AFIT/GSO/ENY/99M-07

EAST COAST ABORT MODELING WITH RTLS ELIMINATION FOR THE SPACE SHUTTLE-LIQUID FLY BACK BOOSTER LAUNCH SYSTEM

THESIS

Thomas L. Miller, Jr., Captain, USAF

AFIT/GSO/ENY/99M-07

REPORT DOCUMENTATION PAGE			*Form Approved* *OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 1999	Master's Thesis

4. TITLE AND SUBTITLE
East Coast Abort Modeling with RTLS Elimination for the Space Shuttle-Liquid Fly Back Booster Launch System

5. FUNDING NUMBERS

6. AUTHOR(S)
Thomas L. Miller Jr., Captain, U.S. Air Force

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
2950 P Street
WPAFB, OH. 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GSO/ENY/99M-07

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Ira Victer
Program Director, Liquid Fly Back Booster (LFBB)
Boeing Defense Group P.O. Box 3644
Seal Beach, CA. 90740-7644
(562) 797-1428

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
William E. Wiesel, Professor Astronautical Engineering
Air Force Institute of Technology, WPAFB OH 45433-7765
(937) 255-6565 x4312 William.Wiesel@afit.af.mil

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE
A

13. ABSTRACT *(Maximum 200 words)*
This study investigates the ability of the proposed Liquid Fly Back Booster, a replacement for the Space Shuttle's Solid Rocket Boosters (SRBs), to eliminate the need for the Return to Launch Site (RTLS) abort mode. A Fortran model of a nominal launch trajectory was perturbed to simulate a single Space Shuttle Main Engine-out (SSME-out) abort scenario at different times in the launch with the goal being to abort to the southern most landing site possible. The model accounts for vehicle lift, drag, dynamic pressure, throttling times and includes atmospheric effects to enhance fidelity. Results show the RTLS abort mode can be eliminated and successful landings made as far south as Savannah, Georgia. This success is attributed to the throttling capability of the Liquid Fly Back Booster engines.

14. SUBJECT TERMS
Liquid Fly Back Booster, Return to Launch Site, Space Shuttle Main Engine, Solid Rocket Booster, Throttling Capability.

15. NUMBER OF PAGES
183

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

AFIT/GSO/ENY/99M-07

AFIT/GSO/ENY/99M-07

EAST COAST ABORT MODELING WITH RTLS ELIMINATION FOR THE SPACE SHUTTLE-LIQUID FLY BACK BOOSTER LAUNCH SYSTEM

THESIS

Presented to the faculty of the Graduate School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Space Operations

Thomas L. Miller, Jr., B.S.E.E.T.

Captain, USAF

March, 1999

AFIT/GSO/ENY/99M-07

EAST COAST ABORT MODELING WITH RTLS ELIMINATION FOR THE SPACE SHUTTLE-LIQUID FLY BACK BOOSTER LAUNCH SYSTEM

Thomas L. Miller, Jr., B.S.E.E.T.

Captain, USAF

Approved: William E. Wiesel — 8 Mar 99

William E. Wiesel, PhD
Thesis Advisor — Date

Stuart C. Kramer — 8MAR99

Stuart C. Kramer, Lt Col, USAF
Committee Member — Date

Jeffery K. Little — 8 Mar 99

Jeffery K. Little, Lt Col, USAF
Committee Member — Date

Acknowledgements

I would like to thank God for the gifts I have been given to complete a task such as this. To my wonderful wife Pattie, whose support and love have no end. I thank you and I am glad to have your support. I only hope that someday, I can repay all you have done for me. To all my family and friends, thanks for understanding and for all the support. For being in such close proximity to you all, I have never felt so far away as I have while completing this thesis. To Doctor William Wiesel, whose expertise and patience helped me attain that which we thought had slipped through our fingers, my sincere thanks. To Ira Victer, Tom Healy, and Mark Hines at Boeing Defense Group, Downey CA., thank you for all the help. I could not have done this without you. To Doug Whitehead, Dennis Bentley and the other people at NASA who helped in innumerable ways, my thanks. To General Starbuck and the members of the 45th Space Wing, thanks for giving me this chance to excel. To John Bond and Mark Farrar of Range Safety, and to the members of the 45th Range Squadron, thanks for all the information you provided about the Eastern Test Range, Launch, and Range Safety Operations. I hope to work with you again someday. To my friend Ric Sims, a fellow DAT and a guy who taught me that a wheelchair is almost as good as an M-1 Abrams...

Toujour's Pret, Always Ready.

Captain Thomas L. Miller Jr.

Table of Contents

List of Figures

List of Tables

Key Acronyms

AFIT	Air Force Institute of Technology
AOA	Abort Once Around
APU	Auxiliary Power Unit
ATO	Abort to Orbit
DTIC	Defense Technical Information Center
ECAL	East Coast Abort Landing
ET	External Tank
ISS	International Space Station
KSC	Kennedy Space Center
LFBB	Liquid Fly Back Booster
MAC	Mean Aerodynamic Chord
Max Q	Maximum Dynamic Pressure
MECO	Main Engine Cut Off
NASA	National Aeronautics & Space Administration
OMS	Orbital Maneuvering System
RCS	Reaction Control System
RTLS	Return to Launch Site
SLF	Shuttle Landing Facility
SRB (SRM)	Solid Rocket Booster (Motor)
SSME	Space Shuttle Main Engine
STS	Space Transportation System
TAEM	Terminal Area Energy Management
TAL	Transoceanic (Atlantic) Abort Landing

AFIT/GSO/ENY/99M-07

Abstract

This study investigated the ability of the proposed Liquid Fly Back Booster, a replacement for the Space Shuttle's Solid Rocket Booster (SRB), which is being developed by Boeing Defense Space Group, to eliminate the need for the Return to Launch Site (RTLS) abort mode. A Fortran model of a nominal launch trajectory was perturbed to simulate a *Single Space Shuttle Main Engine out* abort scenario, at different times during a high inclination (51.6°) launch. The model accounted for lift, drag, dynamic pressure, and variable throttle settings, and included atmospheric effects to enhance fidelity. Different control strategies were then applied with the goal of aborting to the southernmost possible landing site. Results show that RTLS can be eliminated, and successful landings made as far south as Savannah, Georgia. This unprecedented success is attributed to the throttling capability and enhanced performance of Boeing's Liquid Fly Back Booster.

