Kirby" \
                        -bg grey -activebackground LightCoral -width 50 -state disabled

                button .stars.bottom.frame.aero -relief raised -text \
                        "Run Aero Codes" -command ".stars.bottom.frame.aero conf -state
disabled ;Runaero" \
                        -bg grey -activebackground LightCoral -width 50 -state disabled
```

```
            button .stars.bottom.quit -relief raised -text \
                "QUIT AND DON'T SAVE" -command exit \
                -bg PaleVioletRed -activebackground plum1 -fg black -width 50 -
border 8
            label .stars.bottom.image -relief raised -text \
                "STARS Ver 3.0 Nov 16,96    Note: This easy to use interface
compliments of Michelle & Mark" \
                -bg plum4 -fg white


        pack .stars.bottom.image  -side bottom -fill x
        pack .stars.bottom.create -side top

        pack .stars.bottom.frame.run .stars.bottom.frame.flops585
.stars.bottom.frame.flops57 .stars.bottom.frame.kirby \
                .stars.bottom.frame.aero  -side top
        pack .stars.bottom.frame -side left
        pack .stars.bottom.quit -side right


    pack .stars.bottom -side bottom -fill x


}

FillSets


#       Main program ends here. The rest is procedures
```

CreateFiles.tcl

This script control the creation of the cases files generated by the DoE chosen from the top-level script STARS. This procedure looks into every case file and switches the appropriate variable with the values entered into the GUI with the corresponding value of the DoE table. A special option is inserted if the basleine file is called "kirby". This option switched the low speed drag polars in the case file being created.


```
proc CreateFiles {} {

        global Number_of_Sets Matrix Number_of_Cases alpha
        set flag [.stars.ml.basefile get]
        set alpha [.stars.ml.alpha get]
        Message "Verifying Fields"
        for {set i 1} { $i <= $Number_of_Sets} { incr i 1} {
                global Variable$i Namelist$i Minimum$i Maximum$i Leader$i
#               global Variable$i Namelist$i Minimum$i Midpoint$i Maximum$i Leader$i
                if { [set [set temp Variable$i]] == "" } {Message "Please fix empty
fields";return}
                if { [set [set temp Namelist$i]] == "" } {Message "Please fix empty
fields";return}
```

```
                        if { [set [set temp Minimum$i]] == "" } {Message "Please fix empty
fields";return}
#                       if { [set [set temp Midpoint$i]] == "" } {Message "Please fix empty
fields";return}
                        if { [set [set temp Maximum$i]] == "" } {Message "Please fix empty
fields";return}
               }

       for {set i 1} { $i <= $Number_of_Cases} { incr i 1} {
               Message "Creating Case File $i  --   Please Wait."

#   I am switching out the specific low speed drag polars from files low$i
               set k 0
               if { $flag == "kirby" } {
                       set lowfile [open low$i "r"]
                       while ![eof $lowfile] {
                               incr k 1
                               set lsline [gets $lowfile]
                               if [eof $lowfile] break
                               if { $k == 1 } {
                                       lassign $lsline cl1 cl2 cl3 cl4 cl5 cl6 cl7 cl8 cl9
cl10
                               } else {
                                       lassign $lsline cd1 cd2 cd3 cd4 cd5 cd6 cd7 cd8 cd9
cd10
                               }
                       }
                       close $lowfile
                       set lsnewfile [ open lstemp "w" ]
                       for_file lsnewline case$i {
                               lassign $lsnewline w1
                               if {$w1 == "CLTO=" } {
                                       puts $lsnewfile [format "
CLTO=%6.4f,%6.4f,%6.4f,%6.4f,%6.4f,%6.4f,%6.4f,%6.4f,%6.4f,%6.4f," $cl1 $cl2 $cl3 $cl4
$cl5 $cl6 $cl7 $cl8 $cl9 $cl10]
                               } else {
                                       puts $lsnewfile $lsnewline
                               }
                       }
                       close $lsnewfile
                       exec mv lstemp case$i
                       set lsnewfile [ open lstemp "w" ]
                       for_file lsnewline case$i {
                               lassign $lsnewline w3
                               if {$w3 == "CLLD=" } {
                                       puts $lsnewfile [format "
CLLD=%6.4f,%6.4f,%6.4f,%6.4f,%6.4f,%6.4f,%6.4f,%6.4f,%6.4f,%6.4f," $cl1 $cl2 $cl3 $cl4
$cl5 $cl6 $cl7 $cl8 $cl9 $cl10]
                               } else {
                                       puts $lsnewfile $lsnewline
                               }
                       }
                       close $lsnewfile
                       exec mv lstemp case$i
                       set lsnewfile [ open lstemp "w" ]
                       for_file lsnewline case$i {
                               lassign $lsnewline w4
                               if {$w4 == "CDLD=" } {
                                       puts $lsnewfile [format
"CDLD=%6.4f,%6.4f,%6.4f,%6.4f,%6.4f,%6.4f,%6.4f,%6.4f,%6.4f,%6.4f," $cd1 $cd2 $cd3 $cd4
$cd5 $cd6 $cd7 $cd8 $cd9 $cd10]
                               } else {
                                       puts $lsnewfile $lsnewline
                               }
                       }
                       close $lsnewfile
                       exec mv lstemp case$i
```

```tcl
                    set lsnewfile [ open lstemp "w" }
                    for_file lsnewline case$i {
                            lassign $lsnewline w4
                            if ($w4 == "CDTO=" } {
                                    puts $lsnewfile [format "
CDTO=%6.4f,%6.4f,%6.4f,%6.4f,%6.4f,%6.4f,%6.4f,%6.4f,%6.4f,%6.4f," $cd1 $cd2 $cd3 $cd4
$cd5 $cd6 $cd7 $cd8 $cd9 $cd10]
                            } else {
                                    puts $lsnewfile $lsnewline
                            }
                    }
                    close $lsnewfile
                    exec mv lstemp case$i
            }
            set line ""

            for {set j 1} { $j <= $Number_of_Sets} { incr j 1} {
                    debug puts stdout "Michelle: [lindex [lindex  [set [set temp
Matrix($i)]] 0] [expr $j-1] ] "

                    global SorR
                    if {$SorR == "Screening"} {

                            switch -- [lindex [lindex [set [set temp Matrix($i)]] 0]
[expr $j-1] ] {
                                    "-1" { set value [set [set temp Minimum$j]] }
                                    "0" { set value [expr ([set [set temp
Maximum$j]]+[set [set temp Minimum$j]])/2.] }
                                    "1" { set value [set [set temp Maximum$j]] }
                            }
                            switch -- [set [set temp Variable$j]] {
                                    "VCMN" {
                                            append line "OPTION PMACH $value \n"
                                            append line "CONFIN VCMN $value \n"
                                            append line "COPER SUBMACH $value \n"
                                            append line "MISSIN CRMACH\[1\] $value \n"
                                            append line "MISSIN CRMACH\[2\] 0.3 \n"
                                            append line "MISSIN CRMACH\[3\] $value \n"

                                    }
                                    "TCA" {
                                            append line "CONFIN TCA $value\n"
                                            append line "WTIN TOC\[1\] $value \n"
                                            append line "WTIN TOC\[2\] $value \n"
                                            append line "WTIN TOC\[3\] $value \n"
                                    }
                                    "LAMINAR" {
                                            append line "AERIN TRUW $value \n"
                                            append line "AERIN TRLW $value \n"
                                            append line "AERIN TRUH $value \n"
                                            append line "AERIN TRLH $value \n"
                                            append line "AERIN TRUV $value \n"
                                            append line "AERIN TRLV $value \n"
                                            append line "AERIN TRUN $value \n"
                                            append line "AERIN TRLN $value \n"
                                    }
                                    "SL" {
                                            if { $value <= 5000. } {
                                                    append line "COPER SL\[2\] $value \n"
                                            } elseif { $value == 5000. } {
                                                    append line "COPER SL\[2\] $value \n"
                                            } elseif { $value >= 5000. } {
                                                    append line "COPER SL\[2\] $value \n"
                                            }

                                    }
                                    "LC" {
                                            append line "CMAN LEARN1 $value \n"
                                            append line "CMAN LEARN2 $value \n"
```

```tcl
                                append line "CMAN LEARNA1 $value \n"
                                append line "CMAN LEARNA2 $value \n"
                                append line "CMAN LEARNAS1 $value \n"
                                append line "CMAN LEARNAS2 $value \n"
                                append line "CMAN LEARNFE1 $value \n"
                                append line "CMAN LEARNFE2 $value \n"
                    }
                "LR" {
                                append line "CMAN RE $value \n"
                                append line "CMAN RT $value \n"
                                append line "COPER RL $value \n"

                    }
                "LF" {
                            append line "COPER CLF [set [set temp
Leader$j]]$value \n"

                            append line "COPER FLF [set [set temp
Leader$j]]$value \n"

                    }
                "NV" {
                                    append line "[set [set temp
Namelist$j]] [set [set temp Variable$j]] [set [set temp Leader$j]]$value \n "
                                if { $value == 798. } {
                                        append line "CMAN NVEH(1) $value \n"
                                        append line "CMAN RATE\[1\] 2.   \n"
                                        append line "CMAN RATE\[2\] 4.   \n"
                                        append line "CMAN RATE\[3\] 5.   \n"
                                        append line "CMAN RATE\[4\] 5.5  \n"
                                        append line "CMAN RATE\[5\] 6.   \n"
                                        append line "CMAN RATE\[6\] 6.   \n"
                                        append line "CMAN RATE\[7\] 6.5  \n"
                                        append line "CMAN RATE\[8\] 7.   \n"
                                        append line "CMAN RATE\[9\] 7.   \n"
                                        append line "CMAN RATE\[10\] 6.5  \n"
                                        append line "CMAN RATE\[11\] 6.   \n"
                                        append line "CMAN RATE\[12\] 5.   \n"
                                } elseif { $value == 549. } {
                                        append line "CMAN NVEH(1) $value \n"
                                        append line "CMAN RATE\[1\] 2.   \n"
                                        append line "CMAN RATE\[2\] 4.   \n"
                                        append line "CMAN RATE\[3\] 5.   \n"
                                        append line "CMAN RATE\[4\] 5.5  \n"
                                        append line "CMAN RATE\[5\] 6.   \n"
                                        append line "CMAN RATE\[6\] 6.   \n"
                                        append line "CMAN RATE\[7\] 6.5  \n"
                                        append line "CMAN RATE\[8\] 6.5  \n"
                                        append line "CMAN RATE\[9\] 4.25  \n"
                                        append line "CMAN RATE\[10\] 0.   \n"
                                        append line "CMAN RATE\[11\] 0.   \n"
                                        append line "CMAN RATE\[12\] 0.   \n"
                                } elseif { $value == 300. } {
                                        append line "CMAN NVEH(1) $value \n"
                                        append line "CMAN RATE\[1\] 1.   \n"
                                        append line "CMAN RATE\[2\] 2.   \n"
                                        append line "CMAN RATE\[3\] 3.   \n"
                                        append line "CMAN RATE\[4\] 3.   \n"
                                        append line "CMAN RATE\[5\] 3.   \n"
                                        append line "CMAN RATE\[6\] 4.   \n"
                                        append line "CMAN RATE\[7\] 4.   \n"
                                        append line "CMAN RATE\[8\] 3.   \n"
                                        append line "CMAN RATE\[9\] 2.   \n"
                                        append line "CMAN RATE\[10\] 0.   \n"
                                        append line "CMAN RATE\[11\] 0.   \n"
                                        append line "CMAN RATE\[12\] 0.   \n"
                                }
                    }
            }
            "default" {
                        debug puts stdout "$i $j $value"
```

```tcl
                                                    append line "[set [set temp Namelist$j]] [set
[set temp Variable$j]] [set [set temp Leader$j]]$value \n "
                                }
                        }
                } else {
#       puts stdout "Michelle RSE  $alpha [set [set temp Variable$j]] [set [set temp
Minimum$j]] [set [set temp Maximum$j]]"
                                set low [expr -1*$alpha]
                                set high [expr $alpha]
#puts stdout "$low $high"
                                switch -- [lindex [lindex [set [set temp Matrix($i)]] 0]
[expr $j-1] ] {
                                        "-1" {
                                                set mid [expr ([set [set temp
Maximum$j]]+[set [set temp Minimum$j]])/2.]
                                                if { $alpha == 1. } {
                                                        set value [set [set temp Minimum$j]]
                                                } else {
                                                        set value [expr ($mid - ([set [set
temp Maximum$j]]-$mid)/$alpha)]
                                                }
                                        }
                                        "0" { set value [expr ([set [set temp
Maximum$j]]+[set [set temp Minimum$j]])/2.] }
                                        "1" {
                                                if { $alpha == 1. } {
                                                        set value [set [set temp Maximum$j]]
#puts stdout "$value test1"
                                                } else {
                                                        set mid [expr ([set [set temp
Maximum$j]]+[set [set temp Minimum$j]])/2.]
                                                        set value [expr ($mid + ([set [set
temp Maximum$j]]-$mid)/$alpha)]
                                                }
                                        }
                                        "default" {
                                                if { [lindex [lindex [set [set temp
Matrix($i)]] 0] [expr $j-1] ] == $low } {
                                                        set value [set [set temp Minimum$j]]
#puts stdout "$value test2"
                                                } else { [lindex [lindex [set [set temp
Matrix($i)]] 0] [expr $j-1] ] == $high } {
                                                        set value [set [set temp Maximum$j]]
                                                }
                                        }
                                }
                                switch -- [set [set temp Variable$j]] {
                                        "LAMINAR" {
                                                append line "AERIN TRUW $value \n"
                                                append line "AERIN TRLW $value \n"
                                                append line "AERIN TRUH $value \n"
                                                append line "AERIN TRLH $value \n"
                                                append line "AERIN TRUV $value \n"
                                                append line "AERIN TRLV $value \n"
                                                append line "AERIN TRUN $value \n"
                                                append line "AERIN TRLN $value \n"
                                        }
                                        "VCMN" {
                                                append line "OPTION PMACH $value \n"
                                                append line "CONFIN VCMN $value \n"
                                                  append line "COPER SUBMACH $value \n"
                                                append line "MISSIN CRMACH\[1\] $value \n"
                                                append line "MISSIN CRMACH\[2\] 0.3 \n"
                                                append line "MISSIN CRMACH\[3\] $value \n"
                                        }
                                        "TCA" {
                                                append line "CONFIN TCA $value\n"
```

```
                                                append line "WTIN TOC\[1\] $value \n"
                                                append line "WTIN TOC\[2\] $value \n"
                                                append line "WTIN TOC\[3\] $value \n"
                                        }
                                        "SL" {

                                                if { $value <= 5000. } {
                                                        append line "COPER SL\[2\] $value \n"
                                                } elseif { $value == 5000. } {
                                                        append line "COPER SL\[2\] $value \n"
                                                } elseif { $value >= 5000. } {
                                                        append line "COPER SL\[2\] $value \n"
                                                }

                                        }
                                        "LC" {

                                                append line "CMAN LEARN1 $value \n"
                                                append line "CMAN LEARN2 $value \n"
                                                append line "CMAN LEARNA1 $value \n"
                                                append line "CMAN LEARNA2 $value \n"
                                                append line "CMAN LEARNAS1 $value \n"
                                                append line "CMAN LEARNAS2 $value \n"
                                                append line "CMAN LEARNFE1 $value \n"
                                                append line "CMAN LEARNFE2 $value \n"

                                        }
                                            "LR" {

                                                    append line "CMAN RE $value \n"
                                                    append line "CMAN RT $value \n"
                                                    append line "COPER RL $value \n"

                                            }
                                        "LF" {

                                                append line "COPER CLF [set [set temp
Leader$j]]$value \n"

                                                append line "COPER FLF [set [set temp
Leader$j]]$value \n"

                                        }
                                        "NV" {

                                                        append line "[set [set temp
Namelist$j]] [set [set temp Variable$j]] [set [set temp Leader$j]]$value \n "
                                                if { $value == 798. } {
                                                        append line "CMAN NVEH(1) $value \n"
                                                        append line "CMAN RATE\[1\] 2.    \n"
                                                        append line "CMAN RATE\[2\] 4.    \n"
                                                        append line "CMAN RATE\[3\] 5.    \n"
                                                        append line "CMAN RATE\[4\] 5.5   \n"
                                                        append line "CMAN RATE\[5\] 6.    \n"
                                                        append line "CMAN RATE\[6\] 6.    \n"
                                                        append line "CMAN RATE\[7\] 6.5   \n"
                                                        append line "CMAN RATE\[8\] 7.    \n"
                                                        append line "CMAN RATE\[9\] 7.    \n"
                                                        append line "CMAN RATE\[10\] 6.5  \n"
                                                        append line "CMAN RATE\[11\] 6.   \n"
                                                        append line "CMAN RATE\[12\] 5.   \n"
                                                } elseif { $value == 549. } {
                                                        append line "CMAN NVEH(1) $value \n"
                                                        append line "CMAN RATE\[1\] 2.    \n"
                                                        append line "CMAN RATE\[2\] 4.    \n"
                                                        append line "CMAN RATE\[3\] 5.    \n"
                                                        append line "CMAN RATE\[4\] 5.5   \n"
                                                        append line "CMAN RATE\[5\] 6.    \n"
                                                        append line "CMAN RATE\[6\] 6.    \n"
                                                        append line "CMAN RATE\[7\] 6.5   \n"
                                                        append line "CMAN RATE\[8\] 6.5   \n"
                                                        append line "CMAN RATE\[9\] 4.25  \n"
                                                        append line "CMAN RATE\[10\] 0.   \n"
                                                        append line "CMAN RATE\[11\] 0.   \n"
                                                        append line "CMAN RATE\[12\] 0.   \n"
                                                } elseif { $value == 300. } {
                                                        append line "CMAN NVEH(1) $value \n"
```

```
                                    append line "CMAN RATE\[1\] 1.  \n"
                                    append line "CMAN RATE\[2\] 2.  \n"
                                    append line "CMAN RATE\[3\] 3.  \n"
                                    append line "CMAN RATE\[4\] 3.  \n"
                                    append line "CMAN RATE\[5\] 3.  \n"
                                    append line "CMAN RATE\[6\] 4.  \n"
                                    append line "CMAN RATE\[7\] 4.  \n"
                                    append line "CMAN RATE\[8\] 3.  \n"
                                    append line "CMAN RATE\[9\] 2.  \n"
                                      append line "CMAN RATE\[10\] 0.  \n"
                            append line "CMAN RATE\[11\] 0.  \n"
                            append line "CMAN RATE\[12\] 0.  \n"
                         }
                      }
                      "default" {
                                    puts stdout "$i $j $value"
                                append line "[set [set temp Namelist$j]] [set
[set temp Variable$j]] [set [set temp Leader$j]]$value \n "
                                }
                      }
                  }
               }

            echo $line | tsw -input case$i -output temp -i case
            debug puts stdout "echo $line | tsw -input case$i -output temp -"
            update
            catch { exec echo $line | tsw -input case$i -output temp -  }
            update
            exec mv temp case$i
            update
         }

      Message "Creating Case Files is now completed."
      .stars.bottom.frame.run conf -state normal
      .stars.bottom.frame.kirby conf -state normal
      .stars.bottom.frame.aero conf -state normal
      .stars.bottom.frame.flops585 conf -state normal
      .stars.bottom.frame.flops57 conf -state normal
   }
```

Kirby.tcl

This procedure controls the execution of FLOPS/ALCCA and extracts the response surface data desired, i.e. TOWG, Fuel weight, Empty weight, Wing weight, VAPP, LdgFL, TOFL, block time, Acquisition cost, and RDT&E costs. Also, there is a time limit of 7 minutes put on any one case file execution. This is more than an adequate limit for a normal case, but, if one cases were to fail and get trapped in an infinite loop, the script would halt the execution of that case and continue to the next.

```
proc Kirby {} {

        .stars.bottom.frame.flops585 conf -state disabled
        .stars.bottom.frame.flops57 conf -state disabled
        .stars.bottom.frame.kirby conf -state disabled
        .stars.bottom.frame.run conf -state disabled
        .stars.bottom.frame.aero conf -state disabled

       global Number_of_Sets Matrix Number_of_Cases

#              open the ouput files to write info desired


#       set output1 [open "Fuel_capacity" "w" ]
        set output2 [open "Wing_wt" "w" ]
        set output3 [open "Empty_wt" "w" ]
        set output4 [open "Fuel_wt" "w" ]
        set output5 [open "TOGW" "w" ]
#       set output6 [open "S_Wing" "w" ]
        set output7 [open "VAPP" "w" ]
        set output8 [open "LDGFL" "w" ]
        set output9 [open "TOFL" "w" ]
        set output10 [open "TIME" "w" ]
        set output11 [open "ACQ" "w" ]
        set output12 [open "RDTE" "w" ]


       for {set i 1} { $i <= $Number_of_Cases} { incr i 1} {

            update

            Message "Running FLOPS/ALCCA V5.85 Case File $i. Please Wait."
#           debug puts stdout "exec flops_alcca < case$i > case$i.out"

#           set test [catch "exec flops_alcca < case$i > case$i.out "]


#           Five Minutes

            set MAX_TIME [expr 7*60]
            set START_TIME [getclock]
            set PID [exec flops_alcca < case$i > case$i.out &]

            while { 1 } {
                    update
                    switch [catch {exec ps -p $PID} ] {
                          "0" {
                                   set TIME [expr [getclock]-$START_TIME]
                                   if { $TIME >= $MAX_TIME } {
                                          catch { kill 9 $PID }
                                          Message "case$i has been KILLED. Run time
is greater than $MAX_TIME seconds."
                                          update
                                          sleep 2
                                          break
                                   }
                          }
                          "1" {

#          grep out the info that you want
#

#                                 set line [exec grep "TOTAL FUEL CAPACITY" case$i.out
]
#                                 puts $output1 "Case $i:   $line"
```

```
                                        set line [exec grep "            WING " case$i.out ]
                                        puts $output2 "Case $i:  $line"

                                        set line [exec grep "     OPERATING WEIGHT EMPTY"
case$i.out ]

                                        puts $output3 "Case $i:  $line"

                                        set line [exec grep "          MISSION FUEL"
case$i.out ]

                                        puts $output4 "Case $i:  $line"

                                        set line [exec grep "RAMP (GROSS) WEIGHT" case$i.out
]
                                        puts $output5 "Case $i:  $line"

#              set line [exec grep "     REFERENCE WING AREA"  case$i.out ]
#              puts $output6 "Case $i:  $line"

                                        set line [exec grep "VAPP=" case$i.out ]
                                        puts $output7 "Case $i:  $line"

                                        set line [exec grep "FARLDG="  case$i.out ]
                                        puts $output8 "Case $i:  $line"

                                        set line [exec grep "FAROFF=" case$i.out ]
                                        puts $output9 "Case $i:  $line"

                                        set line [exec grep "BLOCK TIME"  case$i.out ]
                                        puts $output10 "Case $i:  $line"

                                        set line [exec grep "AVERAGE UNIT AIRPLANE COST
(including spares)"  case$i.out ]

                                        puts $output11 "Case $i:  $line"

                                        set line [exec grep "RESEARCH,DEVELOPMENT,TEST,AND
EVALUATION"  case$i.out ]

                                        puts $output12 "Case $i:  $line"


                                        exec mv drag_stuff drag_stuff$i

                                        break

                            }

                        }

                }

            }


#
#              close the ouptut files of interest
#

#      close $output1
       close $output2
       close $output3
       close $output4
       close $output5
#      close $output6
       close $output7
       close $output8
       close $output9
       close $output10
       close $output11
```

```
        close $output12


        Message "FLOPS Version 5.85 execution is now completed!!"

}
```

Low Speed Script and Codes

RUN

This shell script controls the extracting of information from the hermite geometry file, execution of the "convert" code, recompilation of the information from "convert" into input files for BDAP and AERO2S, execution of BDAP and AERO2S, and extraction of low speed drag polar information from those outputs.

```
#!/usr/local/bin/wishx -f

# for RS #!/usr/local/bin/wishx -f

#   for gypsy #!/usr/X11R6/bin/wishx -f


exec mv mission flight
set flight [open "flight" "r"]
set mission [open "mission.info" "w"]

while ![eof $flight] {
        set line [gets $flight]
        if [eof $flight] break
        if {$line == ""} break
        lassign $line case mach alt
}



set ifile [open "baseline.hrm" "r"]
set fuse [open "fuse.info" "w"]
set wing [open "wing.info" "w"]
set v_tail [open "v_tail.info" "w"]
set h_tail [open "h_tail.info" "w"]
set engine [open "engine.info" "w"]

set output1 [open "bdap.in" "w"]
set output2 [open "aero2s.in" "w"]
set geom [open "baseline.ram" "r"]
```

```
# Get fuselage information from the hermite file from RAM
#       and write it to a temporary file called fuse.info

for { set i 1 } { $i <= 9 } { incr i 1 } {
        set temp "[gets $ifile]"
}

set j 1
while ![eof $ifile] {
        set line [gets $ifile]
        if [eof $ifile] break
        if {$line == ""} break
        lassign $line X Y Z
        incr j 1
        if { $X <= 1.0 && $j >= 20 } continue
        if {$Y < 0.0} continue
        if { $X > 0.001 && $X < 1.0 } continue
        if { $j > 976} break
        puts $fuse "$line"
}
puts $fuse ""

# Note: I had to insert an extra line at the end of each info file
#       because when the fortran program "convert" ran, it would
#       reach the end of the file and result in a core dump, not
#       one to be proud of I might add, and would halt execution

# Get engine information from the hermite file from RAM
#       and write it to a temporary file called engine.info


while ![eof $ifile] {
        set line [gets $ifile]
        if [eof $ifile] break
        if {$line == ""} break
}


for { set i 1 } { $i <= 5 } { incr i 1 } {
        set temp "[gets $ifile]"
}

while ![eof $ifile] {
        set line [gets $ifile]
        if [eof $ifile] break
        if {$line == ""} break
        lassign $line X Y Z
        puts $engine "$line"
}
puts $engine ""


# Get horizontal tail information from the hermite file from RAM
#       and write it to a temporary file called h_tail.info


for { set i 1 } { $i <= 5 } { incr i 1 } {
        set temp "[gets $ifile]"
}

while ![eof $ifile] {
        set line [gets $ifile]
        if [eof $ifile] break
        if {$line == ""} break
        lassign $line X Y Z
        puts $h_tail "$line"
```

```
    }
    puts $h_tail ""



    # Get vertical tail information from the hermite file from RAM
    #       and write it to a temporary file called v_tail.info


    for { set i 1 } { $i <= 5 } { incr i 1 } {
            set temp "[gets $ifile]"
    }

    while ![eof $ifile] {
            set line [gets $ifile]
            if [eof $ifile] break
            if {$line == ""} break
            lassign $line X Y Z
            puts $v_tail "$line"
    }
    puts $v_tail ""



    # Get wing information from the hermite file from RAM
    #       and write it to a temporary file called wing.info

    for { set i 1 } { $i <= 5 } { incr i 1 } {
            set temp "[gets $ifile]"
    }

    while ![eof $ifile] {
            set line [gets $ifile]
            if [eof $ifile] break
            if {$line == ""} break
            lassign $line X Y Z
            puts $wing "$line"
    }
    puts $wing ""



    close $fuse
    close $wing
    close $v_tail
    close $h_tail
    close $engine
    close $ifile

    set geom [open "baseline.ram" "r"]

    set i 1
    while ![eof $geom] {
            incr i 1
            set line [gets $geom]
            if [eof $geom] break
            if { $i == 359 } {
                    lassign $line chord dummy dummy
            }
    }


    close $geom


    #puts stdout "$mach $alt $chord"
    puts $mission "$mach          $alt          $chord"
```

```
        puts $mission ""

        close $mission
        close $flight

        puts stdout "Running convert"
        exec convert
        puts stdout "Finished running convert"

        exec mv flight mission
        exec rm mission.info

        set bdap [open "bdap.info" "r"]

        # Formatting bdap.in through various puts statements and scripts

        puts $output1 "GEOM NEW"
        puts $output1 "GLOBAL AERO ANALYSIS STUDY"
        #
        12345678901234567890123456789012345678901234567890123456789012345678901234567890
        puts $output1 "  1   1   1   1   1   1   0 20 12   1 17 20                    1 15   1  10 1 10
        CONTROL"


        #  Scan through the *.ram  file to get the reference area of the wing

        set geom [open "baseline.ram" "r"]
        set i 1
        while ![eof $geom] {
                incr i 1
                 set line [gets $geom]
                 if [eof $geom] break
                if { $i == 357 } {
                        lassign $line ref_area dummy dummy
                }
        }

        close $geom
        set geom [open "baseline.ram" "r"]

        set i 1
        while ![eof $geom] {
                incr i 1
                 set line [gets $geom]
                 if [eof $geom] break
                if { $i == 360 } {
                        lassign $line xbar dummy dummy dummy dummy
                }
        }

        puts $output1 [format "%7.1f%7.2f%7.2f" $ref_area $chord $xbar]



        puts $output1 "    .00    .50    .75   2.50   7.50  20.00  30.00  50.00  70.00  85.00
        XAF"
        puts $output1 "  95.00 100.00
        XAF"

        #  Grab most of the info required for BDAP from the file bdap.info
        #       which was written by convert program

        while ![eof $bdap] {
                set line [gets $bdap]
                if [eof $bdap] break
                if {$line == ""} break
                puts $output1 "$line"
```

```
}

#   Now print the controls for the SKFR module in BDAP
#       Eventually, I want to read a Mach number from somewhere
#       and automatically put it in the file

set temp [expr $alt/1000]
puts $output1 "SKFR"
puts $output1 "SKIN FOR GRA"
puts $output1 "1   1   1   1   1"
puts $output1 "1."
puts $output1 [format "%4.2f    %4.1f    0.0     1.0" $mach $temp]
#puts $output1 "0.82    42.0    0.0     1.0"
puts $output1 "END"



close $output1

puts stdout "Running BDAP"

exec cp bdap.in lar12237.input
set test [catch "exec lar.exe"]

exec cp lar12237.output bdap.out
eval exec rm [glob lar*]


puts stdout "Finished running BDAP: results in file bdap.out"


#   Start to format input for AERO2S input file
#       Note: All gaps imply that something is missing there or needed


#close $newwing
#set newwing [open "wing.stuff" "r"]
#set i 1
#while ![eof $newwing] {
#        incr i 1
#        set line [gets $newwing]
#        if [eof $newwing] break
#        if {$i == 21} {
#               set half_span [string range [string trimleft $line] 6 12]
#        }
#}


set aeroinfo [open "aero.info" "r"]
while ![eof $aeroinfo] {
        set line [gets $aeroinfo]
        if [eof $aeroinfo] break
        if {$line == "" } {
                puts $output2 [format " SREF = %7.3f," $ref_area]
                puts $output2 [format " CBAR = %7.3f," $chord]
                puts $output2 [format " XMC = %7.3f," $xbar]
        } else {
                puts $output2 "$line"
        }
}

close $aeroinfo

set thick [open "thick$case" "r"]
set i 1
while ![eof $thick] {
        set line [gets $thick]
```

```
        · if [eof $thick] break
        if { $i == 1} {
                lassign $line tbtoc tbeta tbroc
        }
        incr i 1
}
puts $output2 [format " TBTOC = %5.3f,%5.3f," $tbtoc $tbtoc]
puts $output2 [format " TBETA = %5.3f,%5.3f," $tbeta $tbeta]
puts $output2 [format " TBROC = %5.3f,%5.3f," $tbroc $tbroc]

close $thick

set aeroinfo2 [open "aero.info2" "r"]
while ![eof $aeroinfo2] {
        set line [gets $aeroinfo2]
        if [eof $aeroinfo2] break
        puts $output2 "$line"
}

close $aeroinfo2

set hthick [open "hthick" "r"]
set i 1
while ![eof $hthick] {
        set line [gets $hthick]
        if [eof $hthick] break
        if { $i == 1} {
                lassign $line tbtoc2 tbeta2 tbroc2
        }
        incr i 1
}
puts $output2 [format " TBTOC2 = %5.3f,%5.3f," $tbtoc2 $tbtoc2]
puts $output2 [format " TBETA2 = %5.3f,%5.3f," $tbeta2 $tbeta2]
puts $output2 [format " TBROC2 = %5.3f,%5.3f," $tbroc2 $tbroc2]
puts $output2 " \$END"

close $hthick
close $output2

puts stdout "Running AERO2S...  Please hold"

set test2 [catch "exec aero2s < aero2s.in > aero2s.out"]

puts stdout "Parsing aero2s.out and bdap.out"

#  To clean up the current directory and free up precious space
#       I am removing most of the temporary files. Only the files
#       needed for output data extraction will remain. If you want
#       to keep these files then just comment the lines with a '#'

eval exec rm [glob *.info]
eval exec rm [glob fort.*]
eval exec rm [glob *.info2]




# Aero2s parser version 2.0


# This parser requires modifications to the Aero2s source code which precede
#       each output line to be parsed with the characters: "a?".  These modifications were
#        made by Michelle Kirby.  Please contact her at gt0038c@cad.gatech.edu for
#        the details.

        set file [open "aero2s.out" r]
        set EOF READ
```

```tcl
        set i 1

#Span the file until the section of interest is reached
        set word garbage
        while {$word != "a?"} {
                set next_line [gets $file]
                scan $next_line "%s %f %f %f %f %f %f %f %f %f %f %f %f %f %f"\
                        word alpha($i) dummy dummy dummy dummy dummy dummy dummy \
                        dummy dummy cm($i) cl($i) cd($i) dummy
        }

# Parse the desired data into an array
        while {$word == "a?"} {
                incr i 1
                set next_line [gets $file]
                scan $next_line "%s %f %f %f %f %f  %f %f %f %f %f %f %f %f %f"\
                        word alpha($i) dummy dummy dummy dummy dummy dummy dummy \
                        dummy dummy cm($i) cl($i) cd($i) dummy
        }


set lowspeed [open "low" "w"]

#       Go into procedure parse_aero2s and determine how many values are in the arrays
#               cl, cd, cm, and ss and get out the values and place them into an
#               associated "line" and then write that line to the appropriate output fil


        foreach j [lsort -increasing -integer [array names cl]] {

                append line1 "$cl($j) "

        }

        foreach j [lsort -increasing -integer [array names cd]] {

                append line2 "$cd($j) "

        }



# Parse the bdap output file to get the skin friction drag

set bdap [open "bdap.out" "r"]
set word1 garbage

        while {$word1 != "0TOTAL"} {

                set next_line [gets $bdap]
                lassign $next_line word1 dummy dummy cdfsub

        }

puts stdout "$cdfsub"

        foreach j  [lsort -increasing -integer [array names cd]] {

                set cd($j) [ expr $cdfsub + $cd($j) ]
                append line3 "$cd($j) "
        }

        puts $lowspeed "$line1"
        puts $lowspeed "$line3"

close $file
close $bdap
```

```
close $lowspeed
exec mv low low$case

puts stdout "Execution complete....!!!"

exit
```

CONVERT.f

"Convert" is a FORTRAN code utilized to convert the hermite geometry definition into information compatible with AERO2S and BDAP. The output files written are used by "RUN" to create those input files.

```
PROGRAM CONVERT
C2345678901234567890123456789012345678901234567890123456789012
       REAL WXLE(20), WYLE(20), WZLE(20), WXTE(20), WYTE(20), WZTE(20)
       REAL VXLE(20), VYLE(20), VZLE(20), VXTE(20), VYTE(20), VZTE(20)
       REAL HXLE(20), HYLE(20), HZLE(20), HXTE(20), HYTE(20), HZTE(20)
       DOUBLE PRECISION WXUP(20,10), WYUP(20,10), WZUP(20,10)
       DOUBLE PRECISION WXLO(20,10), WYLO(20,10), WZLO(20,10)
       REAL VXUP(20,10), VYUP(20,10), VZUP(20,10),VTCH(12),VTHICK(12)
       REAL VXLO(20,10), VYLO(20,10), VZLO(20,10)
       REAL HXUP(20,10), HYUP(20,10), HZUP(20,10),HTCH(12),HTHICK(12)
       REAL HXLO(20,10), HYLO(20,10), HZLO(20,10)
       REAL FX(25,18), FY(25,18), FZ(25,18)
       DOUBLE PRECISION Q1, Q2, Q3, Q4, Q5, Q6
       DOUBLE PRECISION UCFA3(20), UCFA2(20), UCFA1(20), UCFA0(20)
       DOUBLE PRECISION LCFA3(20), LCFA2(20), LCFA1(20), LCFA0(20)
       REAL UQF1A2(20), UQF1A1(20), UQF1A0(20)
       REAL LQF1A2(20), LQF1A1(20), LQF1A0(20)
       REAL UQF2A2(20), UQF2A1(20), UQF2A0(20)
       REAL LQF2A2(20), LQF2A1(20), LQF2A0(20)
       DOUBLE PRECISION ZORD(20,12), THICK(20,12)
       REAL X(46),Y(46),Z(46)
       REAL VX(46),VY(46),VZ(46)
       REAL HX(46),HY(46),HZ(46),CHORD(20)
       REAL DY,WLESW,WTESW,t,MACH,ALT,MAC,RENUM,SOLN
       DOUBLE PRECISION TEMP,A,B, XCHORD(20,12)
       REAL ENGX(16,21),ENGY(16,21),ENGZ(16,21), RADII(15)

       REAL TBLEY(3), TBLEX(3), TBYC(20),TZORDC(20,12)
       REAL TBTEY(3), TBTEX(3),TEMP2,Y1,Y2

       INTEGER I,J,K,UPPER,LOWER


       OPEN(UNIT=11,FILE='fuse.info',STATUS='UNKNOWN')
       OPEN(UNIT=12,FILE='wing.info',STATUS='UNKNOWN')
       OPEN(UNIT=13,FILE='v_tail.info',STATUS='UNKNOWN')
       OPEN(UNIT=14,FILE='h_tail.info',STATUS='UNKNOWN')
       OPEN(UNIT=15,FILE='engine.info',STATUS='UNKNOWN')
       OPEN(UNIT=16,FILE='mission.info',STATUS='UNKNOWN')
       OPEN(UNIT=20,FILE='aero.info',STATUS='UNKNOWN')
```

```
        OPEN(UNIT=25,FILE='bdap.info',STATUS='UNKNOWN')
        OPEN(UNIT=26,FILE='aero.info2',STATUS='UNKNOWN')


C***********************************************************************
C       Variable Definition

C FUSELAGE

C       FX      = Array of fuselage x-location
C       FY      = Array of fuselage y-location
C       FZ      = Array of fuselage z-location

C***********************************************************************
C       Variable Definition
C WING

C  WXLE(I)      = Array of wing LE x-location
C  WYLE(I)      = Array of wing LE y-location
C  WZLE(I)      = Array of wing LE z-location
C  Note: The airfoil definition is for I for inboard to outboard, and
C               J from LE to TE
C  WXUP(I,J)    = Array of wing upper airfoil definition x-location
C  WYUP(I,J)    = Array of wing upper airfoil definition y-location
C  WZUP(I,J)    = Array of wing upper airfoil definition z-location
C  WXLO(I,J)    = Array of wing lower airfoil definition x-location
C  WYLO(I,J)    = Array of wing lower airfoil definition y-location
C  WZLO(I,J)    = Array of wing lower airfoil definition z-location
C  ZORD(I,J)    = Array of z-ordinates WRT WZLE(I) to define the wing
C                 camber
C  CHORD(I)     = Array of wing chord lengths from inboard to outboard
C  Note: The following variables were used to define the airfoil
C               camber based on fitting three curves to the wing
C               airfoil surface. The first was a cubic for the LE of
C               the airfoil, and two successive quadratic curves to
C               define the remainder of the airfoil.

C  CUBIC follows the following form
C
C      Z = A0 + A1*X + A2*X**2 + A3*X**3
C
C  Q1 to Q6     = Temporary variables to reduce the length of the
C                 actual coefficient equations
C
C  UCFA0(I,J)   = Array of constant coefficients for upper surface
C  UCFA1(I,J)   = Array of linear coefficients for upper surface
C  UCFA2(I,J)   = Array of quadratic coefficients for upper surface
C  UCFA3(I,J)   = Array of cubic coefficients for upper surface
C  LCFA0(I,J)    = Array of constant coefficients for lower surface
C  LCFA1(I,J)    = Array of linear coefficients for lower surface
C  LCFA2(I,J)    = Array of quadratic coefficients for lower surface
C  LCFA3(I,J)    = Array of cubic coefficients for lower surface

C  QUADRATIC follows the following form
C
C      Z = A0 + A1*X + A2*X**2
C
C  UQF1A0(I,J) = Array of constant coef for middle upper surface
C  UQF1A1(I,J) = Array of linear coef for middle upper surface
C  UQF1A2(I,J) = Array of quadratic coef for middle upper surface
C  LQF1A0(I,J)  = Array of constant coef for middle lower surface
C  LQF1A1(I,J)  = Array of linear coef for middle lower surface
C  LQF1A2(I,J)  = Array of quadratic coef for middle lower surface

C  UQF2A0(I,J)  = Array of constant coef for aft upper surface
C  UQF2A1(I,J)  = Array of linear coef for aft upper surface
C  UQF2A2(I,J)  = Array of quadratic coef for aft upper surface
```

```
C  LQF2A0(I,J)  = Array of constant coef for aft lower surface
C  LQF2A1(I,J)  = Array of linear coef for aft lower surface
C  LQF2A2(I,J)  = Array of quadratic coef for aft lower surface

C  THICK(I,J)   = Array of wing airfoil half thickness expressed
C                    in % chord
C  XCHORD(I,J) = Array of % chord to defin ZORD and THICK
C  A            = Temporary variable for finding ZORD and THICK
C  B            = Temporary variable for finding ZORD and THICK


C***********************************************************************
C         Variable Definition
C  VERTICAL TAIL
C
C  VX(I)        = Temporary array for reading x from file v_tail.info
C  VY(I)        = Temporary array for reading y from file v_tail.info
C  VZ(I)        = Temporary array for reading z from file v_tail.info

C  VXLE(I)      = Array of vertical tail LE x-location
C  VYLE(I)      = Array of vertical tail LE y-location
C  VZLE(I)      = Array of vertical tail LE z-location
C  Note: The airfoil definition is for I for inboard to outboard, and
C                J from LE to TE
C  VXUP(I,J)    = Array of RHS VT airfoil definition x-location
C  VYUP(I,J)    = Array of RHS VT airfoil definition y-location
C  VZUP(I,J)    = Array of RHS VT airfoil definition z-location
C  VXLO(I,J)    = Array of LHS VT airfoil definition x-location
C  VYLO(I,J)    = Array of LHS VT airfoil definition y-location
C  VZLO(I,J)    = Array of LHS VT airfoil definition z-location
C  VTCH(I)      = Array of VT chord lengths for inboard to outboard
C  VTHICK(I)    = Array of VT airfoil half thickness expressed
C                    in % chord


C***********************************************************************
C         Variable Definition
C  HORIZONTAL TAIL
C
C  HX(I)        = Temporary array for reading x from file h_tail.info
C  HY(I)        = Temporary array for reading y from file h_tail.info
C  HZ(I)        = Temporary array for reading z from file h_tail.info

C  HXLE(I)      = Array of horizontal tail LE x-location
C  HYLE(I)      = Array of horizontal tail LE y-location
C  HZLE(I)      = Array of horizontal tail LE z-location
C  Note: The airfoil definition is for I for inboard to outboard, and
C                J from LE to TE
C  HXUP(I,J)    = Array of upper HT airfoil definition x-location
C  HYUP(I,J)    = Array of upper HT airfoil definition y-location
C  HZUP(I,J)    = Array of upper HT airfoil definition z-location
C  HXLO(I,J)    = Array of lower HT airfoil definition x-location
C  HYLO(I,J)    = Array of lower HT airfoil definition y-location
C  HZLO(I,J)    = Array of lower HT airfoil definition z-location
C  HTCH(I)      = Array of HT chord lengths for inboard to outboard
C  HTHICK(I)    = Array of HT airfoil half thickness expressed
C                    in % chord


C***********************************************************************
C         Variable Definition
C  ENGINE
C
C  ENGX(I,J)   = Array of engine x-locations
C  ENGY(I,J)   = Array of engine y-locations
C  ENGZ(I,J)   = Array of engine z-locations
C  RADII(I)    = Array of engine radius definitions

C***********************************************************************
C         Variable Definition
```

```
C  MACH       = Flight condition Mach number
C  ALT        = Flight condition altitude
C  RENUM      = Reynold's number at MACH and ALT

       REWIND 20


C************************************************************************
C Read the flight conditions from the file mission and use to calculate
C       the Reynolds number. The atmospheric stuff is based on a
C       subroutine extracted from FLOPS by Peter Rhol and adjusted
C       for use here

       READ(16,*) MACH,ALT,MAC
       CALL REYNOLDS(ALT,MACH,MAC,SOLN)
       RENUM = SOLN



C************************************************************************
C Read fuselage x,y,z points from the file wing.info which was created
C       from the shell script "run" and store info into arrays which
C       are to be manipulated later for input to BDAP and AERO2S

       DO 35 I = 1, 25
         DO 30 J = 1, 18
           IF(J.EQ.18) THEN
             READ(11,*)
             FX(I,J) = 0.0
             FY(I,J) = 0.0
             FZ(I,J) = 0.0
             GOTO 35
           ENDIF
           IF(I.EQ.2.OR.I.EQ.12.OR.I.EQ.15.OR.I.EQ.18.OR.I.EQ.20) THEN
             READ(11,*)
             FX(I,J) = 0.0
             FY(I,J) = 0.0
             FZ(I,J) = 0.0
           ELSE
             READ(11,*) FX(I,J), FY(I,J), FZ(I,J)
           ENDIF

   30    CONTINUE
   35 CONTINUE

C  I am restructuring the definition of the x,y,z arrays for the
C       fuselage. The original hermite file had approximately
C       twenty five cross sections defining the fuselage dimensions.
C       Based on an evaluation of those points, I selectively picked
C       out 20 and had to then extract them from the original arrays.
C       That is what the following do loops accomplish.