EAST COAST ABORT MODELING WITH RTLS ELIMINATION FOR THE SPACE SHUTTLE-LIQUID FLY BACK BOOSTER LAUNCH SYSTEM

1. Introduction

1.1 Background

Since its initial flight in 1981, the United States has used the Space Shuttle as its only reusable manned space vehicle. The concept of a reusable space vehicle dates back to the pre-Apollo days. In the early 1960's, virtually every U.S. aerospace company was conducting studies of recoverable space boosters. Developmental efforts such as the Dynasoar and the X-series of rocket powered aircraft paved the way for the creation of the Space Transportation System known today as the Space Shuttle [17:274].

In the early 1960's when the concept of developing a reusable space system was in its infancy, the decision had already been made that the vehicle would be manned. Though the purpose of this thesis is not to debate the *man-in-the-loop* concept, it is beneficial to point out that having a crew onboard complicates matters. If an unexpected event should occur with a crew onboard, procedures other than just terminating the vehicle must be considered if the crew and the Shuttle are to be returned safely. What follows is an explanation of a typical launch mission profile and the procedures used in handling anomalous events.

1.2 Mission Profile

The Space Shuttle launch system is comprised of the orbiter and two solid rocket boosters (SRBs) that are attached to the external tank (ET) as shown in Figure 1-1.

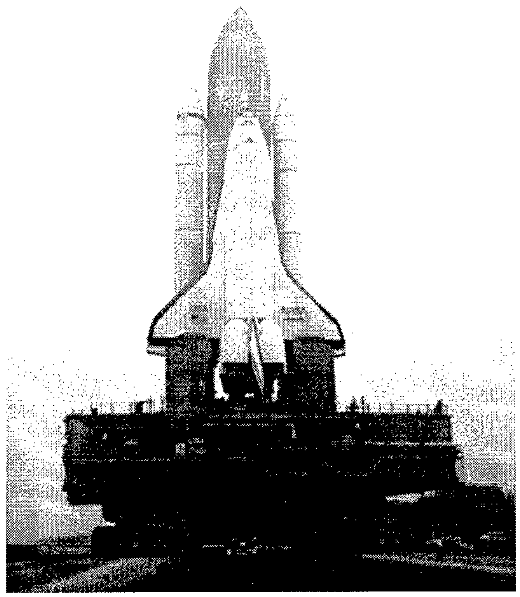

Figure 1-1. Author walking STS-76 to Pad A for the 1st Mir Mission.

While on the pad, the entire launch system is supported by the SRBs, these in turn are attached by eight explosive bolts to the launch platform. Prior to launch at "time to go" minus six seconds (T-6), the Space Shuttle's main engines (SSMEs) are ignited. Once the SSMEs reach the proper thrust levels, a signal is sent to ignite the SRBs and at the proper thrust-to-weight ratio the eight hold-down bolts are fired to release the Space Shuttle for liftoff [28:11]. Maximum dynamic pressure is reached approximately 60 seconds after liftoff. At launch plus 120 seconds (L+120) and an altitude of 50 km the SRBs have consumed their propellant and are jettisoned. The Shuttle now thrusts with just its three main engines. At approximately L+480 seconds the SSMEs are shut down and the external tank (ET) is jettisoned. The Shuttle then completes two thrusting maneuvers with its Orbital Maneuvering System (OMS) engines; the first, to insert the Shuttle into its earth orbit with an altitude ranging from 115 to 250 statute miles, and the second for circularizing the spacecraft's orbit.

During the ascent phase of a Shuttle mission, many opportunities exist where something can go wrong. Not until the Shuttle is in its final orbit do the astronauts breathe a sigh of relief. Selection of an ascent abort mode may become necessary if there is a failure that affects vehicle performance, such as the failure of a Space Shuttle main engine or an orbital maneuvering system. Other failures requiring early termination of a flight, such as a cabin leak or auxiliary power unit (APU) failure, might require the selection of an abort mode as well.

1.3 Abort Modes

There are two types of ascent abort modes for Space Shuttle missions: intact and contingency. Intact aborts are designed to safely land the Shuttle and its crew at some landing site and include: Abort to Orbit (ATO), Abort Once Around (AOA), Transoceanic Abort Landing (TAL), Return to Launch Site (RTLS), and East Coast Abort Landing (ECAL). Contingency aborts are designed so that the crew is returned safely while the Shuttle itself is sacrificed; these more severe types of abort scenarios occur because enough energy does not exist to execute one of the intact abort modes. In contingency mode, the abort usually leads to ditching in the ocean or use of the crew bailout system [28:14]. The intact abort modes will be discussed in this section since the research is focused on the possible elimination of the RTLS abort mode. Figure 1-2 shows the different stages that occur for each abort mode.

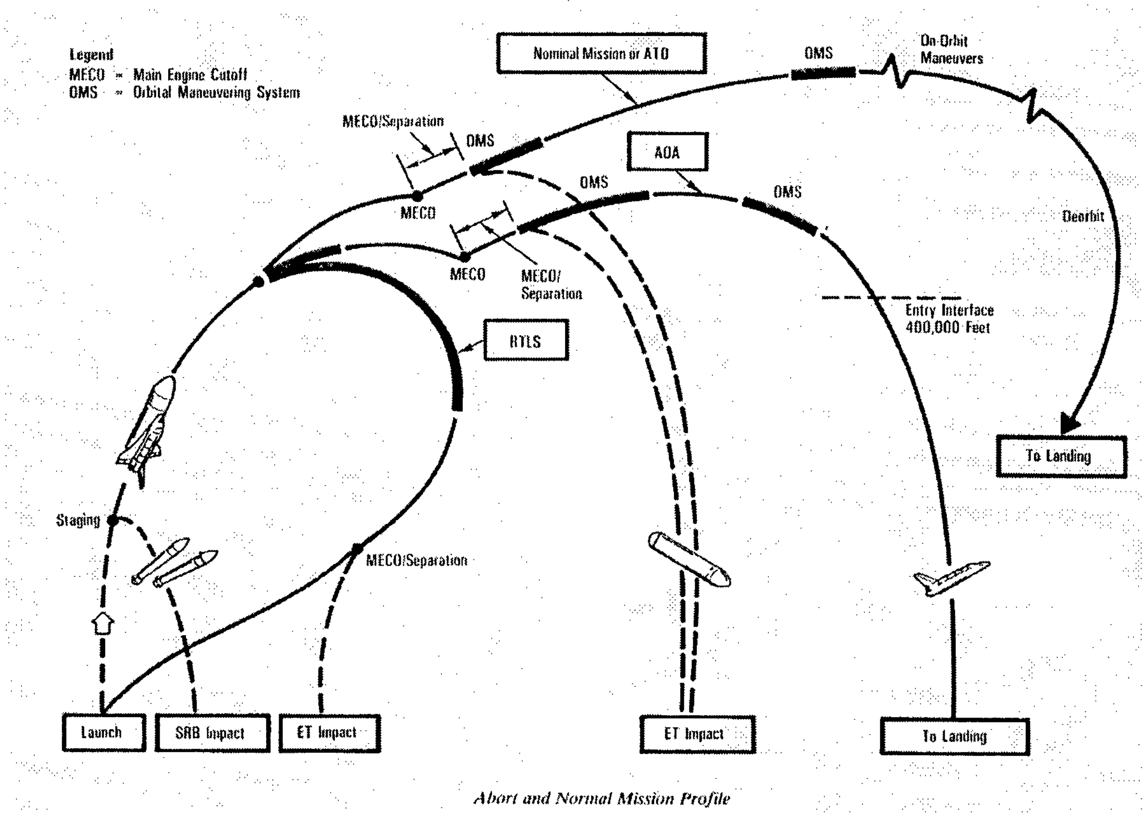

Figure 1-2. Shuttle Launch Profile and Abort Modes [28:10]

1.3.1 Abort to Orbit (ATO).

The ATO abort mode allows for the maximum time to evaluate problems before attempting an abort landing. Typically this mode is selected late in the launch window when enough performance capability exists to put the Shuttle and its crew into a temporary orbit. The decision can later be made to either deorbit or continue with the mission by using the OMS engines to raise the Shuttle to the proper orbit.

1.3.2 Abort Once Around (AOA).

The AOA abort mode is usually selected late in the launch window when performance capability is too low to make it to a temporary orbit. The Shuttle will circle the globe once and then attempt a normal entry and landing. After main engine cutoff (MECO) the OMS engines are commanded to fire twice, first to circularize the orbit and next to initiate re-entry. Landing occurs approximately 90 minutes later.

1.3.3 Transoceanic Abort Landing (TAL).

The TAL abort mode is selected from L+150 seconds until approximately L+560 seconds. At L+560 seconds the AOA and ATO abort modes are available [25]. Completing a TAL requires approximately 25-30 minutes [25], and as shown in Figure 1-3 this mode is ballistic in nature requiring no OMS use.

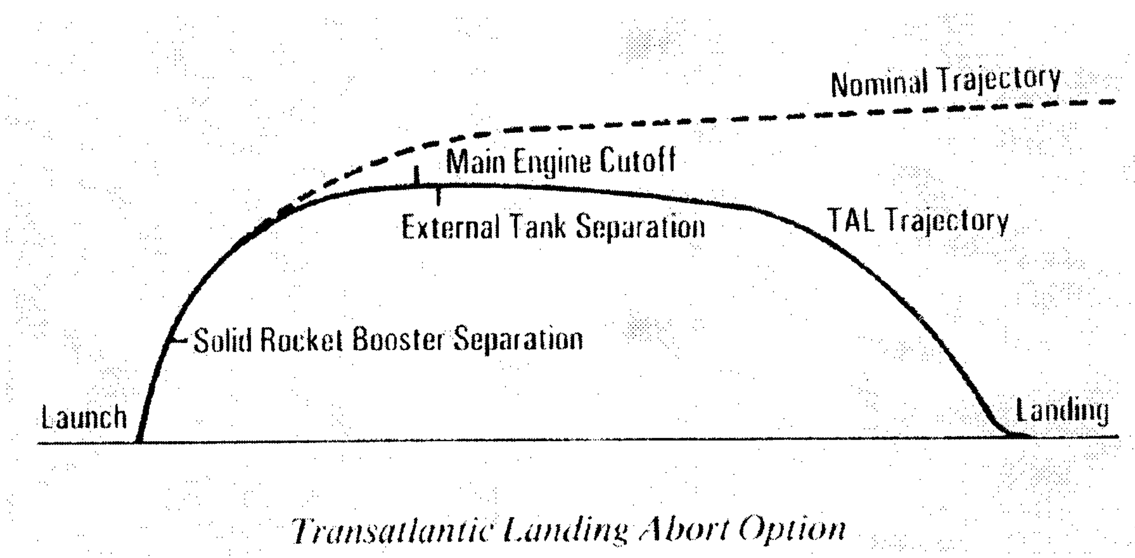

Figure 1-3. TAL profile [25]

Landing sites available are based on the inclination of launch and are listed in Table 1-1.

Table 1-1. TAL Sites and Inclinations When Used [25].

TAL Site	Inclination When Used	
Ben Guerir Air Base, Morocco	Low	28.50° - 42.75°
Yundum Airport, Banjul, The Gambia	Low	28.50° - 42.75°
Moron Air Base, Spain	Low/High	28.50° - 57.00°
Zaragoza Air Base, Spain	High	42.75° - 57.00°

1.3.4 Return to Launch Site (RTLS).

The RTLS abort mode is designed to accommodate the loss of thrust from one Space Shuttle main engine between liftoff and approximately four minutes 20 seconds. At this time not enough main propulsion system propellant remains to return to the launch site. The RTLS abort mode is initiated after SRB separation at L+120 seconds. This time overlaps with the TAL abort mode, which can be initiated as early as L+150 seconds. RTLS may be selected over TAL for various reasons, which could include weather conditions at the TAL site, or just the need to land as quickly as possible. With RTLS, landing can be accomplished as quickly as 25 minutes after launch [28:15].