       K = 0
       DO 45 I = 1, 25
         K = K+1
         DO 40 J = 1, 18
           IF(J.EQ.18) GOTO 45
           IF(I.EQ.2.OR.I.EQ.12.OR.I.EQ.15.OR.I.EQ.18.OR.I.EQ.20) GOTO45
           IF(I.GT.2.AND.I.LT.12) THEN
             FX(K-1,J) = FX(I,J)
             FY(K-1,J) = FY(I,J)
             FZ(K-1,J) = FZ(I,J)
           ELSEIF(I.GT.12.AND.I.LT.15) THEN
             FX(K-2,J) = FX(I,J)
             FY(K-2,J) = FY(I,J)
             FZ(K-2,J) = FZ(I,J)
           ELSEIF(I.GT.15.AND.I.LT.18) THEN
             FX(K-3,J) = FX(I,J)
             FY(K-3,J) = FY(I,J)
```

```
                FZ(K-3,J)  = FZ(I,J)
              ELSEIF(I.GT.18.AND.I.LT.20) THEN
                 FX(K-4,J)  = FX(I,J)
                 FY(K-4,J)  = FY(I,J)
                 FZ(K-4,J)  = FZ(I,J)
              ELSEIF(I.GT.20.AND.I.LE.25) THEN
                 FX(K-5,J)  = FX(I,J)
                 FY(K-5,J)  = FY(I,J)
                 FZ(K-5,J)  = FZ(I,J)
              ENDIF
   40      CONTINUE
   45 CONTINUE


C************************************************************************
C Read wing x,y,z points from the file wing.info which was created from
C      the shell script "run" and store info into arrays which are
C      to be manipulated later for input to BDAP and AERO2S

      K = 0
      L = 0
      DO 50 I = 1, 46
         READ(12,*) X(I),Y(I),Z(I)
   50 CONTINUE

C Note: I am hard coding here because I could not get the correct
C      logic to reformat the input to something easy to work with

      WXLE(1)  = X(12)
      WYLE(1)  = Y(12)
      WZLE(1)  = Z(12)
      WXTE(1)  = X(1)
      WYTE(1)  = Y(1)
      WZTE(1)  = Z(1)
      WXLE(20) = X(35)
      WYLE(20) = Y(35)
      WZLE(20) = Z(35)
      WXTE(20) = X(24)
      WYTE(20) = Y(24)
      WZTE(20) = Z(24)

      K =1
      DO 60 I = 11,2,-1
         WXUP(1,K)  = X(I)
         WYUP(1,K)  = Y(I)
         WZUP(1,K)  = Z(I)
         K=K+1
   60 CONTINUE

      K = 0
      DO 65 I = 13,22
         K = K+1
         WXLO(1,K)  = X(I)
         WYLO(1,K)  = Y(I)
         WZLO(1,K)  = Z(I)
   65 CONTINUE

      K = 0
      DO 70 I = 34, 25, -1
         K = K+1
         WXUP(20,K)  = X(I)
         WYUP(20,K)  = Y(I)
         WZUP(20,K)  = Z(I)
   70 CONTINUE

      K = 0
      DO 75 I = 36,45
         K = K+1
```

```
          WXLO(20,K) = X(I)
          WYLO(20,K) = Y(I)
          WZLO(20,K) = Z(I)
    75 CONTINUE

C***********************************************************************

C  NOTE:  All wing info above is correct and validated!!!!

C***********************************************************************
C Read vertical tail x,y,z points from the file wing.info which was
C       created from the shell script "run" and store info into arrays
C       which are to be manipulated later for input to BDAP and AERO2S

      DO 100 I = 1,46
            READ(13,*) VX(I), VY(I), VZ(I)
   100 CONTINUE

C Note: I am hard coding here because I could not get the correct
C       logic to reformat the input to something easy to work with

      VXLE(1)  = VX(12)
      VYLE(1)  = VY(12)
      VZLE(1)  = VZ(12)
      VXTE(1)  = VX(1)
      VYTE(1)  = VY(1)
      VZTE(1)  = VZ(1)
      VXLE(20) = VX(35)
      VYLE(20) = VY(35)
      VZLE(20) = VZ(35)
      VXTE(20) = VX(24)
      VYTE(20) = VY(24)
      VZTE(20) = VZ(24)

      K = 0
      DO 105 I = 13,22
        K=K+1
        VXUP(1,K)  = VX(I)
        VYUP(1,K)  = VY(I)
        VZUP(1,K)  = VZ(I)
   105 CONTINUE

      K = 0
      DO 110 I = 11,2,-1
        K = K+1
        VXLO(1,K)  = VX(I)
        VYLO(1,K)  = VY(I)
        VZLO(1,K)  = VZ(I)
   110 CONTINUE

      K = 0
      DO 115 I = 36, 45
        K = K+1
        VXUP(20,K)  = VX(I)
        VYUP(20,K)  = VY(I)
        VZUP(20,K)  = VZ(I)
   115 CONTINUE

      K = 0
      DO 120 I = 34,25,-1
        K = K+1
        VXLO(20,K)  = VX(I)
        VYLO(20,K)  = VY(I)
        VZLO(20,K)  = VZ(I)
   120 CONTINUE

C  Note: The above info is correct and validated
```

```
      DO 125 I=1,12
        IF(I.EQ.1.OR.I.EQ.12) THEN
          IF(I.EQ.12) THEN
            VTCH(I) = 100.0
            VTHICK(I) = 0.0
          ELSE
            VTCH(I) = 0.0
            VTHICK(I) = 0.0
          ENDIF
        ELSE
          VTCH(I)  =  (VXUP(1,I-1)-VXLE(1))/(VXTE(1)-VXLE(1))*100.0
          VTHICK(I) = (VYUP(1,I-1)-VYLO(1,I-1))/(VXTE(1)-VXLE(1))
        ENDIF
  125 CONTINUE

      DO 130 I = 1,12
        IF(I.EQ.2) THEN
          VTCH(I) = 0.0
          VTHICK(I) = 0.0
        ENDIF
  130 CONTINUE

      DO 135 I = 4, 12
         VTCH(I-2) = VTCH(I)
         VTHICK(I-2) = VTHICK(I)
  135 CONTINUE

C********************************************************************
C Read horizontal tail x,y,z points from the file wing.info which was
C       created from the shell script "run" and store info into arrays
C       which are to be manipulated later for input to BDAP and AERO2S

      DO 200 I = 1,46
           READ(14,*) HX(I), HY(I), HZ(I)
  200 CONTINUE

C Note: I am hard coding here because I could not get the correct
C       logic to reformat the input to something easy to work with

      HXLE(1)  = HX(12)
      HYLE(1)  = HY(12)
      HZLE(1)  = HZ(12)
      HXTE(1)  = HX(1)
      HYTE(1)  = HY(1)
      HZTE(1)  = HZ(1)
      HXLE(20) = HX(35)
      HYLE(20) = HY(35)
      HZLE(20) = HZ(35)
      HXTE(20) = HX(24)
      HYTE(20) = HY(24)
      HZTE(20) = HZ(24)

      K = 0
      DO 205 I = 11,2,-1
        K= K+1
        HXUP(1,K)  = HX(I)
        HYUP(1,K)  = HY(I)
        HZUP(1,K)  = HZ(I)
  205 CONTINUE

      K = 0
      DO 210 I = 13,22
        K = K+1
        HXLO(1,K)  = HX(I)
        HYLO(1,K)  = HY(I)
```

```
         HZLO(1,K) = HZ(I)
 210 CONTINUE

      K =0
      DO 215 I = 34, 25, -1
        K = K+1
        HXUP(20,K) = HX(I)
        HYUP(20,K) = HY(I)
        HZUP(20,K) = HZ(I)
 215 CONTINUE

      K = 0
      DO 220 I = 36,45
        K = K+1
        HXLO(20,K) = HX(I)
        HYLO(20,K) = HY(I)
        HZLO(20,K) = HZ(I)
 220 CONTINUE


      DO 225 I=1,12
        IF(I.EQ.1.OR.I.EQ.12) THEN
          IF(I.EQ.12) THEN
            HTCH(I) = 100.0
            HTHICK(I) = 0.0
          ELSE
            HTCH(I) = 0.0
            HTHICK(I) = 0.0
          ENDIF
        ELSE
          HTCH(I) = (HXUP(1,I-1)-HXLE(1))/(HXTE(1)-HXLE(1))*100.0
          HTHICK(I) = (HZUP(1,I-1)-HZLO(1,I-1))/(HXTE(1)-HXLE(1))
        ENDIF
 225 CONTINUE

      DO 230 I = 1,12
        IF(I.EQ.2) THEN
          HTCH(I) = 0.0
          HTHICK(I) = 0.0
        ENDIF
 230 CONTINUE

      DO 235 I = 4, 12
         HTCH(I-2) = HTCH(I)
         HTHICK(I-2) = HTHICK(I)
 235 CONTINUE


C*********************************************************************
C  I am reformatting the information obtained above on the same order
C     as is required by BDAP directly



C  Format wing info into 20 segments, i.e. x,y, and z and determine
C     the associated chord lengths .

      WLESW = ATAN((WYLE(20)-WYLE(1))/(WXLE(20)-WXLE(1)))
      WTESW = ATAN((WYTE(20)-WYTE(1))/(WXTE(20)-WXTE(1)))
      DY = (WYLE(20)-WYLE(1))/19

C  Divide the wing into 20 equally spaced sections

      DO 300 I = 1,18
        WYLE(I+1) = (I)*DY
        WYTE(I+1) = WYLE(I+1)
        WXLE(I+1) = WXLE(1)+WYLE(I+1)/(TAN(WLESW))
```

```
          WXTE(I+1) = WXTE(1)+WYLE(I+1)/(TAN(WTESW))
          WZLE(I+1) = WZLE(1)
          WZTE(I+1) = WZTE(1)
          K=I+1

   300 CONTINUE


C  Determine the x,y,z locations of the 20 sections and the 10
C       points defining each section

       DO 310 I = 1,10
         DO 305 J = 2,19

           WYUP(J,I) = WYLE(J)
           WYLO(J,I) = WYLE(J)

           t =-(WYLE(J-1)-WYLE(1))/WYLE(20)+(WYLE(J)-WYLE(1))/WYLE(20)

           WXUP(J,I) = WXUP(J-1,I)+(WXUP(20,I)-WXUP(1,I))*t
           WZUP(J,I) = WZUP(J-1,I)+(WZUP(20,I)-WZUP(1,I))*t
           WXLO(J,I) = WXLO(J-1,I)+(WXLO(20,I)-WXLO(1,I))*t
           WZLO(J,I) = WZLO(J-1,I)+(WZLO(20,I)-WZLO(1,I))*t

   305   CONTINUE
   310 CONTINUE

C  Determine the chord lengths at the 20 sections

       DO 315 I = 1, 20
         CHORD(I) = WXTE(I) - WXLE(I)
   315 CONTINUE


C  Format the information for the x,y,z, and chord lengths of the wing
C       This is the first major input to BDAP after the control
C       switches. It is correct and validated for format and content


C  Format the array of cambered z-values as references to the
C       z-coordinate of the airfoil LE, ordered LE to TE
C       This is the next major input to BDAP after the info above.
C       It is correct and validated for format and content


       DO 340 I = 1,20
         DO 335 J = 1, 10
           ZORD(I,J) = WZUP(I,J) - WZLE(I)
   335   CONTINUE
   340 CONTINUE


C*********************************************************************
C  The following is a redefinition of the airfoils. I am going to fit
C       a 4-point cubic to the first 4 points on the airfoils (i.e.,
C       WXUP(I,1 to 4)) and two 3-point quadratics to WXUP(I,5 to 7)
C       and WXUP(I,8 to 10). From that I will determine the half
C       thickness at the following %chord locations
C       0.0     LE
C       0.5     4-point cubic
C       0.75    4-point cubic
C       2.5     4-point cubic
C       7.5     4-point cubic
C       20.0    First 3-point quadratic
C       30.0    First 3-point quadratic
C       50.0    First 3-point quadratic
C       70.0    Second 3-point quadratic
```

```
C       85.0    Second 3-point quadratic
C       95.0    Second 3-point quadratic
C       100.0   TE

C  Note: These curve fit coefficients were checked and validated for
C        the root chord


C  Do UPPER cubic fit

       DO 400 I = 1, 20

          Q1 = (WXUP(I,3)**3.)*(WXUP(I,2)-WXUP(I,1))
          Q1 = Q1 -(WXUP(I,2)**3.)*(WXUP(I,3)-WXUP(I,1))
          Q1 = Q1+(WXUP(I,1)**3.)*(WXUP(I,3)-WXUP(I,2))

          Q2 = (WXUP(I,4)**3.)*(WXUP(I,2)-WXUP(I,1))
          Q2 = Q2 - (WXUP(I,2)**3.)*(WXUP(I,4)-WXUP(I,1))
          Q2 = Q2 + (WXUP(I,1)**3.)*(WXUP(I,4)-WXUP(I,2))

          Q3 = WXUP(I,3)-WXUP(I,2)
          Q3 = Q3*(WXUP(I,2)-WXUP(I,1))
          Q3 = Q3*(WXUP(I,3)-WXUP(I,1))

          Q4 = WXUP(I,4)-WXUP(I,2)
          Q4 = Q4*(WXUP(I,2)-WXUP(I,1))
          Q4 = Q4*(WXUP(I,4)-WXUP(I,1))

          Q5 = WZUP(I,3)*(WXUP(I,2)-WXUP(I,1))
          Q5 = Q5-WZUP(I,2)*(WXUP(I,3)-WXUP(I,1))
          Q5 = Q5+WZUP(I,1)*(WXUP(I,3)-WXUP(I,2))

          Q6 = WZUP(I,4)*(WXUP(I,2)-WXUP(I,1))
          Q6 = Q6-WZUP(I,2)*(WXUP(I,4)-WXUP(I,1))
          Q6 = Q6+WZUP(I,1)*(WXUP(I,4)-WXUP(I,2))

          UCFA3(I) = (Q3*Q6-Q4*Q5)/(Q2*Q3-Q1*Q4)
          UCFA2(I) = (Q5-UCFA3(I)*Q1)/Q3
          UCFA1(I) = (WZUP(I,2)-WZUP(I,1))/(WXUP(I,2)-WXUP(I,1))
          TEMP = UCFA3(I)*(WXUP(I,2)**3.-WXUP(I,1)**3.)
          UCFA1(I) = UCFA1(I) -TEMP/(WXUP(I,2)-WXUP(I,1))
          UCFA1(I) = UCFA1(I)-UCFA2(I)*(WXUP(I,1)+WXUP(I,2))

          UCFA0(I) = WZUP(I,1)-UCFA1(I)*WXUP(I,1)
          UCFA0(I) = UCFA0(I)-UCFA2(I)*WXUP(I,1)**2.
          UCFA0(I) = UCFA0(I)-UCFA3(I)*WXUP(I,1)**3.

   400 CONTINUE

C  Do LOWER cubic fit

       DO 401 I = 1,20

          Q1 = (WXLO(I,3)**3.)*(WXLO(I,2)-WXLO(I,1))
          Q1 = Q1 -(WXLO(I,2)**3.)*(WXLO(I,3)-WXLO(I,1))
          Q1 = Q1+(WXLO(I,1)**3.)*(WXLO(I,3)-WXLO(I,2))

          Q2 = (WXLO(I,4)**3.)*(WXLO(I,2)-WXLO(I,1))
          Q2 = Q2 - (WXLO(I,2)**3.)*(WXLO(I,4)-WXLO(I,1))
          Q2 = Q2 + (WXLO(I,1)**3.)*(WXLO(I,4)-WXLO(I,2))

          Q3 = WXLO(I,3)-WXLO(I,2)
          Q3 = Q3*(WXLO(I,2)-WXLO(I,1))
          Q3 = Q3*(WXLO(I,3)-WXLO(I,1))

          Q4 = WXLO(I,4)-WXLO(I,2)
          Q4 = Q4*(WXLO(I,2)-WXLO(I,1))
```

```
            Q4 = Q4*(WXLO(I,4)-WXLO(I,1))

            Q5 = WZLO(I,3)*(WXLO(I,2)-WXLO(I,1))
            Q5 = Q5-WZLO(I,2)*(WXLO(I,3)-WXLO(I,1))
            Q5 = Q5+WZLO(I,1)*(WXLO(I,3)-WXLO(I,2))

            Q6 = WZLO(I,4)*(WXLO(I,2)-WXLO(I,1))
            Q6 = Q6-WZLO(I,2)*(WXLO(I,4)-WXLO(I,1))
            Q6 = Q6+WZLO(I,1)*(WXLO(I,4)-WXLO(I,2))

            LCFA3(I) = (Q3*Q6-Q4*Q5)/(Q2*Q3-Q1*Q4)
            LCFA2(I) = (Q5-LCFA3(I)*Q1)/Q3
            LCFA1(I) = (WZLO(I,2)-WZLO(I,1))/(WXLO(I,2)-WXLO(I,1))
            TEMP = LCFA3(I)*(WXLO(I,2)**3.-WXLO(I,1)**3.)
            LCFA1(I) = LCFA1(I) -TEMP/(WXLO(I,2)-WXLO(I,1))
            LCFA1(I) = LCFA1(I)-LCFA2(I)*(WXLO(I,1)+WXLO(I,2))

            LCFA0(I) = WZLO(I,1)-LCFA1(I)*WXLO(I,1)
            LCFA0(I) = LCFA0(I)-LCFA2(I)*WXLO(I,1)**2.
            LCFA0(I) = LCFA0(I)-LCFA3(I)*WXLO(I,1)**3.

    401 CONTINUE

C  Do first UPPER quadratic curve fit

        DO 402 I = 1, 20

        UQF1A2(I) =((WZUP(I,7)-WZUP(I,5))/(WXUP(I,7)-WXUP(I,5))-
     +             (WZUP(I,6)-WZUP(I,5))/(WXUP(I,6)-WXUP(I,5)))/
     +             (WXUP(I,7)-WXUP(I,6))
        UQF1A1(I) =(WZUP(I,6)-WZUP(I,5))/(WXUP(I,6)-WXUP(I,5))-
     +             UQF1A2(I)*(WXUP(I,5)+WXUP(I,6))
        UQF1A0(I) =WZUP(I,5)-UQF1A1(I)*WXUP(I,5)-UQF1A2(I)*WXUP(I,5)**2

C  Do first LOWER quadratic curve fit

        LQF1A2(I) =((WZLO(I,7)-WZLO(I,5))/(WXLO(I,7)-WXLO(I,5))-
     +             (WZLO(I,6)-WZLO(I,5))/(WXLO(I,6)-WXLO(I,5)))/
     +             (WXLO(I,7)-WXLO(I,6))
        LQF1A1(I) =(WZLO(I,6)-WZLO(I,5))/(WXLO(I,6)-WXLO(I,5))-
     +             LQF1A2(I)*(WXLO(I,5)+WXLO(I,6))
        LQF1A0(I)=WZLO(I,5)-LQF1A1(I)*WXLO(I,5)-LQF1A2(I)*WXLO(I,5)**2.

    402 CONTINUE
C********************************************************************

C  Do second UPPER quadratic curve fit

        DO 403 I = 1,20

        UQF2A2(I) =((WZUP(I,10)-WZUP(I,8))/(WXUP(I,10)-WXUP(I,8))-
     +             (WZUP(I,9)-WZUP(I,8))/(WXUP(I,9)-WXUP(I,8)))/
     +             (WXUP(I,10)-WXUP(I,9))
        UQF2A1(I) =(WZUP(I,9)-WZUP(I,8))/(WXUP(I,9)-WXUP(I,8))-
     +             UQF2A2(I)*(WXUP(I,8)+WXUP(I,9))
        UQF2A0(I)=WZUP(I,8)-UQF2A1(I)*WXUP(I,8)-UQF2A2(I)*WXUP(I,8)**2.

C  Do second LOWER quadratic curve fit

        LQF2A2(I) =((WZLO(I,10)-WZLO(I,8))/(WXLO(I,10)-WXLO(I,8))-
     +             (WZLO(I,9)-WZLO(I,8))/(WXLO(I,9)-WXLO(I,8)))/
     +             (WXLO(I,10)-WXLO(I,9))
        LQF2A1(I) =(WZLO(I,9)-WZLO(I,8))/(WXLO(I,9)-WXLO(I,8))-
     +             LQF2A2(I)*(WXLO(I,8)+WXLO(I,9))
        LQF2A0(I)=WZLO(I,8)-LQF2A1(I)*WXLO(I,8)-LQF2A2(I)*WXLO(I,8)**2.

    403 CONTINUE
```

```
C************************************************************************
C  Now that I have the coefficients, I need to establish the points
C       which will define the Z coordinates and half thicknesses
C       based on the above curve fit equations
C  First define the X-locations for the % chords stated above
C       These have been checked and validated

      DO 410 I = 1,20
         XCHORD(I,1) = 0.0 * CHORD(I)/100. + WXLE(I)
         XCHORD(I,2) = 0.5 * CHORD(I)/100. + WXLE(I)
         XCHORD(I,3) = 0.75 * CHORD(I)/100. + WXLE(I)
         XCHORD(I,4) = 2.5 * CHORD(I)/100. + WXLE(I)
         XCHORD(I,5) = 7.5 * CHORD(I)/100. + WXLE(I)
         XCHORD(I,6) = 20.0 * CHORD(I)/100. + WXLE(I)
         XCHORD(I,7) = 30.0 * CHORD(I)/100. + WXLE(I)
         XCHORD(I,8) = 50.0 * CHORD(I)/100. + WXLE(I)
         XCHORD(I,9) = 70.0 * CHORD(I)/100. + WXLE(I)
         XCHORD(I,10) = 85.0 * CHORD(I)/100. + WXLE(I)
         XCHORD(I,11) = 95.0 * CHORD(I)/100. + WXLE(I)
         XCHORD(I,12) = 100.0 * CHORD(I)/100. + WXLE(I)

  410 CONTINUE

C************************************************************************
C  Now find the corresponding Z-coordinate based on the above
C       X-locations and the corresponding curve fit equations

      DO 420 I = 1,20
         DO 415 J = 1,12
            IF(J.EQ.1) THEN
               ZORD(I,J) = 0.0
               THICK(I,J) = 0.0
            ELSEIF(J.EQ.12) THEN
               ZORD(I,J) = 0.0
               THICK(I,J) = 0.0
            ELSEIF(J.GT.1.AND.J.LT.6) THEN

               A = UCFA0(I)+UCFA1(I)*XCHORD(I,J)+UCFA2(I)*XCHORD(I,J)**2.
               A = A+UCFA3(I)*XCHORD(I,J)**3.

               B = LCFA0(I)+LCFA1(I)*XCHORD(I,J)+LCFA2(I)*XCHORD(I,J)**2.
               B = B +LCFA3(I)*XCHORD(I,J)**3.
               ZORD(I,J) = A - WZLE(I)
               THICK(I,J) = (A-B)/CHORD(I)

            ELSEIF(J.GT.5.AND.J.LT.9) THEN
               A = UQF1A0(I)+UQF1A1(I)*XCHORD(I,J)+UQF1A2(I)*
     +            XCHORD(I,J)**2.
               B = LQF1A0(I)+LQF1A1(I)*XCHORD(I,J)+LQF1A2(I)*
     +            XCHORD(I,J)**2.
               ZORD(I,J) = A - WZLE(I)
               THICK(I,J) = (A-B)/CHORD(I)
            ELSEIF(J.GT.8.AND.J.LT.12) THEN
               A = UQF2A0(I)+UQF2A1(I)*XCHORD(I,J)+UQF2A2(I)*
     +            XCHORD(I,J)**2.
               B = LQF2A0(I)+LQF2A1(I)*XCHORD(I,J)+LQF2A2(I)*
     +            XCHORD(I,J)**2.
               ZORD(I,J) = A - WZLE(I)
               THICK(I,J) = (A-B)/CHORD(I)
            ENDIF
  415    CONTINUE
  420 CONTINUE
```

```
C**********************************************************************
C  The next step is to redefine the engines. Thank God this looks like
C       it is going to be easy. First off, I need to read in the
C       geometry from the file engine.info created by the shell script.
C       I then need to sort out the actual location that I want and
C       reformat them to BDAP style.

      DO 505 I = 1, 16
        DO 500 J = 1,21
              READ(15,*) ENGX(I,J),ENGY(I,J),ENGZ(I,J)
500     CONTINUE
505 CONTINUE

      DO 515 I = 1, 16
        DO 510 J = 1,10
          ENGX(I,J) = 0.0
          ENGY(I,J) = 0.0
          ENGZ(I,J) = 0.0
510     CONTINUE
515 CONTINUE

      DO 525 I = 1, 16
        DO 520 J = 11,21
          IF(I.EQ.10) THEN
            ENGX(I,J-10) = 0.0
            ENGY(I,J-10) = 0.0
            ENGZ(I,J-10) = 0.0
          ELSE
            ENGX(I,J-10) = ENGX(I,J)
            ENGY(I,J-10) = ENGY(I,J)
            ENGZ(I,J-10) = ENGZ(I,J)
          ENDIF
520     CONTINUE
525 CONTINUE

        DO 526 I= 11,16
        DO 527 J = 1,11
            ENGX(I-1,J) = ENGX(I,J)
            ENGY(I-1,J) = ENGY(I,J)
            ENGZ(I-1,J) = ENGZ(I,J)
527 CONTINUE
526 CONTINUE

      DO 530 I = 1,15
        RADII(I) = 0.5*(ENGZ(I,11)-ENGZ(I,1))
530 CONTINUE


C  Test the bunching of output write statements for the
C       BDAP input file. Write it to bdap.info which will
C       be read by the shell script



      DO 535 I = 1,20
        WRITE(25,915) WXLE(I),WYLE(I),WZLE(I),CHORD(I)
535 CONTINUE
      DO 540 I = 1,20
        WRITE(25,920) (ZORD(I,J),J=1,12)
540 CONTINUE

      DO 545 I = 1,20
        WRITE(25,925) (THICK(I,J),J=1,12)
545 CONTINUE

C**********************************************************************
C  The next thing if to format the fuselage info to the correct
```

```
C        format. BDAP wants to see an x-location first and then the
C        y and z locations for each x-location

         WRITE(25,930) (FX(I,1),I=1,20)
         DO 550 I = 1,20
           WRITE(25,930) (FY(I,J), J=17,1,-1)
           WRITE(25,930) (FZ(I,J), J=17,1,-1)
  550 CONTINUE
         WRITE(25,950) ENGX(1,1), ENGY(1,1), -(WZLE(5)-ENGZ(1,6))
         WRITE(25,955) ((ENGX(I,1)-ENGX(1,1)),I=1,15)
         WRITE(25,955) (RADII(I),I=1,15)

         WRITE(25,935) VXLE(1),VYLE(1),VZLE(1),VXTE(1)-VXLE(1),
      +               VXLE(20),VYLE(20),VZLE(20),VXTE(20)-VXLE(20)

         WRITE(25,940) (VTCH(I),I= 1,10)
         WRITE(25,945) (100.0*0.5*VTHICK(I),I = 1,10)
         WRITE(25,960) HXLE(1),HYLE(1),HZLE(1),HXTE(1)-HXLE(1),
      +               HXLE(20),HYLE(20),HZLE(20),HXTE(20)-HXLE(20)
         WRITE(25,965) (HTCH(I),I= 1,10)
         WRITE(25,970) (100.0*0.5*HTHICK(I),I = 1,10)


C*********************************************************************
CAERO2SAERO2SAERO2SAERO2SAERO2SAERO2SAERO2SAERO2SAERO2SAERO2S

C  Now I need to take all of the info I have from generating the BDAP
C       input file to stuff that will be compatible with what AERO2S
C       requires. I will directly follow the AERO2S manual in my
C       conversion of the known geometry to that which is required

C*********************************************************************

C The first thing needed is TBLEY and TBLEX. I will assume that 20
C       spanwise station can define the entire geometry: 2 for the
C       fuselage and 18 for the wing.

         WRITE(20,975)

         IF(WYLE(20).LE.128.25) THEN
           UPPER=19
           LOWER = 4
         ELSE
           UPPER = 20
           LOWER = 3
         ENDIF

         TBLEY(1) = 0.0
         TBLEX(1) = 0.0
         TBLEY(2) = 13.5
         TBLEX(2) = WXLE(1)+(WXLE(2)-WXLE(1))*
      +             ((13.5-WYLE(1))/(WYLE(2)-WYLE(1)))
         TBLEY(3) = WYLE(20)
         TBLEX(3) = WXLE(20)

         WRITE(20,980) (TBLEY(I), I=1,3)
         WRITE(20,990) (TBLEX(I), I=1,3)
         WRITE(20,995)

         TBTEY(1) = 0.0
         TBTEX(1) = FX(20,1)
         TBTEY(2) = 13.5
         TBTEX(2) = WXTE(1)+(WXTE(2)-WXTE(1))*
      +             ((13.5-WYTE(1))/(WYTE(2)-WYTE(1)))
         TBTEY(3) = WYTE(20)
         TBTEX(3) = WXTE(20)
```

```
      WRITE(20,1000) (TBTEY(I), I=1,3)
      WRITE(20,1005) (TBTEX(I), I=1,3)

      XMAX = FX(20,1) - FX(1,1)
      IF(XMAX.GT.HXTE(1).AND.XMAX.GT.HXTE(20)) THEN
        WRITE(20,1010) XMAX
      ELSEIF(HXTE(1).GT.XMAX.AND.HXTE(1).GT.HXTE(20)) THEN
        WRITE(20,1010) HXTE(1)
      ELSEIF(HXTE(20).GT.XMAX.AND.HXTE(20).GT.HXTE(1)) THEN
        WRITE(20,1010) HXTE(20)
      ENDIF

      IF(WYLE(20).LT.128.25) THEN
        WRITE(20,1012)
      ELSE
        WRITE(20,1013)
      ENDIF

C  Now I need to define the spanwise sections. I am assumimg that
C       20 section will be sufficient definition

      TBYC(1) = 0.0
      TBYC(2) = TBLEY(2)

      DO 600 I = LOWER, 20
        TBYC(I) = WYLE(I)
  600 CONTINUE

      WRITE(20,1015) (TBYC(I), I = 1,UPPER)
      WRITE(20,1016)
      WRITE(20,1017)

      TZORDC(1,1) = FZ(1,1)
      TEMP2 = 0.005*(FX(20,1)-FX(1,1))
      TZORDC(1,2) = FZ(1,9)+(FZ(2,9)-FZ(1,9))*
     +             ((TEMP2-FX(1,9))/(FX(2,9)-FX(1,9)))
      TEMP2 = 0.0075*(FX(20,1)-FX(1,1))
      TZORDC(1,3) = FZ(1,9)+(FZ(2,9)-FZ(1,9))*
     +             ((TEMP2-FX(1,9))/(FX(2,9)-FX(1,9)))
      TEMP2 = 0.025*(FX(20,1)-FX(1,1))
      TZORDC(1,4) = FZ(1,9)+(FZ(2,9)-FZ(1,9))*
     +             ((TEMP2-FX(1,9))/(FX(2,9)-FX(1,9)))
      TEMP2 = 0.075*(FX(20,1)-FX(1,1))
      TZORDC(1,5) = FZ(7,9)+(FZ(8,9)-FZ(7,9))*
     +             ((TEMP2-FX(7,9))/(FX(8,9)-FX(7,9)))

      DO 605 I = 6,9
        TZORDC(1,I) = FZ(10,9)
  605 CONTINUE

C  These numbers are to estimate the camber of the aft end
C       of the fuselage

      TZORDC(1,10) = 0.2
      TZORDC(1,11) = 0.2
      TZORDC(1,12) = 0.2

      DO 610 I = 1,12
        TZORDC(12,I) = 0.0
  610 CONTINUE

      DO 620 I = 2,20
        DO 615 J = 1,10
          TZORDC(I,J+1) = 0.5*(WZUP(I,J)-WZLO(I,J))-WZLE(I)+WZLO(I,J)
  615   CONTINUE
  620 CONTINUE
```

```
         DO 625 I = 1,UPPER
            IF(I.EQ.1) THEN
               WRITE(20,1020) (TZORDC(I,J),J=1,12)
            ELSE
               WRITE(20,1025) (TZORDC(I,J),J=1,12)
            ENDIF
  625 CONTINUE

      WRITE(20,1030) WYLE(1),WYLE(20)

C  Note: The above write statements are all going to the file
C        'aero.info'. It is split here because the shell script has to
C        insert some information. I could have probably put this info
C        all together, but it was easier if I split it. Therefore,
C        all of the write statements below are going to 'aero.info2'

      WRITE(26,1032) MACH,RENUM
      WRITE(26,1035)
      WRITE(26,1040) HYLE(1),HYLE(20),HXLE(1),HXLE(20)
      WRITE(26,1042)
      WRITE(26,1045) HYTE(1),HYTE(20),HXTE(1),HXTE(20)
      WRITE(26,1050)
      WRITE(26,1055) HYLE(1), HYLE(20),HYLE(1), HYLE(20)


C**********************************************************************
C  FORMAT statements for output for BDAP input


  915 FORMAT(4F7.3)
  920 FORMAT(10F7.3,' ZORD',/,2F7.3)
  925 FORMAT(10F7.3,/,2F7.3)
  930 FORMAT(10F7.2)
  935 FORMAT(8F7.3)
  940 FORMAT(10F7.2,' XFIN1')
  945 FORMAT(10F7.2,' FORD1')
  950 FORMAT(3F7.3)
  955 FORMAT(10F7.3,/,5F7.3)
  960 FORMAT(8F7.3)
  965 FORMAT(10F7.2,' XFIN2')
  970 FORMAT(10F7.2,' FORD2')


C**********************************************************************
C  FORMAT statements for output for AERO2S input. Note: The info is
C        being written to 2 different files (aero.info and aero.info2)
C        so that the script can insert information between them.

  975 FORMAT('GLOBAL AERO ANALYSIS STUDY',/,' $INPT1',/,'NLEY=3,')
  980 FORMAT(' TBLEY(1)= ',F8.4,',',F8.4,',',F8.4,',')
  990 FORMAT(' TBLEX(1)= ',F8.4,',',F8.4,',',F8.4,',')
  995 FORMAT(' NTEY= 3,')
 1000 FORMAT(' TBTEY(1)=',F8.4,',',F8.4,',',F8.4,',')
 1005 FORMAT(' TBTEX(1)=',F8.4,',',F8.4,',',F8.4,',')
 1010 FORMAT('XMAX=',F8.4,',',/,/,'JBYMAX= 20,')
 1012 FORMAT(' NYC= 19,')
 1013 FORMAT(' NYC= 20,')
 1015 FORMAT(' TBYC=',F7.3,',',F7.3,',',F7.3,',',F7.3,',',
     + F7.3,',',F7.3,',',F7.3,',',F7.3,',',/,F7.3,',',F7.3,',',
     + F7.3,',',F7.3,',',F7.3,',',F7.3,',',F7.3,',',F7.3,',',/,
     + F7.3,',',F7.3,',',F7.3,',',F7.3,',')
 1016 FORMAT(' NPCTC=12,')
 1017 FORMAT(' TBPCTC=0.0,0.5,0.75,2.5,7.5,20.0,30.0,50.0,70.0,',
     +'85.0,95.0,100.0,')
 1020 FORMAT('TZORDC=',F6.3,',',F6.3,',',F6.3,',',F6.3,',',
     + F6.3,',',F6.3,',',F6.3,',',F6.3,',',F6.3,',',/,F6.3,',',
     + F6.3,',',F6.3,',14*0.0,')
```

```
1025 FORMAT(F6.3,',',F6.3,',',F6.3,',',F6.3,',',
     + F6.3,',',F6.3,',',F6.3,',',F6.3,',',F6.3,',',F6.3,',',/,
     + F6.3,',',F6.3,',14*0.0,')
1030 FORMAT(' NYR=2,',/,' TBYR=',F4.3,',',F7.3,',')
1032 FORMAT(' XM = ',F4.2,',',/,' RN = ',F6.2,',')
1035 FORMAT(' IVOROP=1,',/,' NALPHA=10,',/,
     + 'TALPHA= -5.,-4.,-2.,-1.,0.,1.,2.,4.,6.,',
     + '8.,',/,' ILS2= 2,',/,' NLEY2= 2,')
1040 FORMAT(' TBLEY2=',F7.3,',',F7.3,',',/,' TBLEX2=',F7.3,',',
     + F7.3,',')
1042 FORMAT(' NTEY2= 2,')
1045 FORMAT(' TBTEY2=',F7.3,',',F7.3,',',/,' TBTEX2=',F7.3,',',
     + F7.3,',')
1050 FORMAT(' NYC2= 2,')
1055 FORMAT(' TBYC2=',F4.3,',',F6.3,',',/,' NPCTC2= 2,',/,
     + ' TBPCTC2= 0.0,100.0,',/,' TZORDC2= 52*0.0,',/,' NYR2= 2,',/,
     + ' TBYR2= ',F4.3,',',F6.3,',',/,' DELTA2=2.0,')


C***********************************************************************

      END

C***********************************************************************

      SUBROUTINE REYNOLDS(Z,M,CBAR,REN)
C ************************************************************************
C  THis subroutine calculates the Reynolds number for the flight
C      condition specified by the Mach number, Altitude, and reference
C      length of the wing, i.e. Mean Aerodynamic Chord (MAC). These
C      subroutines were extracted from FLOPS and also Peter Rohl's
C      thesis work.

      IMPLICIT NONE
      REAL ZFT, DTC, DELTA, THETA, ASTAR, TM, RE, HFT, MACH, VEL
      REAL RHO0, RHO, A0, MAC, REN,Z,M,CBAR
C
C     standard sea level values, rho in slugs/ft^3 and a in ft/s
C     ----------------------------------------------------------
      DATA RHO0 /0.0023769/, A0 /1116.45/
C
C  10 CONTINUE

      ZFT = Z
      MACH = M
      MAC = CBAR
C
      CALL ATMO ( ZFT, DTC, DELTA, THETA, ASTAR, TM, RE, HFT )
C
      RHO = RHO0 * DELTA / THETA
      VEL = MACH * A0 * SQRT(THETA)
      REN = RE * MAC * MACH / 1.E+06

      RETURN
      END
C
      SUBROUTINE ATMO ( ZFT, DTC, DELTA, THETA, ASTAR, TM, RE, HFT )

C 1962 STANDARD ATMOSPHERIC PROPERTIES GOOD UP TO 88743 METERS
C      GEOPOTENTIAL ALTITUDE (90 KM GEOMETRIC ALTITUDE OR 291152 FEET)
C      ALSO SAME AS 1976 STD ATMOSPHERE TO 51 KM (167323 FEET)
C  INPUT/OUTPUT IN ENGLISH UNITS, CALCULATIONS IN SI UNITS
C  BASE PRESSURES AND EXPONENTS FOR EACH LAYER ARE RECOMPUTED TO ASSURE
C  CONTINUITY AT THE CORNERS REGARDLESS OF THE COMPUTER USED

C      ZFT        INPUT ALTITUDE - FEET
C      DTC        DELTA TEMPERATURE FROM STD - DEG C
```

```
C      DELTA      PRESSURE RATIO
C      THETA      TEMPERATURE RATIO
C      ASTAR      SPEED OF SOUND - KNOTS
C      TM         MOLECULAR-SCALE TEMPERATURE - DEG KELVIN
C      RE         REYNOLDS NUMBER PER FOOT AT MACH 1.
C      HFT        GEOPOTENTIAL ALTITUDE - FEET

       IMPLICIT REAL*8 (A-H, O-Z)
       REAL*4 ZFT, DTC, DELTA, THETA, ASTAR, TM, RE, HFT, SFT, STC
       DIMENSION P(9), E(9)
       SAVE P, E, DDLTA, DHETA, DSTAR, DTM, DRE, DFT, SFT, STC, IFIR
       DATA REARTH/6367533./,GR/9.80665/,GNS/9.823693/,CM1/.9985/,
      1     OC2/26.76566E-10/,IFIR/1/,SFT,STC/2*-1000./

C      PRECALCULATE EXPONENTS AND BASE PRESSURE RATIOS

       IF ( IFIR .NE. 1 ) GO TO 5
       P(1) = 1.
       GMOR = 9.80665 * 1.225 * 288.15 / 101.325
       E(1) = GMOR / 6.5
       P(2) = (216.65/288.15)**E(1)
       E(2) = -GMOR / 216.65
       P(3) = P(2) * EXP(E(2) * 9.)
       E(3) = GMOR
       P(4) = P(3) * (216.65/228.65)**GMOR
       E(4) = GMOR / 2.8
       P(5) = P(4) * (228.65/270.65)**E(4)
       E(5) = -GMOR / 270.65
       P(6) = P(5) * EXP(E(5) * 5.)
       E(6) = GMOR / 2.
       P(7) = P(6) * (252.65/270.65)**E(6)
       E(7) = GMOR / 4.
       P(8) = P(7) * (180.65/252.65)**E(7)
       E(8) = -GMOR / 180.65
       Z90  = 90000.
       R    = REARTH + Z90
       GN   = GNS * (REARTH / R)**(CM1 + 1.)
       H90  = (R * GN * ( (R/REARTH)**CM1 - 1.) / CM1
      1      - Z90 * (R - Z90/2.) * OC2) / GR
       DH   = H90/1000. - 79.
       P(9) = P(8) * EXP(E(8) * DH)
       E(9) = 11.056
       IFIR = 0

C CONVERT INPUT FEET TO METERS

     5 IF ( ZFT .EQ. SFT .AND. DTC .EQ. STC ) GO TO 110
       SFT  = ZFT
       STC  = DTC
       Z    = ZFT * .3048

C CALCULATE GEOPOTENTIAL ALTITUDE

       R    = REARTH + Z
       RPOW = (REARTH / R)**CM1
       GN   = GNS * (REARTH / R) * RPOW
       H    = (R * GN * ( 1./RPOW - 1.) / CM1
      1      - Z * (R - Z/2.) * OC2) / GR
       DFT  = H / .3048

C CONVERT H TO KILOMETERS

       H    = H/1000.