The RTLS is performed in three phases: powered flight, ET separation, and glide-flight. During the power-flight portion of the RTLS, if the vehicle is not at the boundary of RTLS capability, the pitch attitude is changed to allow the vehicle to be lofted out of the atmosphere. This will be performed until the required amount of fuel in the ET has been depleted. The pitch-around maneuver is then executed (with approximately 10% ET propellant remaining) to begin the fly back phase for the vehicle. The vehicle is aligned so it is pointing towards the launch site. At this time, the vehicle is still moving away from the launch site, but the main engines are now thrusting to null the downrange velocity. It is important to realize that during this pitch-around maneuver when the main engines are attempting to null the downrange velocity, until sufficient velocity is built up the Shuttle will begin falling, much like a stone. It is this very problem which inspired the research for this thesis. Next, excess OMS and Reaction Control System (RCS) propellants are dumped via continuous thrusting to improve the weight and center of gravity of the Shuttle. When the desired altitude is reached, the vehicle pitches down to an attitude of approximately -4°. The SSMEs are throttled down to 65 percent and MECO is then performed. Shortly after MECO, the ET is separated from the orbiter. After ET separation, the vehicle pitches back up and acquires the glide path for the RTLS runway. At this point the procedures pick up as if it were a nominal entry.

1.3.5 East Coast Abort Landing (ECAL)

This abort mode has been shelved since the addition of Zaragoza Air Base, which is located in Spain, as a TAL site for high inclination launches. ECAL is included because it is the quickest method for getting the Shuttle and its crew back on the ground (10 to 15 minutes in some cases), and because NASA is currently reviewing its potential for future high inclination launches to the International Space Station. Prior to instituting Zaragoza as a TAL site, NASA became concerned about potential abort gaps between RTLS and TAL that could exist due to weight or inclination restrictions of some launches. In 1982, Dennis Bentley [2], the Shuttle Abort Chairman at the time, worked on abort procedures that would land the Shuttle along the East Coast. This new abort mode, East Coast Abort Landing (ECAL) would cover the potential abort gaps between the RTLS and TAL [2]. Table 1-2 list the landing sites instituted by Mr. Bentley's plan. They are listed from the southernmost at Myrtle Beach, S.C., to the northernmost at Gander, Newfoundland.

Table 1-2. 1982 ECAL Landing Sites [7]

Landing Facility	Runway Length	Runway Approach Heading
Myrtle Beach S.C.	9,503 ft	170°/350°
Cherry Point N.C.	8,980 ft	140°/320°
Oceana NAS VA.	11,997 ft	50°/230°
Dover AFB DE.	12,902 ft	140°/320°
Otis ANGB MA.	9,500 ft	140°/320°
Pease AFB NH.	11,318 ft	160°/340°
Halifax Nova Scotia	8,800 ft	60°/240°
St. Johns	8,500 ft	NA
Gander Newfoundland	10,500 ft	40°/220°

Again, with the inclusion of Zaragoza as a TAL site for northerly launches, this abort mode was discontinued in the early 1990's. RTLS would cover the initial launch up to and including the overlap point at L+150 seconds with TAL. With this in mind, the quickest the crew could get back on terra firma was approximately 25 minutes by executing the RTLS abort mode.

1.4 Current Abort Mode Issues

A definite order exists as to which abort mode is selected based on when in the launch window the anomalous event occurs and what type of failure occurs. In cases where degradation of performance is the only factor, the order of preference would be ATO, AOA, TAL and RTLS. If the anomaly affects support systems such as a cabin leak, a cracked wind shield, an APU failure, or cooling problems, then the desired order of preference would be the one that ends the mission the quickest. In these cases, TAL or RTLS might be preferable to AOA or ATO since the orbiter can be on the ground in as little as 25 minutes.

In any event, one of the most important issues is time. As mentioned in the previous RTLS section, if an abort situation requires the use of RTLS shortly after launch, nothing could be accomplished until the SRBs had stopped thrusting and had separated. Separation occurs at L+120 seconds. By this time the Shuttle has attained a high velocity and considerable altitude. Even if time were not the primary issue after beginning the RTLS abort, loosing altitude soon would be. Just after the pitch-around maneuver is completed, the orbiter is pointed at the launch site but its momentum continues to carry it downrange. At some point downrange velocity goes to zero and the orbiter and ET begin to fall. The concern now becomes, can the Shuttle increase its forward momentum towards the landing site enough so that it is not forced to ditch in the ocean? This thought leads to the motivation behind this topic as will be discussed in the next section.

1.5 Research Motivation

Motivation for this research began during launch preparations for the STS-76 Shuttle mission in 1996. This mission had an inclination of 51.6° and would be the maiden flight to the Mir Space Station. During the launch rehearsal the Shuttle crew was suited up and running abort scenarios, after a few unsuccessful Return to Launch Site (RTLS) attempts one of the astronauts commented that the RTLS abort mode would never work and would always result in ditching in the ocean. After investigating, the general consensus revealed the astronaut community considered the RTLS procedure very risky and would only be used in the most extreme emergency. In this case, extreme meant getting the Shuttle and its crew down as quickly as possible due to an engine failure or support system failure such as a loss of cabin pressure. Finding a way to eliminate the RTLS abort procedure became the goal of the researcher. The new liquid booster that Boeing was developing for the Space Shuttle seemed to be a possible solution. Since, unlike the current SRBs, liquid boosters were capable of being throttled, the goal became achieving a successful abort landing somewhere along the southern east coast and thus eliminate the need for RTLS.

1.6 Research Perspective

This research is a small part of the feasibility analysis currently under way by Boeing to evaluate the performance capabilities of its Liquid Fly Back Booster, a potential replacement for the Space Shuttle's current Solid Rocket Boosters (SRBs). With the current Shuttle fleet being almost two decades old, NASA has started looking for ways to extend the fleet's service-life by improving safety, reliability, and performance, and by reducing operations costs. NASA has placed contracts with both Boeing and Lockheed Martin to investigate the feasibility of Liquid Fly Back Boosters for the Shuttle. These contracts were originally placed in May 1997, extended in February 1998, and again in January 1999. Currently, NASA is interested in Boeing's LFBB design and has required Boeing to formulate possible methods to eliminate the RTLS abort procedure by sizing the system to perform a *TAL-from-launch* scenario in the case of an *Intact Abort*. *Contingency Aborts* are not being addressed by these contracts. To date, Boeing has completed work on the modeling of the LFBB's performance characteristics. The 1999 effort includes refining the aerodynamic shape of the LFBB to minimize the impact on the Orbiter, and to improve the fly back characteristics, and some technology development of TPS (composite structure and actuators) [12]. The next phase, *Preliminary Design and Prototype Development*, is on the horizon. This research addresses the *Contingency Aborts*, with an approach that will also eliminate the RTLS mode for those situations, while still minimizing the time of flight. The success of this research will lend support to Boeing's feasibility analysis of the Liquid Fly Back Booster's capability in meeting NASA's requirements, especially in improving system safety.

1.7 Research Problem

With the astronaut community perceiving that the as of yet untried RTLS abort mode will not succeed, it is this researcher's desire to eliminate this abort mode. Elimination of the RTLS abort mode entails developing an alternative, which will allow successful intact abort landings from liftoff until TAL availability, approximately L+150 seconds.

1.8 Research Goal

The goal of this research is to assess the combined Shuttle-Liquid Fly Back Booster's (LFBB's) capability to significantly alter its nominal flight trajectory and successfully land after declaring an abort. Furthermore, landing at a southern East Coast airport or military landing facility will minimize the time it takes to get the Shuttle and its crew back safely on the ground. It is desired to cover the entire ascent window from *tower-clear* until TAL availability with this new method of abort, and thus eliminate the need for RTLS. With the enhanced capabilities of the LFBB, it is theorized that all the aforementioned points are within the performance envelope of the LFBB.

1.9 Approach

The initial approach for attempting this goal centered on the concept of helicopter flight. A helicopter rides a column of air, and changes the pitch of its blades to vector this supporting column of air. This vectoring enables movement in any direction. A similar approach would be used with the Shuttle. The thrust vector of the Shuttle-LFBB launch system would be modified to allow for forward momentum, while at the same time delaying the inevitable decay of the flight path angle γ. It is theorized that the capabilities of the LFBB combined with yawing the Shuttle about its local vertical axis, will aid in the shaping of the abort trajectory in such a manner as to allow a successful landing at an airport or military landing facility.

The first step in this approach was the design of a nominal Shuttle/LFBB launch trajectory model in Fortran. Validation of this model was accomplished by comparing state vector data from this nominal model to equivalent LFBB performance data obtained from Boeing [13]. The Shuttle's state vector data was collected for each second up to the LFBB separation time of 135 seconds. These state vectors would act as the initial conditions for various abort points along the nominal trajectory. The nominal model was cloned into an abort version with modifications to account for reentry into the atmosphere and various throttle settings. Code was also included to ensure the Shuttle's external tank was jettisoned prior to exceeding two pounds per square foot dynamic pressure.

Success or failure was gauged against whether a suitable approach to a landing facility could be found while meeting certain conditions. The conditions included:

1) Did the abort trajectory terminate prior to, and above the Shuttle's Terminal Area Energy Management (TAEM) point? Vehicle energy is adjusted at the TAEM point with banking maneuvers so the landing site is not over or under flown. The TAEM point is defined as being 95 kilometers out from the end of the runway, and 25 kilometers up in altitude.

2) Since the abort trajectory's flight path angle would be larger than that of a normal approach, could the vehicle structure survive the accelerations encountered during pull-up when correcting for the proper glide path angle? This is referencing the use of a *modified skip-reentry* maneuver. NASA representatives. stated that during contingency situations the maximum acceleration force the Shuttle can withstand is five times the gravitational acceleration of the earth [8;37].

3) Associated with flight path angle is velocity. Is the model's trajectory termination point velocity low enough to prevent skipping out of the atmosphere during reentry? Ideal TAEM interface velocity is 762 meters per

second. An additional equation for the final pullout height, associated with the TAEM interface velocity, is checked to ensure the orbiter does not impact the ground. This check for final pullout height is also part of the *modified skip_reentry* calculations.

4) At the trajectory termination point, will the vehicle's heading allow intersection with a runway's heading alignment circle? Tangency to a heading alignment circle will lead to a proper final approach for a given runway.

1.10 Scope

Due to the infinite number of possible abort scenarios and the fact that this researcher wanted to investigate the performance capabilities of the Liquid Fly Back Booster (LFBB), the scope of this research was narrowed by a few conditions. First, an abort could be initiated for any number of reasons; the single Space Shuttle Main Engine out scenario was chosen to be the impetus for these aborts. This type of performance anomaly would truly test the capabilities of the LFBB compared to a support system failure, where the full performance capability of the Shuttle-LFBB combination still existed.

Secondly, the abort scenario initiation times were limited to periods in the trajectory where the LFBB's performance could be influential. This entailed restricting abort times to liftoff until approximately launch plus 119 seconds. This would allow for 16 seconds of LFBB trajectory influence. From L+119 seconds until L+150 seconds, where TAL picks up, it is assumed that NASA's East Coast Abort Landing (ECAL) procedures could be used.

To further limit the scope of this thesis, the abort times were placed at the extremes of the ascent trajectory phase of the launch. The researcher felt that if solutions for the worst case scenarios could be found, then the less demanding cases could be solved at a later date. The researcher defined the worst cases for an abort situation as occurring at one of two times. The first would be during the very first few seconds of launch; here the Shuttle is slow, encumbered by the majority of its propellant mass, and has most of the atmosphere to climb through. The second was at the end of the ascent trajectory where the number of potential landing sites has quickly diminished.

1.11 Executive Overview

This research work has shown that successful abort landings to the East Coast can be made during the period normally covered by the RTLS abort procedure. The inherent throttling capability of the Liquid Fly Back Boosters (LFBBs) has shown that abort scenarios, which take place within the first second of clearing the launch tower, or as late as 119 seconds into the ascent phase of a launch, can end successfully. Not only can the Shuttle and its crew land safely but also in a relatively short period of time the Shuttle can be wheels down at Savannah International Airport.