C SEA LEVEL TO 11 KM

       IF ( H .GT. 11. ) GO TO 11
```

```
      DTM   = 288.15 - 6.5 * H
      DDLTA = ((DTM)/288.15)**E(1)
      GO TO 100

C 11 KM TO 20 KM

   11 IF ( H .GT. 20. ) GO TO 20
      DH    = H - 11.
      DTM   = 216.65
      DDLTA = P(2) * EXP(E(2) * DH)
      GO TO 100

C 20 KM TO 32 KM

   20 IF ( H .GT. 32. ) GO TO 32
      DH    = H - 20.
      DTM   = 216.65 + DH
      DDLTA = P(3) * (216.65/DTM)**E(3)
      GO TO 100

C 32 KM TO 47 KM

   32 IF ( H .GT. 47. ) GO TO 47
      DH    = H - 32.
      DTM   = 228.65 + 2.8 * DH
      DDLTA = P(4) * (228.65/DTM)**E(4)
      GO TO 100

C 47 KM TO 52 KM

   47 IF ( H .GT. 52. ) GO TO 52
      DH    = H - 47.
      DTM   = 270.65
      DDLTA = P(5) * EXP(E(5) * DH)
      GO TO 100

C 52 KM TO 61 KM

   52 IF ( H .GT. 61. ) GO TO 61
      DH    = H - 52.
      DTM   = 270.65 - 2.0 * DH
      DDLTA = P(6) * (DTM/270.65)**E(6)
      GO TO 100

C 61 KM TO 79 KM

   61 IF ( H .GT. 79. ) GO TO 79
      DH    = H - 61.
      DTM   = 252.65 - 4.0 * DH
      DDLTA = P(7) * (DTM/252.65)**E(7)
      GO TO 100

C 79 KM TO 88743 METERS

   79 IF ( Z .GT. 90000. ) GO TO 90
      DH    = H - 79.
      DTM   = 180.65
      DDLTA = P(8) * EXP(E(8) * DH)
      GO TO 100

C ABOVE 88743 M, 1962 STD ATMOSPHERE SWITCHES TO GEOMETRIC ALTITUDE
C THE EQUATIONS BELOW ARE CLOSE UP TO 100 KM AND DIVERGE AFTER THAT

   90 DH    = Z/1000. - 90.
      DTM   = 180.65 + 3.0 * DH
      DDLTA = P(9) * (180.65/DTM)**E(9)
```

```
C CALCULATE TEMPERATURE RATIO, SPEED OF SOUND, AND REYNOLDS NUMBER

  100 DHETA = (DTM + DTC) / 288.15
      DSTAR = 661.479 * DSQRT(DHETA)
      DRE   = 1.479301E+9 * DDLTA * (DTM + 110.4) / DTM**2

  110 DELTA = DDLTA
      THETA = DHETA
      ASTAR = DSTAR
      TM    = DTM
      RE    = DRE
      HFT   = DFT
      RETURN
      END
```

Temporary Files

Most of the files contained in this section are used for passing information and are removed upon completion of the analysis. Only three files contained herein, remain after execution. Those files are "thick", "hthick", and "mission".

aero.info

This file is written by "convert" for use by "RUN" to generate the AERO2S input file.

```
GLOBAL AERO ANALYSIS STUDY
 $INPT1
NLEY=3,
 TBLEY(1)=     .0000, 13.5000,131.5200,
 TBLEX(1)=     .0000, 53.6050,114.8444,
 NTEY= 3,
 TBTEY(1)=    .0000, 13.5000,131.5200,
 TBTEX(1)=163.5000, 89.0420,124.2558,
XMAX=178.5628,

JBYMAX= 20,
 NYC= 20,
 TBYC=   .000, 13.500, 13.844, 20.766, 27.688, 34.611, 41.533, 48.455,
 55.377, 62.299, 69.221, 76.143, 83.065, 89.987, 96.909,103.832,
110.754,117.676,124.598,131.520,
 NPCTC=12,
 TBPCTC=0.0,0.5,0.75,2.5,7.5,20.0,30.0,50.0,70.0,85.0,95.0,100.0,
TZORDC= -.811, -.811, -.811, -.811, -.022,  .000,  .000,  .000,  .000,
  .200,  .200,  .200,14*0.0,
  .000,  .001,  .011,  .052,  .138,  .257,  .353,  .360,  .289,  .166,
  .048,  .000,14*0.0,
  .000,  .001,  .011,  .050,  .132,  .246,  .339,  .345,  .277,  .159,
  .046,  .000,14*0.0,
  .000,  .001,  .010,  .047,  .126,  .235,  .324,  .330,  .265,  .152,
  .044,  .000,14*0.0,
  .000,  .001,  .010,  .045,  .121,  .225,  .310,  .315,  .253,  .145,
```

```
 .042,   .000,14*0.0,
 .000,   .001,   .009,   .043,   .115,   .214,   .295,   .301,   .241,   .139,
 .040,   .000,14*0.0,
 .000,   .001,   .009,   .041,   .109,   .204,   .280,   .286,   .229,   .132,
 .038,   .000,14*0.0,
 .000,   .001,   .008,   .039,   .104,   .193,   .266,   .271,   .217,   .125,
 .036,   .000,14*0.0,
 .000,   .001,   .008,   .037,   .098,   .182,   .251,   .256,   .205,   .118,
 .034,   .000,14*0.0,
 .000,   .000,   .008,   .035,   .092,   .172,   .236,   .241,   .193,   .111,
 .032,   .000,14*0.0,
 .000,   .000,   .007,   .032,   .086,   .161,   .222,   .226,   .181,   .104,
 .030,   .000,14*0.0,
 .000,   .000,   .007,   .030,   .081,   .150,   .207,   .211,   .169,   .097,
 .028,   .000,14*0.0,
 .000,   .000,   .006,   .028,   .075,   .140,   .193,   .196,   .157,   .090,
 .026,   .000,14*0.0,
 .000,   .000,   .006,   .026,   .069,   .129,   .178,   .181,   .145,   .084,
 .024,   .000,14*0.0,
 .000,   .000,   .005,   .024,   .064,   .119,   .163,   .166,   .134,   .077,
 .022,   .000,14*0.0,
 .000,   .000,   .005,   .022,   .058,   .108,   .149,   .151,   .122,   .070,
 .020,   .000,14*0.0,
 .000,   .000,   .004,   .020,   .052,   .097,   .134,   .137,   .110,   .063,
 .018,   .000,14*0.0,
 .000,   .000,   .004,   .017,   .047,   .087,   .119,   .122,   .098,   .056,
 .016,   .000,14*0.0,
 .000,   .000,   .003,   .015,   .041,   .076,   .105,   .107,   .086,   .049,
 .014,   .000,14*0.0,
 .000,   .000,   .003,   .013,   .035,   .065,   .090,   .092,   .074,   .042,
 .012,   .000,14*0.0,
NYR=2,
TBYR=.000,131.520,
```

aero.info2

This file is written by "convert" for use by "RUN" to create the AERO2S input file. The information contained here is the latter half if the input file.

```
XM =   .30,
 RN =  59.65,
 IVOROP=1,
 NALPHA=10,
TALPHA= -5.,-4.,-2.,-1.,0.,1.,2.,4.,6.,8.,
 ILS2= 2,
 NLEY2= 2,
 TBLEY2=   .000, 34.560,
 TBLEX2=147.000,171.474,
 NTEY2= 2,
 TBTEY2=   .000, 34.560,
 TBTEX2=175.357,178.563,
 NYC2= 2,
 TBYC2=.000,34.560,
 NPCTC2= 2,
 TBPCTC2= 0.0,100.0,
 TZORDC2= 52*0.0,
```

```
NYR2= 2,
TBYR2= .000,34.560,
DELTA2=2.0,
```

bdap.info

This file is written by "convert" for use by "RUN" to create the BDAP input file.

```
 46.600    .000  9.650 38.414
 50.192   6.922  9.650 36.888
 53.784  13.844  9.650 35.361
 57.375  20.766  9.650 33.835
 60.967  27.688  9.650 32.308
 64.559  34.611  9.650 30.782
 68.151  41.533  9.650 29.255
 71.743  48.455  9.650 27.729
 75.334  55.377  9.650 26.202
 78.926  62.299  9.650 24.676
 82.518  69.221  9.650 23.149
 86.110  76.143  9.650 21.623
 89.702  83.065  9.650 20.097
 93.294  89.987  9.650 18.570
 96.885  96.909  9.650 17.044
100.477 103.832  9.650 15.517
104.069 110.754  9.650 13.991
107.661 117.676  9.650 12.464
111.253 124.598  9.650 10.938
114.844 131.520  9.650  9.411
    .000    .451   .589  1.039  1.201  2.299  2.463  2.241  1.584   .895 ZORD
    .347    .000
    .000    .433   .565   .998  1.153  2.207  2.366  2.152  1.521   .860 ZORD
    .334    .000
    .000    .415   .542   .956  1.106  2.116  2.268  2.063  1.458   .824 ZORD
    .320    .000
    .000    .398   .518   .915  1.058  2.024  2.170  1.973  1.395   .789 ZORD
    .306    .000
    .000    .380   .495   .874  1.010  1.933  2.072  1.884  1.332   .753 ZORD
    .292    .000
    .000    .362   .472   .833   .962  1.842  1.974  1.795  1.269   .718 ZORD
    .278    .000
    .000    .344   .448   .791   .915  1.750  1.876  1.706  1.206   .682 ZORD
    .265    .000
    .000    .326   .425   .750   .867  1.659  1.778  1.617  1.143   .646 ZORD
    .251    .000
    .000    .308   .402   .709   .819  1.568  1.680  1.528  1.080   .611 ZORD
    .237    .000
    .000    .290   .378   .667   .772  1.476  1.582  1.439  1.017   .575 ZORD
    .223    .000
    .000    .272   .355   .626   .724  1.385  1.485  1.350   .954   .540 ZORD
    .209    .000
    .000    .254   .331   .585   .676  1.294  1.387  1.261   .891   .504 ZORD
    .196    .000
    .000    .236   .308   .544   .628  1.202  1.289  1.172   .829   .468 ZORD
    .182    .000
    .000    .218   .285   .502   .581  1.111  1.191  1.083   .766   .433 ZORD
    .168    .000
    .000    .200   .261   .461   .533  1.020  1.093   .994   .703   .397 ZORD
    .154    .000
    .000    .182   .238   .420   .485   .928   .995   .905   .640   .362 ZORD
    .140    .000
```

```
 .000    .164    .214    .378    .437    .837    .897    .816    .577    .326 ZORD
 .126    .000
 .000    .146    .191    .337    .390    .746    .799    .727    .514    .291 ZORD
 .113    .000
 .000    .129    .168    .296    .342    .654    .701    .638    .451    .255 ZORD
 .099    .000
 .000    .111    .144    .255    .294    .563    .604    .549    .388    .219 ZORD
 .085    .000
 .000    .022    .028    .051    .057    .105    .110    .097    .067    .038
 .015    .000
 .000    .022    .028    .051    .057    .105    .110    .097    .067    .038
 .015    .000
 .000    .022    .028    .051    .057    .105    .110    .097    .067    .038
 .015    .000
 .000    .022    .028    .051    .057    .105    .110    .097    .067    .038
 .015    .000
 .000    .022    .028    .051    .057    .105    .110    .097    .067    .038
 .015    .000
 .000    .022    .028    .051    .057    .105    .110    .097    .067    .038
 .015    .000
 .000    .022    .028    .051    .057    .105    .110    .097    .067    .038
 .015    .000
 .000    .022    .028    .051    .057    .105    .110    .097    .067    .038
 .015    .000
 .000    .022    .028    .051    .057    .105    .110    .097    .067    .038
 .015    .000
 .000    .022    .028    .051    .057    .105    .110    .097    .067    .038
 .015    .000
 .000    .022    .028    .051    .057    .105    .110    .097    .067    .038
 .015    .000
 .000    .022    .028    .051    .057    .105    .110    .097    .067    .038
 .015    .000
 .000    .022    .028    .051    .057    .105    .110    .097    .067    .038
 .015    .000
 .000    .022    .028    .051    .057    .105    .110    .097    .067    .038
 .015    .000
 .000    .022    .028    .051    .057    .105    .110    .097    .067    .038
 .015    .000
 .000    .022    .028    .051    .057    .105    .110    .097    .067    .038
 .015    .000
 .000    .022    .028    .051    .057    .105    .110    .097    .067    .038
 .015    .000
 .000    .022    .028    .051    .057    .105    .110    .097    .067    .038
 .015    .000
 .000    .022    .028    .051    .057    .105    .110    .097    .067    .038
 .015    .000
 .00    1.66    4.44    6.53    8.16    9.80   11.15   12.52   14.88   17.13
102.44  104.89  143.16  153.35  159.53  161.59  162.40  163.00  163.37  163.50
 .00     .00     .00     .00     .00     .00     .00     .00     .00     .00
 .00     .00     .00     .00     .00     .00     .00
-.81    -.81    -.81    -.81    -.81    -.81    -.81    -.81    -.81    -.81
-.81    -.81    -.81    -.81    -.81    -.81    -.81
 .00     .59    1.15    1.67    2.12    2.50    2.77    2.94    3.00    2.94
2.77    2.50    2.12    1.67    1.15     .59     .00
-3.81   -3.75   -3.58   -3.31   -2.93   -2.48   -1.96   -1.40    -.81    -.22
 .34     .86    1.31    1.68    1.96    2.13    2.19
 .00     .98    1.92    2.78    3.54    4.16    4.62    4.90    5.00    4.90
4.62    4.16    3.54    2.78    1.92     .98     .00
-5.81   -5.72   -5.43   -4.97   -4.35   -3.59   -2.73   -1.79    -.81     .17
1.10    1.97    2.73    3.35    3.81    4.09    4.19
 .00    1.26    2.47    3.59    4.57    5.39    6.00    6.37    6.50    6.37
6.00    5.39    4.57    3.59    2.47    1.26     .00
-7.32   -7.21   -6.85   -6.27   -5.49   -4.53   -3.43   -2.23    -.97     .28
1.48    2.59    3.55    4.33    4.91    5.26    5.38
 .00    1.55    3.03    4.41    5.62    6.63    7.38    7.84    8.00    7.84
```

```
   7.38    6.63    5.62    4.41    3.03    1.55     .00
  -8.55   -8.41   -7.97   -7.26   -6.30   -5.12   -3.77   -2.29    -.75     .79
   2.27    3.62    4.80    5.76    6.47    6.91    7.05
    .00    1.78    3.50    5.08    6.46    7.61    8.45    8.98    9.15    8.98
   8.45    7.61    6.46    5.08    3.50    1.78     .00
  -9.42   -9.25   -8.74   -7.90   -6.77   -5.39   -3.82   -2.11    -.32    1.46
   3.17    4.74    6.12    7.25    8.09    8.60    8.78
    .00    1.97    3.87    5.63    7.18    8.48    9.45   10.05   10.25   10.05
   9.45    8.48    7.18    5.63    3.87    1.97     .00
  -9.99   -9.81   -9.27   -8.38   -7.17   -5.68   -3.96   -2.09    -.12    1.85
   3.72    5.44    6.93    8.14    9.03    9.57    9.76
    .00    2.14    4.21    6.13    7.84    9.27   10.35   11.02   11.25   11.02
  10.35    9.27    7.84    6.13    4.21    2.14     .00
 -10.62  -10.43   -9.86   -8.91   -7.62   -6.02   -4.17   -2.14     .00    2.14
   4.17    6.02    7.62    8.91    9.86   10.43   10.62
    .00    2.34    4.60    6.71    8.59   10.17   11.37   12.12   12.38   12.12
  11.37   10.17    8.59    6.71    4.60    2.34     .00
 -11.50  -11.30  -10.68   -9.67   -8.28   -6.56   -4.55   -2.33     .00    2.33
   4.55    6.56    8.28    9.67   10.68   11.30   11.50
    .00    2.50    4.93    7.21    9.27   11.02   12.36   13.21   13.50   13.21
  12.36   11.02    9.27    7.21    4.93    2.50     .00
 -12.00  -11.80  -11.18  -10.15   -8.73   -6.95   -4.84   -2.49     .00    2.49
   4.84    6.95    8.73   10.15   11.18   11.80   12.00
    .00    2.50    4.93    7.21    9.27   11.02   12.36   13.21   13.50   13.21
  12.36   11.02    9.27    7.21    4.93    2.50     .00
 -12.00  -11.80  -11.18  -10.15   -8.73   -6.95   -4.84   -2.49     .00    2.49
   4.84    6.95    8.73   10.15   11.18   11.80   12.00
    .00    2.48    4.89    7.16    9.22   10.98   12.34   13.20   13.50   13.20
  12.34   10.98    9.22    7.16    4.89    2.48     .00
 -11.75  -11.55  -10.96   -9.97   -8.59   -6.85   -4.78   -2.46     .00    2.46
   4.78    6.85    8.59    9.97   10.96   11.55   11.75
    .00    1.07    2.13    3.13    4.06    4.86    5.51    5.93    6.08    5.93
   5.51    4.86    4.06    3.13    2.13    1.07     .00
  -1.62   -1.54   -1.31    -.93    -.38     .32    1.18    2.16    3.23    4.29
   5.28    6.14    6.84    7.38    7.77    8.00    8.07
    .00     .73    1.45    2.14    2.79    3.36    3.83    4.14    4.25    4.14
   3.83    3.36    2.79    2.14    1.45     .73     .00
    .56     .60     .75     .99    1.33    1.79    2.35    3.01    3.73    4.45
   5.11    5.68    6.13    6.47    6.72    6.86    6.90
    .00     .51    1.02    1.51    1.98    2.39    2.74    2.98    3.06    2.98
   2.74    2.39    1.98    1.51    1.02     .51     .00
   1.77    1.80    1.89    2.05    2.28    2.58    2.96    3.41    3.92    4.42
   4.88    5.26    5.56    5.79    5.94    6.03    6.06
    .00     .38     .75    1.11    1.45    1.76    2.02    2.19    2.25    2.19
   2.02    1.76    1.45    1.11     .75     .38     .00
   2.46    2.48    2.55    2.67    2.83    3.05    3.33    3.67    4.04    4.41
   4.75    5.03    5.25    5.41    5.53    5.60    5.62
    .00     .28     .56     .83    1.09    1.32    1.51    1.64    1.69    1.64
   1.51    1.32    1.09     .83     .56     .28     .00
   2.94    2.96    3.01    3.10    3.22    3.39    3.60    3.85    4.13    4.40
   4.66    4.86    5.03    5.15    5.24    5.29    5.31
    .00     .18     .36     .53     .69     .84     .96    1.04    1.07    1.04
    .96     .84     .69     .53     .36     .18     .00
   3.47    3.48    3.51    3.57    3.64    3.75    3.88    4.04    4.22    4.40
   4.56    4.69    4.79    4.87    4.93    4.96    4.97
    .00     .08     .15     .23     .30     .36     .42     .45     .46     .45
    .42     .36     .30     .23     .15     .08     .00
   3.99    3.99    4.00    4.03    4.06    4.11    4.16    4.23    4.31    4.39
   4.46    4.51    4.56    4.59    4.62    4.63    4.64
    .00     .00     .00     .00     .00     .00     .00     .00     .00     .00
    .00     .00     .00     .00     .00     .00     .00
   4.38    4.38    4.38    4.38    4.38    4.38    4.38    4.38    4.38    4.38
   4.38    4.38    4.38    4.38    4.38    4.38    4.38
  36.622  27.480 -11.482
    .000    .107    .397    .963   1.895   3.286   5.228   7.811  11.128  16.570
  20.221  22.438  23.581  24.007  24.074
   6.578   6.875   7.046   7.120   7.121   7.076   7.011   6.953   6.928   6.927
```

```
  6.918   6.892   6.842   6.760   6.638
117.659   -.095   3.500  42.056149.234   -.254  35.671   9.060
   .00     .42    2.12    6.22   13.71   24.98   39.60   56.24   72.85  100.00 XFIN1
   .00    1.90    2.87    3.69    4.14    4.05    3.45    2.50    1.49     .00 FORD1
147.000   .000  35.577  28.357171.474  34.560  35.577   7.089
   .00    2.75    8.26   18.20   32.63   50.14   68.54   84.82   96.01  100.00 XFIN2
   .00    1.87    2.93    3.75    3.99    3.53    2.54    1.39     .45     .00 FORD2
```

engine.info

This file is generated by "RUN" for use by "convert". It is the engine geometry definition from the hermite file.

```
36.62229   27.48000    4.74635        36.72929   23.43655    3.73285
  36.62229   29.51478    4.42710       36.72929   25.35354    4.70912
  36.62229   31.34912    3.49293       36.72929   27.48000    5.04275
  36.62229   32.80493    2.03712       37.01972   27.48000    5.21447
  36.62229   33.73910    0.20278       37.01972   29.65957    4.87250
  36.62229   34.05835   -1.83200       37.01972   31.62444    3.87185
  36.62229   33.73910   -3.86678       37.01972   33.18385    2.31244
  36.62229   32.80493   -5.70112       37.01972   34.18450    0.34757
  36.62229   31.34912   -7.15693       37.01972   34.52647   -1.83200
  36.62229   29.51478   -8.09110       37.01972   34.18450   -4.01157
  36.62229   27.48000   -8.41035       37.01972   33.18385   -5.97644
  36.62229   25.44522   -8.09110       37.01972   31.62444   -7.53585
  36.62229   23.61088   -7.15693       37.01972   29.65957   -8.53650
  36.62229   22.15507   -5.70112       37.01972   27.48000   -8.87847
  36.62229   21.22090   -3.86678       37.01972   25.30043   -8.53650
  36.62229   20.90165   -1.83200       37.01972   23.33556   -7.53585
  36.62229   21.22090    0.20278       37.01972   21.77615   -5.97644
  36.62229   22.15507    2.03712       37.01972   20.77550   -4.01157
  36.62229   23.61088    3.49293       37.01972   20.43353   -1.83200
  36.62229   25.44522    4.42710       37.01972   20.77550    0.34757
  36.62229   27.48000    4.74635       37.01972   21.77615    2.31244
  36.72929   27.48000    5.04275       37.01972   23.33556    3.87185
  36.72929   29.60646    4.70912       37.01972   25.30043    4.87250
  36.72929   31.52345    3.73285       37.01972   27.48000    5.21447
  36.72929   33.04485    2.21145       37.58531   27.48000    5.28775
  36.72929   34.02112    0.29446       37.58531   29.68224    4.94222
  36.72929   34.35475   -1.83200       37.58531   31.66754    3.93117
  36.72929   34.02112   -3.95846       37.58531   33.24316    2.35554
  36.72929   33.04485   -5.87545       37.58531   34.25422    0.37024
  36.72929   31.52345   -7.39685       37.58531   34.59975   -1.83200
  36.72929   29.60646   -8.37312       37.58531   34.25422   -4.03424
  36.72929   27.48000   -8.70675       37.58531   33.24317   -6.01954
  36.72929   25.35354   -8.37312       37.58531   31.66754   -7.59517
  36.72929   23.43655   -7.39685       37.58531   29.68224   -8.60622
  36.72929   21.91515   -5.87545       37.58531   27.48000   -8.95175
  36.72929   20.93888   -3.95846       37.58531   25.27776   -8.60622
  36.72929   20.60525   -1.83200       37.58531   23.29246   -7.59517
  36.72929   20.93888    0.29446       37.58531   21.71683   -6.01954
  36.72929   21.91515    2.21145       37.58531   20.70578   -4.03424
                                       37.58531   20.36025   -1.83200
                                       37.58531   20.70578    0.37024
                                       37.58531   21.71683    2.35554
                                       37.58531   23.29246    3.93117
                                       37.58531   25.27776    4.94222
                                       37.58531   27.48000    5.28775
                                       38.51776   27.48000    5.28884
                                       38.51776   29.68257    4.94326
                                       38.51776   31.66819    3.93205
                                       38.51776   33.24405    2.35619
                                       38.51776   34.25526    0.37058
                                       38.51776   34.60084   -1.83200
                                       38.51776   34.25526   -4.03457
                                       38.51776   33.24405   -6.02019
                                       38.51776   31.66819   -7.59605
                                       38.51776   29.68258   -8.60726
                                       38.51776   27.48000   -8.95284
```

38.51776	25.27742	-8.60726	44.43347	20.52650	-1.83200
38.51776	23.29181	-7.59605	44.43347	20.86396	0.31881
38.51776	21.71595	-6.02019	44.43347	21.85140	2.25776
38.51776	20.70474	-4.03457	44.43347	23.39024	3.79660
38.51776	20.35916	-1.83200	44.43347	25.32919	4.78404
38.51776	20.70474	0.37058	44.43347	27.48000	5.12150
38.51776	21.71595	2.35619	47.75055	27.48000	5.09634
38.51776	23.29181	3.93205	47.75055	29.62303	4.76011
38.51776	25.27742	4.94326	47.75055	31.55497	3.77623
38.51776	27.48000	5.28884	47.75055	33.08823	2.24297
39.90879	27.48000	5.24400	47.75055	34.07211	0.31103
39.90879	29.66870	4.90060	47.75055	34.40834	-1.83200
39.90879	31.64181	3.89575	47.75055	34.07211	-3.97503
39.90879	33.20775	2.32981	47.75055	33.08823	-5.90697
39.90879	34.21260	0.35670	47.75055	31.55497	-7.44023
39.90879	34.55600	-1.83200	47.75055	29.62303	-8.42411
39.90879	34.21260	-4.02070	47.75055	27.48000	-8.76034
39.90879	33.20775	-5.99381	47.75055	25.33697	-8.42411
39.90879	31.64181	-7.55975	47.75055	23.40503	-7.44023
39.90879	29.66870	-8.56460	47.75055	21.87177	-5.90697
39.90879	27.48000	-8.90800	47.75055	20.88789	-3.97503
39.90879	25.29129	-8.56460	47.75055	20.55166	-1.83200
39.90879	23.31819	-7.55975	47.75055	20.88789	0.31103
39.90879	21.75225	-5.99381	47.75055	21.87177	2.24297
39.90879	20.74740	-4.02070	47.75055	23.40503	3.77623
39.90879	20.40400	-1.83200	47.75055	25.33697	4.76011
39.90879	20.74740	0.35670	47.75055	27.48000	5.09634
39.90879	21.75225	2.32981	47.75055	27.48000	5.09634
39.90879	23.31819	3.89575	47.75055	29.62303	4.76011
39.90879	25.29129	4.90060	47.75055	31.55497	3.77623
39.90879	27.48000	5.24400	47.75055	33.08823	2.24297
41.85013	27.48000	5.17947	47.75055	34.07211	0.31103
41.85013	29.64874	4.83920	47.75055	34.40834	-1.83200
41.85013	31.60386	3.84352	47.75055	34.07211	-3.97503
41.85013	33.15552	2.29186	47.75055	33.08823	-5.90697
41.85013	34.15120	0.33674	47.75055	31.55497	-7.44023
41.85013	34.49147	-1.83200	47.75055	29.62303	-8.42411
41.85013	34.15120	-4.00074	47.75055	27.48000	-8.76034
41.85013	33.15552	-5.95586	47.75055	25.33697	-8.42411
41.85013	31.60386	-7.50752	47.75055	23.40503	-7.44023
41.85013	29.64874	-8.50320	47.75055	21.87177	-5.90697
41.85013	27.48000	-8.84347	47.75055	20.88789	-3.97503
41.85013	25.31125	-8.50320	47.75055	20.55166	-1.83200
41.85013	23.35614	-7.50752	47.75055	20.88789	0.31103
41.85013	21.80448	-5.95586	47.75055	21.87177	2.24297
41.85013	20.80880	-4.00074	47.75055	23.40503	3.77623
41.85013	20.46853	-1.83200	47.75055	25.33697	4.76011
41.85013	20.80880	0.33674	47.75055	27.48000	5.09634
41.85013	21.80448	2.29186	53.19272	27.48000	5.09500
41.85013	23.35614	3.84352	53.19272	29.62262	4.75883
41.85013	25.31125	4.83920	53.19272	31.55418	3.77515
41.85013	27.48000	5.17947	53.19272	33.08715	2.24218
44.43347	27.48000	5.12150	53.19272	34.07083	0.31062
44.43347	29.63081	4.78404	53.19272	34.40700	-1.83200
44.43347	31.56976	3.79660	53.19272	34.07083	-3.97462
44.43347	33.10859	2.25776	53.19272	33.08715	-5.90618
44.43347	34.09604	0.31881	53.19272	31.55418	-7.43915
44.43347	34.43350	-1.83200	53.19272	29.62262	-8.42283
44.43347	34.09604	-3.98281	53.19272	27.48000	-8.75900
44.43347	33.10860	-5.92176	53.19272	25.33738	-8.42283
44.43347	31.56976	-7.46059	53.19272	23.40582	-7.43915
44.43347	29.63081	-8.44804	53.19272	21.87285	-5.90618
44.43347	27.48000	-8.78550	53.19272	20.88917	-3.97462
44.43347	25.32918	-8.44804	53.19272	20.55300	-1.83200
44.43347	23.39024	-7.46059	53.19272	20.88917	0.31062
44.43347	21.85140	-5.92176	53.19272	21.87285	2.24218
44.43347	20.86396	-3.98281	53.19272	23.40582	3.77515

53.19272	25.33738	4.75883
53.19272	27.48000	5.09500
56.84335	27.48000	5.08560
56.84335	29.61971	4.74989
56.84335	31.54865	3.76754
56.84335	33.07954	2.23665
56.84335	34.06189	0.30771
56.84335	34.39760	-1.83200
56.84335	34.06189	-3.97171
56.84335	33.07954	-5.90065
56.84335	31.54865	-7.43154
56.84335	29.61971	-8.41389
56.84335	27.48000	-8.74960
56.84335	25.34029	-8.41389
56.84335	23.41135	-7.43154
56.84335	21.88046	-5.90065
56.84335	20.89811	-3.97171
56.84335	20.56240	-1.83200
56.84335	20.89811	0.30771
56.84335	21.88046	2.23665
56.84335	23.41135	3.76754
56.84335	25.34029	4.74989
56.84335	27.48000	5.08560
59.06075	27.48000	5.06009
59.06075	29.61182	4.72561
59.06075	31.53364	3.74689
59.06075	33.05889	2.22165
59.06075	34.03761	0.29982
59.06075	34.37209	-1.83200
59.06075	34.03761	-3.96382
59.06075	33.05889	-5.88564
59.06075	31.53365	-7.41089
59.06075	29.61182	-8.38961
59.06075	27.48000	-8.72409
59.06075	25.34818	-8.38961
59.06075	23.42635	-7.41089
59.06075	21.90111	-5.88564
59.06075	20.92238	-3.96382
59.06075	20.58791	-1.83200
59.06075	20.92238	0.29982
59.06075	21.90111	2.22165
59.06075	23.42635	3.74689
59.06075	25.34818	4.72561
59.06075	27.48000	5.06009
60.20322	27.48000	5.01041
60.20322	29.59645	4.67834
60.20322	31.50442	3.70667
60.20322	33.01867	2.19243
60.20322	33.99034	0.28445
60.20322	34.32241	-1.83200
60.20322	33.99034	-3.94845
60.20322	33.01867	-5.85642
60.20322	31.50442	-7.37067
60.20322	29.59645	-8.34234
60.20322	27.48000	-8.67441
60.20322	25.36355	-8.34234
60.20322	23.45557	-7.37067
60.20322	21.94133	-5.85642
60.20322	20.96965	-3.94845
60.20322	20.63759	-1.83200
60.20322	20.96965	0.28445
60.20322	21.94133	2.19243
60.20322	23.45557	3.70667
60.20322	25.36355	4.67834
60.20322	27.48000	5.01041
60.62907	27.48000	4.92850
60.62907	29.57112	4.60041

60.62907	31.45625	3.64037
60.62907	32.95237	2.14425
60.62907	33.91241	0.25912
60.62907	34.24050	-1.83200
60.62907	33.91241	-3.92312
60.62907	32.95237	-5.80825
60.62907	31.45625	-7.30437
60.62907	29.57112	-8.26441
60.62907	27.48000	-8.59250
60.62907	25.38888	-8.26441
60.62907	23.50375	-7.30437
60.62907	22.00763	-5.80825
60.62907	21.04759	-3.92312
60.62907	20.71949	-1.83200
60.62907	21.04759	0.25912
60.62907	22.00763	2.14425
60.62907	23.50375	3.64037
60.62907	25.38888	4.60041
60.62907	27.48000	4.92850
60.69661	27.48000	4.80632
60.69661	29.53332	4.48416
60.69661	31.38439	3.54147
60.69661	32.85347	2.07239
60.69661	33.79616	0.22132
60.69661	34.11832	-1.83200
60.69661	33.79616	-3.88532
60.69661	32.85347	-5.73639
60.69661	31.38439	-7.20547
60.69661	29.53332	-8.14816
60.69661	27.48000	-8.47032
60.69661	25.42667	-8.14816
60.69661	23.57561	-7.20547
60.69661	22.10653	-5.73639
60.69661	21.16384	-3.88532
60.69661	20.84168	-1.83200
60.69661	21.16384	0.22132
60.69661	22.10653	2.07239
60.69661	23.57561	3.54147
60.69661	25.42668	4.48416
60.69661	27.48000	4.80632

fuse.info

This file is generated by "RUN" for use by "convert". It is the fuselage geometry definition from the hermite file. Note that only the positive values of the "y" coordinate are extracted from the hermite file.

0.00000	0.00000	-0.81116
0.00000	0.00000	-0.81116
0.00000	0.00000	-0.81116

0.00000	0.00000	-0.81116	4.43710	0.00000	-5.81116
0.00000	0.00000	-0.81116	4.43710	0.00000	4.18884
0.00000	0.00000	-0.81116	6.52820	0.00000	5.37661
0.00000	0.00000	-0.81116	6.52820	1.25676	5.25862
0.00000	0.00000	-0.81116	6.52820	2.46648	4.90552
0.00000	0.00000	-0.81116	6.52820	3.58528	4.32777
0.00000	0.00000	-0.81116	6.52820	4.57178	3.54543
0.00000	0.00000	-0.81116	6.52820	5.38764	2.58575
0.00000	0.00000	-0.81116	6.52820	5.99739	1.48431
0.00000	0.00000	-0.81116	6.52820	6.37359	0.28226
0.00000	0.00000	-0.81116	6.52820	6.50000	-0.97339
0.00000	0.00000	-0.81116	6.52820	6.37359	-2.22904
0.00000	0.00000	-0.81116	6.52820	5.99739	-3.43110
0.00000	0.00000	-0.81116	6.52820	5.38764	-4.53254
0.00000	0.00000	-0.81116	6.52820	4.57177	-5.49222
1.29095	0.00000	2.03600	6.52820	3.58528	-6.27456
1.29095	0.55667	1.98189	6.52820	2.46648	-6.85231
1.29095	1.09107	1.82038	6.52820	1.25675	-7.20541
1.29095	1.58307	1.55733	6.52820	0.00000	-7.32339
1.29095	2.01437	1.20321	6.52820	0.00000	5.37661
1.29095	2.36849	0.77191	8.16112	0.00000	7.05000
1.29095	2.63154	0.27991	8.16112	1.54533	6.90535
1.29095	2.79305	-0.25450	8.16112	3.03316	6.47234
1.29095	2.84716	-0.81116	8.16112	4.40967	5.76358
1.29095	2.79305	-1.36783	8.16112	5.62400	4.80338
1.29095	2.63154	-1.90224	8.16112	6.62888	3.62482
1.29095	2.36849	-2.39423	8.16112	7.38031	2.27145
1.29095	2.01437	-2.82553	8.16112	7.84412	0.79386
1.29095	1.58307	-3.17965	8.16112	8.00000	-0.75000
1.29095	1.09107	-3.44270	8.16112	7.84412	-2.29386
1.29095	0.55667	-3.60421	8.16112	7.38031	-3.77145
1.29095	0.00000	-3.65832	8.16112	6.62888	-5.12482
1.29095	0.00000	2.03600	8.16112	5.62400	-6.30338
1.65697	0.00000	2.18884	8.16112	4.40967	-7.26358
1.65697	0.58655	2.13182	8.16112	3.03316	-7.97234
1.65697	1.14964	1.96164	8.16112	1.54533	-8.40535
1.65697	1.66806	1.68447	8.16112	0.00000	-8.55000
1.65697	2.12250	1.31134	8.16112	0.00000	7.05000
1.65697	2.49564	0.85689	9.80099	0.00000	8.77554
1.65697	2.77280	0.33848	9.80099	1.78427	8.60351
1.65697	2.94298	-0.22461	9.80099	3.49829	8.08971
1.65697	3.00000	-0.81116	9.80099	5.07801	7.25198
1.65697	2.94298	-1.39771	9.80099	6.46472	6.12263
1.65697	2.77280	-1.96081	9.80099	7.60527	4.74483
1.65697	2.49564	-2.47922	9.80099	8.45367	3.17090
1.65697	2.12250	-2.93366	9.80099	8.97517	1.45943
1.65697	1.66805	-3.30680	9.80099	9.15000	-0.32447
1.65697	1.14964	-3.58396	9.80099	8.97517	-2.10836
1.65697	0.58655	-3.75414	9.80099	8.45367	-3.81983
1.65697	0.00000	-3.81116	9.80099	7.60527	-5.39376
1.65697	0.00000	2.18884	9.80099	6.46472	-6.77156
4.43710	0.00000	4.18884	9.80099	5.07801	-7.90091
4.43710	0.97758	4.09381	9.80099	3.49829	-8.73864
4.43710	1.91607	3.81018	9.80099	1.78427	-9.25244
4.43710	2.78009	3.34823	9.80099	0.00000	-9.42447
4.43710	3.53750	2.72634	9.80099	0.00000	8.77554
4.43710	4.15939	1.96893	11.14651	0.00000	9.75500
4.43710	4.62134	1.10491	11.14651	1.96881	9.57395
4.43710	4.90497	0.16642	11.14651	3.86684	9.03129
4.43710	5.00000	-0.81116	11.14651	5.62678	8.14102
4.43710	4.90497	-1.78874	11.14651	7.18394	6.93141
4.43710	4.62134	-2.72723	11.14651	8.47719	5.44142
4.43710	4.15939	-3.59125	11.14651	9.44746	3.72486
4.43710	3.53750	-4.34866	11.14651	10.04789	1.84605
4.43710	2.78009	-4.97055	11.14651	10.25000	-0.12000
4.43710	1.91607	-5.43250	11.14651	10.04789	-2.08605
4.43710	0.97758	-5.71613	11.14651	9.44746	-3.96486

11.14651	8.47719	-5.68142	21.77243	12.36251	4.83837
11.14651	7.18394	-7.17141	21.77243	13.21100	2.48781
11.14651	5.62678	-8.38102	21.77243	13.50000	0.00000
11.14651	3.86684	-9.27128	21.77243	13.21100	-2.48781
11.14651	1.96881	-9.81395	21.77243	12.36251	-4.83837
11.14651	0.00000	-9.99500	21.77243	11.01719	-6.94722
11.14651	0.00000	9.75500	21.77243	9.26832	-8.73427
12.51889	0.00000	10.62500	21.77243	7.21021	-10.15356
12.51889	2.14090	10.43384	21.77243	4.92730	-11.17921
12.51889	4.20912	9.85967	21.77243	2.49914	-11.79585
12.51889	6.13384	8.91429	21.77243	0.00000	-12.00000
12.51889	7.84497	7.62371	21.77243	0.00000	12.00000
12.51889	9.27451	6.02481	102.43995	0.00000	12.00000
12.51889	10.35321	4.17243	102.43995	2.49915	11.79585
12.51889	11.02370	2.13627	102.43995	4.92730	11.17921
12.51889	11.25000	0.00000	102.43995	7.21021	10.15356
12.51889	11.02370	-2.13628	102.43995	9.26832	8.73427
12.51889	10.35321	-4.17243	102.43995	11.01719	6.94722
12.51889	9.27451	-6.02481	102.43995	12.36251	4.83837
12.51889	7.84497	-7.62371	102.43995	13.21100	2.48781
12.51889	6.13384	-8.91429	102.43995	13.50000	0.00000
12.51889	4.20912	-9.85967	102.43995	13.21100	-2.48781
12.51889	2.14090	-10.43384	102.43995	12.36251	-4.83837
12.51889	0.00000	-10.62500	102.43995	11.01719	-6.94722
12.51889	0.00000	10.62500	102.43995	9.26832	-8.73427
14.87571	0.00000	11.50000	102.43995	7.21021	-10.15356
14.87571	2.33746	11.29621	102.43995	4.92730	-11.17921
14.87571	4.59922	10.68312	102.43995	2.49914	-11.79585
14.87571	6.71004	9.67078	102.43995	0.00000	-12.00000
14.87571	8.59383	8.28361	102.43995	0.00000	12.00000
14.87571	10.17499	6.55725	104.88999	0.00000	11.75000
14.87571	11.37393	4.54775	104.88999	2.47596	11.55385
14.87571	12.12195	2.33096	104.88999	4.88588	10.96033
14.87571	12.37500	0.00000	104.88999	7.15907	9.96999
14.87571	12.12195	-2.33096	104.88999	9.21789	8.59350
14.87571	11.37393	-4.54775	104.88999	10.97703	6.85146
14.87571	10.17499	-6.55725	104.88999	12.34015	4.78160
14.87571	8.59382	-8.28361	104.88999	13.20454	2.46260
14.87571	6.71004	-9.67078	104.88999	13.50000	0.00000
14.87571	4.59922	-10.68312	104.88999	13.20454	-2.46260
14.87571	2.33746	-11.29621	104.88999	12.34015	-4.78160
14.87571	0.00000	-11.50000	104.88999	10.97703	-6.85146
14.87571	0.00000	11.50000	104.88999	9.21789	-8.59350
17.13131	0.00000	12.00000	104.88999	7.15907	-9.96999
17.13131	2.49915	11.79585	104.88999	4.88588	-10.96033
17.13131	4.92730	11.17921	104.88999	2.47596	-11.55385
17.13131	7.21021	10.15356	104.88999	0.00000	-11.75000
17.13131	9.26832	8.73427	104.88999	0.00000	11.75000
17.13131	11.01719	6.94722	117.90611	0.00000	11.04786
17.13131	12.36251	4.83837	117.90611	1.98656	10.87580
17.13131	13.21100	2.48781	117.90611	3.90961	10.35794
17.13131	13.50000	0.00000	117.90611	5.70572	9.50219
17.13131	13.21100	-2.48781	117.90611	7.31032	8.32839
17.13131	12.36251	-4.83837	117.90611	8.65886	6.86579
17.13131	11.01719	-6.94722	117.90611	9.68283	5.16105
17.13131	9.26832	-8.73427	117.90611	10.32235	3.27855
17.13131	7.21021	-10.15356	117.90611	10.53884	1.29786
17.13131	4.92730	-11.17921	117.90611	10.32235	-0.68283
17.13131	2.49914	-11.79585	117.90611	9.68283	-2.56533
17.13131	0.00000	-12.00000	117.90611	8.65886	-4.27007
17.13131	0.00000	12.00000	117.90611	7.31032	-5.73267
21.77243	0.00000	12.00000	117.90611	5.70572	-6.90647
21.77243	2.49915	11.79585	117.90611	3.90961	-7.76222
21.77243	4.92730	11.17921	117.90611	1.98656	-8.28008
21.77243	7.21021	10.15356	117.90611	0.00000	-8.45214
21.77243	9.26832	8.73427	117.90611	0.00000	11.04786
21.77243	11.01719	6.94722	143.15634	0.00000	8.07112

143.15634	1.07368	7.99615	159.53050	1.02225	1.89453
143.15634	2.12546	7.76795	159.53050	0.51462	1.80278
143.15634	3.12976	7.38272	159.53050	0.00000	1.77283
143.15634	4.05573	6.83823	159.53050	0.00000	6.06391
143.15634	4.86469	6.13510	160.62208	0.00000	5.87656
143.15634	5.51052	5.27650	160.62208	0.45736	5.84994
143.15634	5.93123	4.29039	160.62208	0.90850	5.76840
143.15634	6.07745	3.22670	160.62208	1.34541	5.62900
143.15634	5.93123	2.16301	160.62208	1.75730	5.42800
143.15634	5.51052	1.17689	160.62208	2.12846	5.16125
143.15634	4.86469	0.31829	160.62208	2.43631	4.82479
143.15634	4.05573	-0.38484	160.62208	2.64782	4.41960
143.15634	3.12976	-0.92932	160.62208	2.72367	3.96977
143.15634	2.12546	-1.31456	160.62208	2.64782	3.51994
143.15634	1.07368	-1.54276	160.62208	2.43631	3.11475
143.15634	0.00000	-1.61772	160.62208	2.12846	2.77829
143.15634	0.00000	8.07112	160.62208	1.75730	2.51154
153.34991	0.00000	6.90440	160.62208	1.34541	2.31054
153.34991	0.73159	6.85786	160.62208	0.90850	2.17114
153.34991	1.45102	6.71571	160.62208	0.45736	2.08960
153.34991	2.14332	6.47403	160.62208	0.00000	2.06298
153.34991	2.78927	6.12873	160.62208	0.00000	5.87656
153.34991	3.36281	5.67631	161.59118	0.00000	5.61838
153.34991	3.83000	5.11414	161.59118	0.37844	5.59635
153.34991	4.14212	4.45365	161.59118	0.75174	5.52888
153.34991	4.25223	3.73134	161.59118	1.11326	5.41353
153.34991	4.14212	3.00904	161.59118	1.45408	5.24722
153.34991	3.83000	2.34855	161.59118	1.76119	5.02649
153.34991	3.36281	1.78637	161.59118	2.01592	4.74809
153.34991	2.78927	1.33395	161.59118	2.19094	4.41281
153.34991	2.14332	0.98865	161.59118	2.25370	4.04060
153.34991	1.45102	0.74698	161.59118	2.19094	3.66839
153.34991	0.73159	0.60482	161.59118	2.01592	3.33311
153.34991	0.00000	0.55829	161.59118	1.76119	3.05471
153.34991	0.00000	6.90440	161.59118	1.45408	2.83399
158.42068	0.00000	6.15429	161.59118	1.11326	2.66767
158.42068	0.54225	6.12273	161.59118	0.75174	2.55232
158.42068	1.07713	6.02606	161.59118	0.37844	2.48486
158.42068	1.59514	5.86078	161.59118	0.00000	2.46283
158.42068	2.08348	5.62247	161.59118	0.00000	5.61838
158.42068	2.52353	5.30620	162.39552	0.00000	5.30826
158.42068	2.88852	4.90730	162.39552	0.28365	5.29175
158.42068	3.13929	4.42690	162.39552	0.56344	5.24118
158.42068	3.22922	3.89357	162.39552	0.83442	5.15472
158.42068	3.13929	3.36025	162.39552	1.08986	5.03007
158.42068	2.88852	2.87985	162.39552	1.32006	4.86463
158.42068	2.52353	2.48095	162.39552	1.51098	4.65596
158.42068	2.08348	2.16468	162.39552	1.64216	4.40466
158.42068	1.59514	1.92637	162.39552	1.68920	4.12568
158.42068	1.07713	1.76109	162.39552	1.64216	3.84670
158.42068	0.54225	1.66442	162.39552	1.51098	3.59540
158.42068	0.00000	1.63286	162.39552	1.32006	3.38674
158.42068	0.00000	6.15429	162.39552	1.08986	3.22130
159.53050	0.00000	6.06391	162.39552	0.83442	3.09664
159.53050	0.51462	6.03396	162.39552	0.56344	3.01018
159.53050	1.02225	5.94221	162.39552	0.28365	2.95961
159.53050	1.51387	5.78535	162.39552	0.00000	2.94310
159.53050	1.97733	5.55919	162.39552	0.00000	5.30826
159.53050	2.39496	5.25903	162.99896	0.00000	4.96946
159.53050	2.74136	4.88045	162.99896	0.18009	4.95898
159.53050	2.97936	4.42452	162.99896	0.35774	4.92687
159.53050	3.06470	3.91837	162.99896	0.52978	4.87198
159.53050	2.97936	3.41222	162.99896	0.69197	4.79283
159.53050	2.74136	2.95629	162.99896	0.83812	4.68779
159.53050	2.39496	2.57771	162.99896	0.95934	4.55531
159.53050	1.97733	2.27755	162.99896	1.04263	4.39576
159.53050	1.51387	2.05139	162.99896	1.07249	4.21863

162.99896	1.04263	4.04150	171.05143	0.00000	36.19012
162.99896	0.95934	3.88195	166.43676	0.00000	36.69306
162.99896	0.83812	3.74947	161.21884	0.00000	37.09704
162.99896	0.69197	3.64443	156.25287	0.00000	37.27500
162.99896	0.52978	3.56528	152.16008	0.00000	37.12294
162.99896	0.35774	3.51039	149.34337	0.00000	36.68527
162.99896	0.18009	3.47828	147.78065	0.00000	36.21527
162.99896	0.00000	3.46780	147.14524	0.00000	35.85648
162.99896	0.00000	4.96946	147.00271	0.00000	35.64552
163.37317	0.00000	4.63519	147.00000	0.00000	35.57694
163.37317	0.07792	4.63065	147.02056	0.00000	35.51146
163.37317	0.15477	4.61676	147.21206	0.00000	35.34454
163.37317	0.22921	4.59301	147.90829	0.00000	35.15263
163.37317	0.29938	4.55877	149.50193	0.00000	35.02266
163.37317	0.36261	4.51332	152.26985	0.00000	34.99535
163.37317	0.41506	4.45600	156.24823	0.00000	35.01156
163.37317	0.45109	4.38697	161.18606	0.00000	35.09773
163.37317	0.46401	4.31034	166.39156	0.00000	35.25262
163.37317	0.45109	4.23370	171.01622	0.00000	35.40401
163.37317	0.41506	4.16467	174.21283	0.00000	35.51263
163.37317	0.36261	4.10736	175.35670	0.00000	35.57694
163.37317	0.29938	4.06191	178.56283	34.55973	35.57694
163.37317	0.22921	4.02767	178.28029	34.55973	35.62458
163.37317	0.15477	4.00392	177.48651	34.55973	35.73024
163.37317	0.07792	3.99003	176.33284	34.55973	35.85597
163.37317	0.00000	3.98549	175.02836	34.55973	35.95697
163.37317	0.00000	4.63519	173.78687	34.55973	36.00146
163.50000	0.00000	4.38027	172.76367	34.55973	35.96344
163.50000	0.00000	4.38027	172.05949	34.55973	35.85402
163.50000	0.00000	4.38027	171.66881	34.55973	35.73652
163.50000	0.00000	4.38027	171.50996	34.55973	35.64683
163.50000	0.00000	4.38027	171.47433	34.55973	35.59409
163.50000	0.00000	4.38027	171.47365	34.55973	35.57694
163.50000	0.00000	4.38027	171.47879	34.55973	35.56057
163.50000	0.00000	4.38027	171.52667	34.55973	35.51884
163.50000	0.00000	4.38027	171.70072	34.55973	35.47086
163.50000	0.00000	4.38027	172.09913	34.55973	35.43837
163.50000	0.00000	4.38027	172.79111	34.55973	35.43154
163.50000	0.00000	4.38027	173.78571	34.55973	35.43560
163.50000	0.00000	4.38027	175.02017	34.55973	35.45714
163.50000	0.00000	4.38027	176.32154	34.55973	35.49586
163.50000	0.00000	4.38027	177.47771	34.55973	35.53371
163.50000	0.00000	4.38027	178.27686	34.55973	35.56086
163.50000	0.00000	4.38027	178.56283	34.55973	35.57694
163.50000	0.00000	4.38027			

h_tail.info

This file is generated by "RUN" for use by "convert". It is the horizontal tail geometry definition from the hermite file.

175.35670	0.00000	35.57694
174.22655	0.00000	35.76748

v_tail.info

This file is generated by "RUN" for use by "convert". It is the vertical tail geometry definition from the hermite file.

159.71515	0.00000	3.50000
158.29393	-0.18626	3.50000
154.26788	-0.57065	3.50000

148.29596	-1.06657	3.50000
141.31044	-1.53317	3.50000
134.31272	-1.83535	3.50000
128.16423	-1.87452	3.50000
123.42474	-1.64303	3.50000
120.27650	-1.23133	3.50000
118.54947	-0.76894	3.50000
117.83603	-0.36566	3.50000
117.65947	-0.09506	3.50000
117.64725	0.00000	3.50000
117.65947	0.09506	3.50000
117.83603	0.36566	3.50000
118.54947	0.76894	3.50000
120.27650	1.23133	3.50000
123.42474	1.64303	3.50000
128.16423	1.87452	3.50000
134.31272	1.83535	3.50000
141.31044	1.53317	3.50000
148.29596	1.06657	3.50000
154.26788	0.57065	3.50000
158.29393	0.18626	3.50000
159.71515	0.00000	3.50000
162.81893	0.00000	35.67142
162.34993	-0.06146	35.67142
161.02133	-0.18831	35.67142
159.05060	-0.35197	35.67142
156.74538	-0.50595	35.67142
154.43613	-0.60567	35.67142
152.40713	-0.61859	35.67142
150.84310	-0.54220	35.67142
149.80418	-0.40634	35.67142
149.23426	-0.25375	35.67142
148.99882	-0.12067	35.67142
148.94056	-0.03137	35.67142
148.93653	0.00000	35.67142
148.94056	0.03137	35.67142
148.99882	0.12067	35.67142
149.23426	0.25375	35.67142
149.80418	0.40634	35.67142
150.84310	0.54220	35.67142
152.40713	0.61859	35.67142
154.43613	0.60567	35.67142
156.74538	0.50595	35.67142
159.05060	0.35197	35.67142
161.02133	0.18831	35.67142
162.34993	0.06146	35.67142
162.81893	0.00000	35.67142

85.01401	0.00000	9.65000
83.48122	0.00000	9.93801
79.17639	0.00000	10.55550
72.92105	0.00000	11.29293
65.84954	0.00000	11.88729
59.11044	0.00000	12.12603
53.61118	0.00000	11.90075
49.82186	0.00000	11.34772
47.69893	0.00000	10.69955
46.81894	0.00000	10.14199
46.60963	0.00000	9.77657
46.60000	0.00000	9.65000
46.62189	0.00000	9.52496
46.86508	0.00000	9.18144
47.78903	0.00000	8.70801
49.94193	0.00000	8.23910
53.71795	0.00000	7.93390
59.15246	0.00000	7.91017
65.82976	0.00000	8.16277
72.87856	0.00000	8.60887
79.13947	0.00000	9.09022
83.46624	0.00000	9.46287
85.01401	0.00000	9.65000
124.25581	131.51996	9.65000
123.88028	131.51996	9.72056
122.82559	131.51996	9.87185
121.29304	131.51996	10.05252
119.56051	131.51996	10.19814
117.90944	131.51996	10.25663
116.56212	131.51996	10.20143
115.63373	131.51996	10.06594
115.11362	131.51996	9.90714
114.89802	131.51996	9.77054
114.84674	131.51996	9.68101
114.84438	131.51996	9.65000
114.84974	131.51996	9.61937
114.90932	131.51996	9.53520
115.13569	131.51996	9.41921
115.66315	131.51996	9.30433
116.58827	131.51996	9.22956
117.91973	131.51996	9.22374
119.55567	131.51996	9.28563
121.28262	131.51996	9.39492
122.81655	131.51996	9.51285
123.87661	131.51996	9.60415
124.25581	131.51996	9.65000

wing.info

This file is generated by "RUN" for use by "convert". It is the wing geometry definition from the hermite file.

hthick

This file is created by the user from information obtained in RAM. The three numbers listed are the maximum thickness, location of maximum

thickness, and the leading edge radius of the horizontal tail airfoil. This information is needed by AERO2S and is inserted into the AERO2S input file via "RUN".

0.08 0.3 0.705

thick

This file is created by the user from information obtained in RAM. The three numbers listed are the maximum thickness, location of maximum thickness, and the leading edge radius of the wing airfoil. This information is needed by AERO2S and is inserted into the AERO2S input file via "RUN".

AERO2S and BDAP Input Files

AERO2S

This file is the input file for the AERO2S program. It is a product of both the shell script "RUN" and the geometry conversion program "convert".

0.11 0.3 1.333

mission

This file is also created by the user. The three numbers listed are the case number under consideration, the Mach number for analysis, and the altitude for analysis. This file is used by "RUN" and is inserted to both BDAP and AERO2S. It is also used by "convert" to determine the Reynolds number for the given flight condition.

1 0.3 1000.0