Switching to liquid propellant boosters eliminates the concern of not being able to act until the booster fuel is spent. According to the data collected by this researcher, the Shuttle can be on the ground in as little as ten minutes after declaring an abort situation. The enhanced performance capabilities of the LFBB are credited as being the reason for the success of this research.

The following chapters will detail the work of this research beginning with a detailed discussion of the problem definition and the method of resolution decided upon in Chapter 2. Chapter 3 contains a description of the methodology used in modeling the Shuttle's nominal trajectory as well as the steps used in creating an abort version of the trajectory model. Chapter 4 is an analysis of the data that was collected from the abort version of the trajectory model, and Chapter 5 summarizes the work accomplished with this research project in its entirety along with conclusions and recommendations for future work in this area.

2. Problem Definition & Method Resolution

2.1 Introduction

By the time this research work has gone to publication the first segments for the new International Space Station will have been launched into orbit. The work done in this thesis is connected with this new space station and a string of rocket mishaps that occurred in the mid 1980's.

After the loss of Challenger and a Delta rocket at Cape Canaveral, and the loss of a Titan IV rocket at Vandenberg AFB in California, America's space program was in dire straits. On the 3rd of February, 1986 President Reagan announced the formation of the Presidential Commission on the Space Shuttle Challenger Accident [27:40]. The effects of this Presidential Commission would influence all future American space programs. It effectively became a noose around the neck of the Shuttle program, and generated an insurmountable mountain of new expectations for the practically nonexistent expendable rocket program. No longer would any commercial payload, unless Shuttle-unique or of National interest, be allowed on board the Shuttle [28:41]. The responsibility for launching commercial payloads would shift to expendable rockets such as the Atlas, Delta, and Titan. With this being the case, many people averse to an American Space Program began voicing their views even louder. Critics of NASA questioned whether the United States needed a Shuttle at all. They argued the money spent would be better used on other programs that could benefit a larger segment of society. These critics pointed out that the then Soviet Union had shelved its plans for the Buran, a Shuttle-like reusable vehicle. They went on to state that because of the cost involved, the U.S. should shelve the Shuttle as well. These same critics failed to understand that the Soviet Union had already built multiple Burans. Termination of the Buran project stemmed not from a desire to save their economy, but because propellant funds were being spent on tanks and nuclear weapons [10;35].

All the time this was going on, advocates for a modern U.S. space station were fighting for the program's existence. On numerous occasions funding for Space Station Freedom was either cut or excluded altogether from the fiscal budget. Finally in 1993, after President Clinton called for a final design selection, and inclusion of foreign governments in the development of the space station, it looked like the new International Space Station (ISS) would become a reality [24]. The affect this had was a rejuvenation of the Shuttle program's lifeblood. With 45 expected launches taking place to build the new station, not including future missions for service and support to the ISS, the Shuttle program's existence, for the moment, seemed secure. For that matter, the entire U.S. Space Program benefited from the ISS finally entering its implementation phase. Now with a real purpose, the U.S. Space Program would no longer be thought of as a nebulous subject area.

With the expected service life of the space station now exceeding that of the Shuttle's, the development of the Shuttle's replacement got into full swing. Besides planning for the next generation of reusable space vehicles, ways to extend the life and increase the capability of the current Shuttle fleet had to be found. The ISS would be fully functional and in need of service and supply missions well before any future vehicle would be taking its maiden voyage. This then, is where this research work ties in. With the International Space Station's 220-mile high orbit lying at an inclination of 51.6°, increasing the Shuttle's performance while maintaining its high safety standards became of paramount importance. The clock was ticking and a solution had to be found.

In 1998, Boeing proposed its Liquid Fly Back Booster (LFBB) to NASA as a means to meet the increased performance requirements while reducing the overall expected cost of the Shuttle program. The LFBB design proposes to increase payloads to all planned orbits. Specifically, it would be able to lift 47,000 pounds to the ISS, compared to contemporary Shuttle payloads of 35,000 pounds. Cost would be reduced by the simple fact that fewer flights would be necessary during logistical support operations to the ISS. Boeing further stated that with the improved performance capabilities of the LFBB, it may be possible to eliminate the RTLS abort mode. NASA was interested in this as well as pursuing a *TAL-off-the-pad* capability. These last two points were the basis for this thesis. A limiting factor, which would force the use of RTLS as an abort mode, was not being able to alter the Shuttle's trajectory until the Solid Rocket Boosters (SRB) had stop thrusting. Boeing saw that liquid engines, which are capable of being throttled, clearly possessed some key advantages over the current SRB design. Also, Boeing was interested in using the capabilities of the LFBB to get the Shuttle on the ground quicker than what TAL offered.

With elimination of RTLS being purely theoretical to this point, the researcher decided that the best method for finding solutions to these quandaries would be the development of a simulation using Fortran. One of Fortran's strengths is its ability to handle massive quantities of complicated mathematical expressions. Since this simulation was expected to deal with the mathematical resolution of multiple equations describing the state of a rocket, as well as numerous iterations involving the evaluation of the mathematically expressed rocket trajectories, a simulation in Fortran seemed to be the only viable alternative. This simulation would attempt to model a normal launch and ascent trajectory. Once the *nominal* model had been created an *abort* version would be spawned to simulate abort scenarios at different times during a normal launch. The Fortran model would then solve for trajectory solutions that would potentially eliminate RTLS and provide a quicker means for getting the Shuttle and its crew back on the ground as compared to TAL.

After the proposal, *East Coast Abort Modeling with RTLS Elimination for the Space Shuttle-Liquid Fly Back Booster (LFBB) Launch System*, was submitted and approved by Boeing and the researcher's adviser, work began in earnest. Work focused on understanding the different facets involved in designing a trajectory model and how to incorporate these different parts into a Fortran simulation. As previously stated, the goals of the simulation would focus on proving whether the LFBB could or could not aid in the elimination of the RTLS abort mode. And, whether the LFBB could provide a means to get the Shuttle and its crew back on the ground in a shorter period of time than what TAL offered. With an APU failure or the loss of a windshield, the 25 minutes needed to execute a TAL may prove to be too long.

The sections that follow give a top-level explanation of the process used in developing a method to resolve the questions: Could the RTLS abort mode be eliminated? And, is there a faster means than TAL for getting the Shuttle back on the ground? Chapter 3, Methodology, contains a detailed explanation of the high-level process elements presented in the next section and how these elements were incorporated into the Fortran simulation.