```
GLOBAL AERO ANALYSIS STUDY
 $INPT1
NLEY=3,
  TBLEY(1)=     .0000, 13.5000,131.5200,
  TBLEX(1)=     .0000, 53.6050,114.8444,
  NTEY= 3,
  TBTEY(1)=    .0000, 13.5000,131.5200,
  TBTEX(1)=163.5000, 89.0420,124.2558,
```

```
XMAX=178.5628,
 SREF = 2900.000,
 CBAR =  28.780,
 XMC =  40.822,
JBYMAX= 20,
 NYC= 20,
 TBYC=    .000, 13.500, 13.844, 20.766, 27.688, 34.611, 41.533, 48.455,
 55.377, 62.299, 69.221, 76.143, 83.065, 89.987, 96.909,103.832,
110.754,117.676,124.598,131.520,
 NPCTC=12,
 TBPCTC=0.0,0.5,0.75,2.5,7.5,20.0,30.0,50.0,70.0,85.0,95.0,100.0,
TZORDC= -.811, -.811, -.811, -.811, -.022,  .000,  .000,  .000,  .000,
   .200,  .200,  .200,14*0.0,
   .000,  .001,  .011,  .052,  .138,  .257,  .353,  .360,  .289,  .166,
   .048,  .000,14*0.0,
   .000,  .001,  .011,  .050,  .132,  .246,  .339,  .345,  .277,  .159,
   .046,  .000,14*0.0,
   .000,  .001,  .010,  .047,  .126,  .235,  .324,  .330,  .265,  .152,
   .044,  .000,14*0.0,
   .000,  .001,  .010,  .045,  .121,  .225,  .310,  .315,  .253,  .145,
   .042,  .000,14*0.0,
   .000,  .001,  .009,  .043,  .115,  .214,  .295,  .301,  .241,  .139,
   .040,  .000,14*0.0,
   .000,  .001,  .009,  .041,  .109,  .204,  .280,  .286,  .229,  .132,
   .038,  .000,14*0.0,
   .000,  .001,  .008,  .039,  .104,  .193,  .266,  .271,  .217,  .125,
   .036,  .000,14*0.0,
   .000,  .001,  .008,  .037,  .098,  .182,  .251,  .256,  .205,  .118,
   .034,  .000,14*0.0,
   .000,  .000,  .008,  .035,  .092,  .172,  .236,  .241,  .193,  .111,
   .032,  .000,14*0.0,
   .000,  .000,  .007,  .032,  .086,  .161,  .222,  .226,  .181,  .104,
   .030,  .000,14*0.0,
   .000,  .000,  .007,  .030,  .081,  .150,  .207,  .211,  .169,  .097,
   .028,  .000,14*0.0,
   .000,  .000,  .006,  .028,  .075,  .140,  .193,  .196,  .157,  .090,
   .026,  .000,14*0.0,
   .000,  .000,  .006,  .026,  .069,  .129,  .178,  .181,  .145,  .084,
   .024,  .000,14*0.0,
   .000,  .000,  .005,  .024,  .064,  .119,  .163,  .166,  .134,  .077,
   .022,  .000,14*0.0,
   .000,  .000,  .005,  .022,  .058,  .108,  .149,  .151,  .122,  .070,
   .020,  .000,14*0.0,
   .000,  .000,  .004,  .020,  .052,  .097,  .134,  .137,  .110,  .063,
   .018,  .000,14*0.0,
   .000,  .000,  .004,  .017,  .047,  .087,  .119,  .122,  .098,  .056,
   .016,  .000,14*0.0,
   .000,  .000,  .003,  .015,  .041,  .076,  .105,  .107,  .086,  .049,
   .014,  .000,14*0.0,
   .000,  .000,  .003,  .013,  .035,  .065,  .090,  .092,  .074,  .042,
   .012,  .000,14*0.0,
NYR=2,
TBYR=.000,131.520,
TBTOC = 0.090,0.090,
TBETA = 0.300,0.300,
TBROC = 0.893,0.893,
XM =  .30,
RN =  59.65,
IVOROP=1,
NALPHA=10,
TALPHA= -5.,-4.,-2.,-1.,0.,1.,2.,4.,6.,8.,
ILS2= 2,
NLEY2= 2,
TBLEY2=    .000, 34.560,
TBLEX2=147.000,171.474,
NTEY2= 2,
TBTEY2=    .000, 34.560,
```

```
TBTEX2=175.357,178.563,
NYC2= 2,
TBYC2=.000,34.560,
NPCTC2= 2,
TBPCTC2= 0.0,100.0,
TZORDC2= 52*0.0,
NYR2= 2,
TBYR2= .000,34.560,
DELTA2=2.0,
TBTOC2 = 0.080,0.080,
TBETA2 = 0.300,0.300,
TBROC2 = 0.705,0.705,
$END
```

BDAP

This file is the input file for the BDAP program. It is a product of both the shell script "RUN" and the geometry conversion program "convert".

```
GEOM NEW
GLOBAL AERO ANALYSIS STUDY
   1  1   1  1   1   1   0 20 12  1 17 20                    1 15   1  10 1 10  CONTROL
2900.0  28.78  40.82
     .00     .50     .75    2.50    7.50  20.00  30.00  50.00  70.00  85.00    XAF
  95.00 100.00                                                                 XAF
 46.600    .000  9.650 38.414
 50.192   6.922  9.650 36.888
 53.784  13.844  9.650 35.361
 57.375  20.766  9.650 33.835
 60.967  27.688  9.650 32.308
 64.559  34.611  9.650 30.782
 68.151  41.533  9.650 29.255
 71.743  48.455  9.650 27.729
 75.334  55.377  9.650 26.202
 78.926  62.299  9.650 24.676
 82.518  69.221  9.650 23.149
 86.110  76.143  9.650 21.623
 89.702  83.065  9.650 20.097
 93.294  89.987  9.650 18.570
 96.885  96.909  9.650 17.044
100.477 103.832  9.650 15.517
104.069 110.754  9.650 13.991
107.661 117.676  9.650 12.464
111.253 124.598  9.650 10.938
114.844 131.520  9.650  9.411
    .000    .451    .589   1.039   1.201   2.299   2.463   2.241   1.584    .895 ZORD
    .347    .000
    .000    .433    .565    .998   1.153   2.207   2.366   2.152   1.521    .860 ZORD
    .334    .000
    .000    .415    .542    .956   1.106   2.116   2.268   2.063   1.458    .824 ZORD
    .320    .000
    .000    .398    .518    .915   1.058   2.024   2.170   1.973   1.395    .789 ZORD
    .306    .000
    .000    .380    .495    .874   1.010   1.933   2.072   1.884   1.332    .753 ZORD
    .292    .000
    .000    .362    .472    .833    .962   1.842   1.974   1.795   1.269    .718 ZORD
```

```
.278   .000
.000   .344   .448   .791   .915  1.750  1.876  1.706  1.206   .682 ZORD
.265   .000
.000   .326   .425   .750   .867  1.659  1.778  1.617  1.143   .646 ZORD
.251   .000
.000   .308   .402   .709   .819  1.568  1.680  1.528  1.080   .611 ZORD
.237   .000
.000   .290   .378   .667   .772  1.476  1.582  1.439  1.017   .575 ZORD
.223   .000
.000   .272   .355   .626   .724  1.385  1.485  1.350   .954   .540 ZORD
.209   .000
.000   .254   .331   .585   .676  1.294  1.387  1.261   .891   .504 ZORD
.196   .000
.000   .236   .308   .544   .628  1.202  1.289  1.172   .829   .468 ZORD
.182   .000
.000   .218   .285   .502   .581  1.111  1.191  1.083   .766   .433 ZORD
.168   .000
.000   .200   .261   .461   .533  1.020  1.093   .994   .703   .397 ZORD
.154   .000
.000   .182   .238   .420   .485   .928   .995   .905   .640   .362 ZORD
.140   .000
.000   .164   .214   .378   .437   .837   .897   .816   .577   .326 ZORD
.126   .000
.000   .146   .191   .337   .390   .746   .799   .727   .514   .291 ZORD
.113   .000
.000   .129   .168   .296   .342   .654   .701   .638   .451   .255 ZORD
.099   .000
.000   .111   .144   .255   .294   .563   .604   .549   .388   .219 ZORD
.085   .000
.000   .022   .028   .051   .057   .105   .110   .097   .067   .038
.015   .000
.000   .022   .028   .051   .057   .105   .110   .097   .067   .038
.015   .000
.000   .022   .028   .051   .057   .105   .110   .097   .067   .038
.015   .000
.000   .022   .028   .051   .057   .105   .110   .097   .067   .038
.015   .000
.000   .022   .028   .051   .057   .105   .110   .097   .067   .038
.015   .000
.000   .022   .028   .051   .057   .105   .110   .097   .067   .038
.015   .000
.000   .022   .028   .051   .057   .105   .110   .097   .067   .038
.015   .000
.000   .022   .028   .051   .057   .105   .110   .097   .067   .038
.015   .000
.000   .022   .028   .051   .057   .105   .110   .097   .067   .038
.015   .000
.000   .022   .028   .051   .057   .105   .110   .097   .067   .038
.015   .000
.000   .022   .028   .051   .057   .105   .110   .097   .067   .038
.015   .000
.000   .022   .028   .051   .057   .105   .110   .097   .067   .038
.015   .000
.000   .022   .028   .051   .057   .105   .110   .097   .067   .038
.015   .000
.000   .022   .028   .051   .057   .105   .110   .097   .067   .038
.015   .000
.000   .022   .028   .051   .057   .105   .110   .097   .067   .038
.015   .000
.000   .022   .028   .051   .057   .105   .110   .097   .067   .038
.015   .000
.000   .022   .028   .051   .057   .105   .110   .097   .067   .038
.015   .000
```

```
  .000    .022    .028    .051    .057    .105    .110    .097    .067    .038
  .015    .000
   .00    1.66    4.44    6.53    8.16    9.80   11.15   12.52   14.88   17.13
102.44  104.89  143.16  153.35  159.53  161.59  162.40  163.00  163.37  163.50
   .00     .00     .00     .00     .00     .00     .00     .00     .00     .00
   .00     .00     .00     .00     .00     .00     .00
  -.81    -.81    -.81    -.81    -.81    -.81    -.81    -.81    -.81    -.81
  -.81    -.81    -.81    -.81    -.81    -.81    -.81
   .00     .59    1.15    1.67    2.12    2.50    2.77    2.94    3.00    2.94
  2.77    2.50    2.12    1.67    1.15     .59     .00
 -3.81   -3.75   -3.58   -3.31   -2.93   -2.48   -1.96   -1.40    -.81    -.22
   .34     .86    1.31    1.68    1.96    2.13    2.19
   .00     .98    1.92    2.78    3.54    4.16    4.62    4.90    5.00    4.90
  4.62    4.16    3.54    2.78    1.92     .98     .00
 -5.81   -5.72   -5.43   -4.97   -4.35   -3.59   -2.73   -1.79    -.81     .17
  1.10    1.97    2.73    3.35    3.81    4.09    4.19
   .00    1.26    2.47    3.59    4.57    5.39    6.00    6.37    6.50    6.37
  6.00    5.39    4.57    3.59    2.47    1.26     .00
 -7.32   -7.21   -6.85   -6.27   -5.49   -4.53   -3.43   -2.23    -.97     .28
  1.48    2.59    3.55    4.33    4.91    5.26    5.38
   .00    1.55    3.03    4.41    5.62    6.63    7.38    7.84    8.00    7.84
  7.38    6.63    5.62    4.41    3.03    1.55     .00
 -8.55   -8.41   -7.97   -7.26   -6.30   -5.12   -3.77   -2.29    -.75     .79
  2.27    3.62    4.80    5.76    6.47    6.91    7.05
   .00    1.78    3.50    5.08    6.46    7.61    8.45    8.98    9.15    8.98
  8.45    7.61    6.46    5.08    3.50    1.78     .00
 -9.42   -9.25   -8.74   -7.90   -6.77   -5.39   -3.82   -2.11    -.32    1.46
  3.17    4.74    6.12    7.25    8.09    8.60    8.78
   .00    1.97    3.87    5.63    7.18    8.48    9.45   10.05   10.25   10.05
  9.45    8.48    7.18    5.63    3.87    1.97     .00
 -9.99   -9.81   -9.27   -8.38   -7.17   -5.68   -3.96   -2.09    -.12    1.85
  3.72    5.44    6.93    8.14    9.03    9.57    9.76
   .00    2.14    4.21    6.13    7.84    9.27   10.35   11.02   11.25   11.02
 10.35    9.27    7.84    6.13    4.21    2.14     .00
-10.62  -10.43   -9.86   -8.91   -7.62   -6.02   -4.17   -2.14     .00    2.14
  4.17    6.02    7.62    8.91    9.86   10.43   10.62
   .00    2.34    4.60    6.71    8.59   10.17   11.37   12.12   12.38   12.12
 11.37   10.17    8.59    6.71    4.60    2.34     .00
-11.50  -11.30  -10.68   -9.67   -8.28   -6.56   -4.55   -2.33     .00    2.33
  4.55    6.56    8.28    9.67   10.68   11.30   11.50
   .00    2.50    4.93    7.21    9.27   11.02   12.36   13.21   13.50   13.21
 12.36   11.02    9.27    7.21    4.93    2.50     .00
-12.00  -11.80  -11.18  -10.15   -8.73   -6.95   -4.84   -2.49     .00    2.49
  4.84    6.95    8.73   10.15   11.18   11.80   12.00
   .00    2.50    4.93    7.21    9.27   11.02   12.36   13.21   13.50   13.21
 12.36   11.02    9.27    7.21    4.93    2.50     .00
-12.00  -11.80  -11.18  -10.15   -8.73   -6.95   -4.84   -2.49     .00    2.49
  4.84    6.95    8.73   10.15   11.18   11.80   12.00
   .00    2.48    4.89    7.16    9.22   10.98   12.34   13.20   13.50   13.20
 12.34   10.98    9.22    7.16    4.89    2.48     .00
-11.75  -11.55  -10.96   -9.97   -8.59   -6.85   -4.78   -2.46     .00    2.46
  4.78    6.85    8.59    9.97   10.96   11.55   11.75
   .00    1.07    2.13    3.13    4.06    4.86    5.51    5.93    6.08    5.93
  5.51    4.86    4.06    3.13    2.13    1.07     .00
 -1.62   -1.54   -1.31    -.93    -.38     .32    1.18    2.16    3.23    4.29
  5.28    6.14    6.84    7.38    7.77    8.00    8.07
   .00     .73    1.45    2.14    2.79    3.36    3.83    4.14    4.25    4.14
  3.83    3.36    2.79    2.14    1.45     .73     .00
   .56     .60     .75     .99    1.33    1.79    2.35    3.01    3.73    4.45
  5.11    5.68    6.13    6.47    6.72    6.86    6.90
   .00     .51    1.02    1.51    1.98    2.39    2.74    2.98    3.06    2.98
  2.74    2.39    1.98    1.51    1.02     .51     .00
  1.77    1.80    1.89    2.05    2.28    2.58    2.96    3.41    3.92    4.42
  4.88    5.26    5.56    5.79    5.94    6.03    6.06
   .00     .38     .75    1.11    1.45    1.76    2.02    2.19    2.25    2.19
  2.02    1.76    1.45    1.11     .75     .38     .00
  2.46    2.48    2.55    2.67    2.83    3.05    3.33    3.67    4.04    4.41
```

```
    4.75    5.03    5.25    5.41    5.53    5.60    5.62
     .00     .28     .56     .83    1.09    1.32    1.51    1.64    1.69    1.64
    1.51    1.32    1.09     .83     .56     .28     .00
    2.94    2.96    3.01    3.10    3.22    3.39    3.60    3.85    4.13    4.40
    4.66    4.86    5.03    5.15    5.24    5.29    5.31
     .00     .18     .36     .53     .69     .84     .96    1.04    1.07    1.04
     .96     .84     .69     .53     .36     .18     .00
    3.47    3.48    3.51    3.57    3.64    3.75    3.88    4.04    4.22    4.40
    4.56    4.69    4.79    4.87    4.93    4.96    4.97
     .00     .08     .15     .23     .30     .36     .42     .45     .46     .45
     .42     .36     .30     .23     .15     .08     .00
    3.99    3.99    4.00    4.03    4.06    4.11    4.16    4.23    4.31    4.39
    4.46    4.51    4.56    4.59    4.62    4.63    4.64
     .00     .00     .00     .00     .00     .00     .00     .00     .00     .00
     .00     .00     .00     .00     .00     .00     .00
    4.38    4.38    4.38    4.38    4.38    4.38    4.38    4.38    4.38    4.38
    4.38    4.38    4.38    4.38    4.38    4.38    4.38
  36.622  27.480-11.482
    .000    .107    .397    .963   1.895   3.286   5.228   7.811  11.128  16.570
  20.221  22.438  23.581  24.007  24.074
   6.578   6.875   7.046   7.120   7.121   7.076   7.011   6.953   6.928   6.927
   6.918   6.892   6.842   6.760   6.638
 117.659   -.095   3.500  42.056 149.234   -.254  35.671   9.060
     .00     .42    2.12    6.22   13.71   24.98   39.60   56.24   72.85  100.00 XFIN1
     .00    1.90    2.87    3.69    4.14    4.05    3.45    2.50    1.49     .00 FORD1
 147.000    .000  35.577  28.357 171.474  34.560  35.577   7.089
     .00    2.75    8.26   18.20   32.63   50.14   68.54   84.82   96.01  100.00 XFIN2
     .00    1.87    2.93    3.75    3.99    3.53    2.54    1.39     .45     .00 FORD2
SKFR
SKIN FOR GRA
1  1  1  1  1
1.
0.30    1.0    0.0    1.0
END
```

Parsed RAM Files

baseline.ram

This file is the file required to execute RAM. The geometry definition given is specific to the RAM program. The information needed form this file is the wing reference area and the mean aerodynamic chord length. Both of these values are at the end of the file.

```
RAM GEOMETRY FILE 1.05
5                              Number Of Components

//**************** FUSE COMPONENT ****************//
//==== General Parms ====//
1                              Type
          Fuselage             Name
0                              ID Number
          5915991              ID String
```

```
0                               Color
0                               Symmetry Code
    0.000    0.000    0.000     Translation
    0.000    0.000    0.000     Rotation
//==== Fuse Parms ====//
163.500000                      Fuse Length
0.000000                        Camber
0.500000                        Camber Location
0.000000                        Aft Offset
0.000000                        Nose Angle
0.300000                        Nose Strength
0.640000                        Nose Rho
0.640000                        Aft Rho
18                              Number of Xsecs
//==== Cross Section Number 0 ====//
0                               Fuse Xsec Type
-0.811161                       Z_Offset
0.000000                        Location On Spine
33                              Number of Pnts Per Xsec
//==== Cross Section Number 1 ====//
4                               Fuse Xsec Type
-0.811161                       Z_Offset
0.010000                        Location On Spine
33                              Number of Pnts Per Xsec
6.000000                        Height
6.000000                        Width
0.830000                        Top Tan Strength
0.830000                        Upper Tan Strength
0.830000                        Lower Tan Strength
0.830000                        Bottom Tan Strength
0.000000                        Max Width Location
0.000000                        Corner Radius
1.570796                        Top Tan Angle
1.570796                        Bot Tan Angle
//==== Cross Section Number 2 ====//
4                               Fuse Xsec Type
-0.811161                       Z_Offset
0.027109                        Location On Spine
33                              Number of Pnts Per Xsec
10.000000                       Height
10.000000                       Width
0.830000                        Top Tan Strength
0.830000                        Upper Tan Strength
0.830000                        Lower Tan Strength
0.830000                        Bottom Tan Strength
0.000000                        Max Width Location
0.000000                        Corner Radius
1.570796                        Top Tan Angle
1.570796                        Bot Tan Angle
//==== Cross Section Number 3 ====//
4                               Fuse Xsec Type
-0.973394                       Z_Offset
0.040000                        Location On Spine
33                              Number of Pnts Per Xsec
12.700000                       Height
13.000000                       Width
0.830000                        Top Tan Strength
0.830000                        Upper Tan Strength
0.830000                        Lower Tan Strength
0.830000                        Bottom Tan Strength
0.000000                        Max Width Location
0.000000                        Corner Radius
1.570796                        Top Tan Angle
1.570796                        Bot Tan Angle
//==== Cross Section Number 4 ====//
2                               Fuse Xsec Type
-0.750000                       Z_Offset
```

```
0.050000                         Location On Spine
33                               Number of Pnts Per Xsec
15.600000                        Height
16.000000                        Width
//==== Cross Section Number 5 ====//
2                                Fuse Xsec Type
-0.324465                        Z_Offset
0.060000                         Location On Spine
33                               Number of Pnts Per Xsec
18.200001                        Height
18.299999                        Width
//==== Cross Section Number 6 ====//
2                                Fuse Xsec Type
-0.120000                        Z_Offset
0.068438                         Location On Spine
33                               Number of Pnts Per Xsec
19.750000                        Height
20.500000                        Width
//==== Cross Section Number 7 ====//
2                                Fuse Xsec Type
0.000000                         Z_Offset
0.076875                         Location On Spine
33                               Number of Pnts Per Xsec
21.250000                        Height
22.500000                        Width
//==== Cross Section Number 8 ====//
2                                Fuse Xsec Type
0.000000                         Z_Offset
0.091094                         Location On Spine
33                               Number of Pnts Per Xsec
23.000000                        Height
24.750000                        Width
//==== Cross Section Number 9 ====//
2                                Fuse Xsec Type
0.000000                         Z_Offset
0.105313                         Location On Spine
33                               Number of Pnts Per Xsec
24.000000                        Height
27.000000                        Width
//==== Cross Section Number 10 ====//
2                                Fuse Xsec Type
0.000000                         Z_Offset
0.133750                         Location On Spine
33                               Number of Pnts Per Xsec
24.000000                        Height
27.000000                        Width
//==== Cross Section Number 11 ====//
2                                Fuse Xsec Type
0.000000                         Z_Offset
0.625000                         Location On Spine
33                               Number of Pnts Per Xsec
24.000000                        Height
27.000000                        Width
//==== Cross Section Number 12 ====//
2                                Fuse Xsec Type
0.000000                         Z_Offset
0.640000                         Location On Spine
33                               Number of Pnts Per Xsec
23.500000                        Height
27.000000                        Width
//==== Cross Section Number 13 ====//
2                                Fuse Xsec Type
1.297859                         Z_Offset
0.720000                         Location On Spine
33                               Number of Pnts Per Xsec
19.500000                        Height
21.077679                        Width
```

```
//==== Cross Section Number 14 ====//
2                               Fuse Xsec Type
3.226697                        Z_Offset
0.875000                        Location On Spine
33                              Number of Pnts Per Xsec
9.688839                        Height
12.154900                       Width
//==== Cross Section Number 15 ====//
2                               Fuse Xsec Type
3.731343                        Z_Offset
0.937500                        Location On Spine
33                              Number of Pnts Per Xsec
6.346115                        Height
8.504459                        Width
//==== Cross Section Number 16 ====//
2                               Fuse Xsec Type
3.893575                        Z_Offset
0.968750                        Location On Spine
33                              Number of Pnts Per Xsec
4.521437                        Height
6.458439                        Width
//==== Cross Section Number 17 ====//
0                               Fuse Xsec Type
4.380273                        Z_Offset
1.000000                        Location On Spine
33                              Number of Pnts Per Xsec


//**************** WING COMPONENT ****************//
//==== General Parms ====//
0                               Type
              Wing              Name
1                               ID Number
         6255751                ID String
17                              Color
0                               Symmetry Code
   46.600     0.000     9.650   Translation
    0.000     0.000     0.000   Rotation
//==== Wing Parms ====//
1                               Wing Driver Group
263.039917                      Span
11.000000                       Aspect Ratio
0.245000                        Taper Ratio
6290.000000                     Area
38.414009                       Root Chord
9.411432                        Tip Chord
0.463760                        Tan Sweep
0.250000                        Sweep Loc
0.000000                        Tan Dihedral
0.000000                        Twist Loc
0.000000                        Twist
3                               Flap Type
0.110507                        Flap Inboard Span
0.608696                        Flap Outboard Span
0.200000                        Flap Chord
2                               Slat Type
0.110507                        Slat Inboard Span
0.943841                        Slat Outboard Span
0.110507                        Slat Chord
0                               All Move CS
//==== Root Airfoil ====//
23                              Num of Airfoil Pnts
0.010000                        Airfoil Camber
0.410000                        Camber Loc
0.110000                        Thickness
//==== Tip Airfoil ====//
23                              Num of Airfoil Pnts
0.010000                        Airfoil Camber
```

```
0.410000                        Camber Loc
0.110000                        Thickness


//**************** WING COMPONENT ****************//
//==== General Parms ====//
0                               Type
              Vertical          Name
2                               ID Number
              6623343           ID String
4                               Color
0                               Symmetry Code
 117.647     0.000    3.500     Translation
  90.000     0.000    0.000     Rotation
//==== Wing Parms ====//
1                               Wing Driver Group
64.342834                       Span
2.300000                        Aspect Ratio
0.330000                        Taper Ratio
1800.000000                     Area
42.067890                       Root Chord
13.882404                       Tip Chord
0.753554                        Tan Sweep
0.250000                        Sweep Loc
0.000000                        Tan Dihedral
0.000000                        Twist Loc
0.000000                        Twist
1                               Flap Type
0.137681                        Flap Inboard Span
0.844203                        Flap Outboard Span
0.318841                        Flap Chord
0                               Slat Type
0.000000                        Slat Inboard Span
1.000000                        Slat Outboard Span
0.200000                        Slat Chord
0                               All Move CS
//==== Root Airfoil ====//
25                              Num of Airfoil Pnts
0.000000                        Airfoil Camber
0.000000                        Camber Loc
0.090000                        Thickness
//==== Tip Airfoil ====//
25                              Num of Airfoil Pnts
0.000000                        Airfoil Camber
0.000000                        Camber Loc
0.090000                        Thickness


//**************** WING COMPONENT ****************//
//==== General Parms ====//
0                               Type
              H_Tail            Name
3                               ID Number
              6855951           ID String
62                              Color
0                               Symmetry Code
 147.000     0.000   35.577     Translation
   0.000     0.000    0.000     Rotation
//==== Wing Parms ====//
1                               Wing Driver Group
69.119461                       Span
3.900000                        Aspect Ratio
0.250000                        Taper Ratio
1225.000000                     Area
28.356701                       Root Chord
7.089175                        Tip Chord
0.554309                        Tan Sweep
0.250000                        Sweep Loc
0.000000                        Tan Dihedral
```

```
0.000000                      Twist Loc
0.000000                      Twist
1                             Flap Type
0.047101                      Flap Inboard Span
0.925725                      Flap Outboard Span
0.300725                      Flap Chord
0                             Slat Type
0.000000                      Slat Inboard Span
1.000000                      Slat Outboard Span
0.200000                      Slat Chord
0                             All Move CS
//==== Root Airfoil ====//
23                            Num of Airfoil Pnts
0.020000                      Airfoil Camber
0.300000                      Camber Loc
0.080000                      Thickness
//==== Tip Airfoil ====//
23                            Num of Airfoil Pnts
0.020000                      Airfoil Camber
0.300000                      Camber Loc
0.080000                      Thickness


//**************** ENGINE COMPONENT ****************//
//==== General Parms ====//
2                             Type
            Engine_1          Name
4                             ID Number
            6975946           ID String
1                             Color
0                             Symmetry Code
   41.301    27.480    -1.832 Translation
    0.000     0.000     0.000 Rotation
//==== Engine Parms ====//
0                             Engine_Type
4.694000                      Radius_Tip
1.476000                      Max_Tip
0.325000                      Hub_Tip
4.122000                      Eng_Length
//==== Inlet Parms ====//
1                             Inlet_Type
0                             Inl_Noz_Color
0                             Inlet_XY_Sym_Flag
//==== Subsonic Pitot Inlet Parms ====//
0                             Inlet_Half_Split_Flag
1.000000                      Cowl_Length
1.000000                      Eng_Thrt_Ratio
1.706000                      Hilight_Thrt_Ratio
1.868494                      Lip_Finess_Ratio
1.000000                      Height_Width_Ratio
-2.000000                     Upper_Surf_Shape_Factor
-1.000000                     Lower_Surf_Shape_Factor
0.000000                      Inlet_X_Axis_Rot
0.000000                      Inlet_Scarf_Angle
//==== Inlet Duct Parms ====//
0                             Inlet_Duct_On/Off Flag
3.000000                      Inlet_Duct_X_Offset
1.000000                      Inlet_Duct_Y_Offset
0.500000                      Inlet_Duct_Shape_Factor
//==== Divertor Parms ====//
0                             Divertor On/Off Flag
0.500000                      Divertor Height
1.000000                      Divertor Length
//==== Nozzle Parms ====//
0                             Nozzle_Type
//==== Converg and Diverge Parms ====//
0.010000                      Nozzle Length
0.500000                      Exit_Area_Ratio
```

```
1.000000                        Nozzle_Height_Width_Ratio
1.500000                        Exit_Throat_Ratio
1.000000                        Dive_Flap_Ratio
//==== Nozzle Duct Parms ====//
0                               Nozzle_Duct_On/Off Flag
1.000000                        Nozzle_Duct_X_Offset
0.000000                        Nozzle_Duct_Y_Offset
0.000000                        Nozzle_Duct_Shape_Factor


//************* AERO PARMS *************//
1                Wing           Reference Component (ID #/Name)
2900.000000                     Reference Area
107.703293                      Reference Span
28.780294                       Reference Chord
   40.822    0.000    0.000     C.G. Location
-1                   None                Trimming Component (ID #/Name)
```