2.2 Elements of a Rocket Trajectory

In tackling this research problem, understanding the different parts that comprised a rocket's launch trajectory consumed the majority of energy. In trying to track down sources of information it initially looked to be a daunting task. Searches conducted using FirstSearch and the Defense Technical Information Center (DTIC), at the Air Force Institute of Technology's (AFIT) library, turned up a large selection of resources, but unfortunately only a few were useful to this research area. The majority of the resources discovered were largely methods dealing with improvements to, or the effects on, various components of the Shuttle. This apparent glut of information seemed to originate shortly after the Challenger accident.

Out of all the information available, two useful pieces were gleaned. The first was MASTRE [23], a computer simulation that would shape a trajectory to meet specific mission requirements. This would be useful in designing the optimum trajectory for launching a given payload into a particular orbit. The second item, A Simulation Model for Probabilistic Analysis of Space Shuttle Abort Modes, by R.T Hage [11] turned out to be very useful. This simulation model focused on the propulsion elements of the Shuttle system (i.e., external tank (ET), main engines, and solid boosters). The model was developed to provide a better understanding of the probability of the occurrence, and successful completion of abort modes during the ascent phase of the mission [11;1]. Hage's model documented the various abort modes along with a rather comprehensive listing of possible abort causes. Upon executing the model, data was input into the program to determine the frequency of occurrence of the various ascent/abort options for the flight of STS-32. The model was setup to run 1,000,000 simulated launches. Out of these 1,000,000 runs, the number of various abort scenarios and their outcomes were recorded. Information that was of interest included data showing out of 1,000,000 launch attempts, 21,677 required a RTLS abort mode attempt. Of the 21,677 RTLS attempts, 327 ended in a RTLS catastrophe. The data presented had its valid points, but it was still felt that doing away with the RTLS abort mode would prevent possibly 327 catastrophic situations from occurring.

Having exhausted the useful resources found by DTIC and FirstSearch, it was decided to pursue some personnel expertise. After a few phone calls to NASA's Johnson Space Flight Center and Cape Canaveral, some valuable resources where located. During one phone call with a Doug Whitehead [37], it was pointed out that much of the corporate knowledge from the Apollo era was gone forever. Little, if any, formal steps were in place to collect and archive all the data generated during America's race to space. Efforts were now being made to write everything down about the Shuttle program, but the best source of information still lay with those people still around from that by-gone era [30;37].

From the phone calls to NASA and the Cape, numerous contacts were made with people associated in one way or another with some aspect of trajectory design. Probably the most noteworthy and richest source of information was obtained from a Mr. Dennis Bentley. During his career at NASA he had done work on Apollo, various missions to Mars and other planets in our solar system, and was currently working on the X-38 project. Most importantly for this research, he had chaired the Shuttle Abort Panel in the early 1980's [2]. It was during this panel in 1983 that the East Coast Abort Landing

(ECAL) mode was adopted for the Shuttle program. The information Mr. Bentley provided about ECAL benefited many aspects of this research work.

After reading through the various sources and referencing the conversations with Mr. Bentley, it was decided the best approach to modeling a rocket's trajectory was to break it into segments. The trajectory model was divided into two phases, each containing various elements. The first phase, Ascent, covered the period from initial liftoff until abort scenario initiation. The Ascent phase was comprised of the following elements:

- Gravity Turn Trajectory Modeling
- Dynamic Pressure Modeling
- Mass Dynamics
- Nominal Model Validation
- Abort Mode Fundamentals

The second phase Abort Reentry, included the following list of elements:

- Abort Trajectory Modeling
- External Tank Separation Conditions
- Aerodynamic Force Modeling for Reentry
- Final Trajectory State Analysis for Shuttle Approach

Phase two extended from abort initialization until the point where the final conditions had been met for Shuttle approach.

What follows is a brief description of the elements comprising each of the two phases, as well as a description of the source for the information used in each element.

2.3 Phase One: Ascent

2.3.1 Gravity Turn Trajectory Modeling.

During the initial stage of the ascent, where aerodynamic loading is greatest, the thrust vector of the rocket is kept aligned with the velocity vector so that the vehicle is not torn apart by lateral forces. This particular type of trajectory is called the gravity turn trajectory due to the fact that the force of gravity is the single force that is causing the rocket to rotate from the geocentric vertical orientation to one that is horizontal. Numerous references, both human and documented, were available to discuss the techniques used in implementing this portion of the trajectory model. Of most benefit were two books; the first, Space Propulsion Analysis and Design by Humble, Henry, and Larson, provided a top level understanding of the gravity turn trajectory [16:69-71]. The second, Spaceflight Dynamics by Dr. William Wiesel, gave an in-depth explanation of the math involved with the gravity turn technique as well as comprehensive sections on the rocket equation, the staging of rockets, and atmospheric affects on trajectories [39:193-254].

Information gleaned from the sources indicated that the equations of motion could be simplified and the earth treated as being flat. This would provide a good first order approximation to the actual launch trajectory model. The reason this simplification works is that during the initial portion of the ascent trajectory the Shuttle is moving at a relatively low velocity. Centrifugal acceleration is then treated as a force thus allowing the approximation to flight over a flat earth. This treatment of flight over a flat earth is beneficial when attempting to describe the location of the rocket when it is near the surface of the planet such as during launch or reentry.

The first three equations of motion gave the altitude, latitude, and change in longitude of the vehicle. The approximation had yet to take into account the effects of a rotating planet, the effects of varying thrust, or the change in mass. Vinn, Busemann, and Culp's book, Hypersonic and Planetary Entry Flight Mechanics [34:19-28] gave a detailed account of the effects of a rotating planet on the equations of motion. Understanding how the effects of a rotating earth could affect the equations of motion led to the development of three other equations of motion. These three would show the effects the time rate of change would induce in the velocity dV, flight path angle $d\gamma$, and heading angle $d\psi$ of the vehicle. Also, thrust T and mass m, are broken up into their respective components. These then get added to the appropriate equations of motion. The effects of thrust and mass would then properly influence the equations of motion defining the trajectory.

The only forces missing from the ascent portion of the trajectory were the forces of drag and lift. But, Dr. Wiesel's discussions in his book about lift and drag forces [39:240], allow for a simplification during the initial ascent phase of the trajectory to take place. The two forces, lift and drag, can be eliminated with minimal effect. The reasoning behind this is that when the rocket first leaves the launch pad, it is traveling straight up at a relatively low rate of speed thus eliminating the influences of lift or drag. By the time substantial velocity has built up, the vehicle has climbed through the densest region of the atmosphere. This again supports the simplification of eliminating the forces of lift and drag during the initial phase of the ascent trajectory. Reentry is a different story, drag and lift are significantly influential in this portion of the abort trajectory as will be discussed below.

At some point the trajectory model must deviate from the gravity turn trajectory if it becomes desirous to head towards a particular target. The variation to the flight trajectory is accomplished by taking the last three equations of motion, and solving for thrust instead of dV, $d\gamma$, and $d\psi$. Using trigonometric identities it is possible to solve for the angle of attack α, yaw β, and thrust T necessary to obtain certain values of dV, $d\gamma$, and $d\psi$. This then is how components of the six equations of motion are *pre-solved* so as to obtain specific values for the time rate of change of velocity, heading, and flight path angle.

2.3.2 Dynamic Pressure Modeling for Ascent.

Dynamic pressure has a way of dismantling a rocket if allowed to rise unchecked. To match the performance data supplied by Boeing [13], and strive for as realistic a model as possible, it was necessary to include a detailed model of the atmosphere. Chosen for this was the model atmosphere developed by Regan and Anandarskarian [29: Appendix A]. A key assumption of this model was that atmospheric density was relatively insignificant above 50 km altitude. When incorporated as a Fortran subroutine and given vehicle altitude and ground level pressure, this atmospheric model returned values for pressure at altitude, density at altitude, and mean free path. Using the vehicle's velocity at a particular altitude, it was then possible to calculate dynamic pressure based on the density returned by the atmospheric model.

All the while the rocket is climbing off its launch pad, calculations were being done to solve for the dynamic pressure the vehicle was experiencing as it climbs up through the atmosphere. NASA and Boeing both place the maximum dynamic pressure, or *Max Q*, for ascent at approximately 750 pounds per square foot (psf) before structural deformation would occur [2:14]. The trajectory model was designed to sense when dynamic pressure had fallen below 2 psf. When 2 psf had been reached, the model signaled the guidance software that it was now possible to change heading, angle of attack, yaw, and flight path angle without concern for the tremendous lateral forces that would have been experienced anywhere inside the earth's atmosphere.

2.3.3 Mass Dynamics.

In modeling a rocket's trajectory, a key item was how the mass of the rocket would change with time. This *thrust* was of course tied directly to the rocket equation. As the rocket climbed in altitude, the atmospheric pressure changed. This change in pressure affected the levels of thrust, and this in turn affected the mass flow rate, or the rate at which mass was ejected from the engine nozzles. The model would be required to keep track of the atmospheric pressure for a given altitude, and use this information to modify thrust. Boeing made information available concerning nozzle exit areas. This information when combined with atmospheric pressure for a given altitude, and the value for thrust in a vacuum for the different engines involved, gave thrust at altitude. Numerous sources helped in understanding the concepts of rocket propulsion and its affects on changing the mass of a vehicle. A good propulsion overview was provided by Larson and Wertz's Space Mission Analysis and Design [20:640-42]. Of primary benefit were again; Space Propulsion Analysis and Design [16:6-13], and Spaceflight Dynamics

[39:193-200], which handled the topic of propulsion with finite detail. Isakowitz's International Reference Guide to Space Launch Systems [17:273-289], was an excellent resource concerning Shuttle propulsion specifications. A further aid in understanding various propulsion concepts was Mark Hines, a trajectory specialist at Boeing [14].

2.3.4 Nominal Model Validation.

A major source of information used in validating the nominal model was the performance data provided by Boeing [13]. Values generated by the model for velocity, altitude, mass, thrust, and dynamic pressure were compared to the expected values from the LFBB performance data. As the model increased in complexity by taking the affects of a rotating earth into account, the differences between the nominal model and the expected values of Boeing's performance data decreased. The result of this variant decrease was that fidelity improved. Lastly, Isakowitz's International Reference Guide to Space Launch Systems [17:273-289], provided references for maximum dynamic pressure and SSME thrust capabilities.

2.3.5 Fundamentals of Shuttle Abort Modes.

Understanding the different abort modes and the times at which they could take place was very important. A thorough understanding of what each abort mode entailed, both performance-wise and support-wise, would aid in understanding just how to approach coming up with an alternative abort method for RTLS. Information about the various abort modes was obtained from a number of sources. First, general information about abort modes was obtained from NASA's press-release web site [25;28:14-17]. Next, information that had been collected from the DTIC search mentioned previously, was analyzed [11]. The information from Hage's A Simulation Model for Probabilistic Analysis of Space Shuttle Abort Modes [11], gave probabilities for potential successes from the various abort modes based on the cause of the abort. It was desired to garner from the analysis, an understanding of what exactly initiated an abort scenario. Would any reason suffice for an abort to the East Coast? Also, what were the deciding factors for the order of precedence. Not every abort scenario would work for a given situation. It was important to understand the conditions that would dictate which mode was chosen over another.

There is a definite order of preference for the various abort modes. The type of failure and the time of the failure determine which type of abort is selected. In cases where performance loss is the only factor, the preferred modes would be ATO, AOA, TAL and RTLS, in that order. The mode chosen is the highest one, in the previously given order, that can be accomplished with the remaining vehicle performance capability. In the case of some support system failures, such as cabin leaks or vehicle cooling problems, the preferred mode might be the one that will end the mission most quickly. In these cases, TAL or RTLS might be preferable to AOA or ATO. A contingency abort is never chosen if another abort option exists. During flight, Mission Control Center-Houston is prime for calling these aborts because they have a better handle on the overall picture. They are more aware of all the different Shuttle systems than the crew flying it. Periodically, calls are made to notify the crew when certain abort modes are no longer available [25:28]. Since this research focused on the elimination of the RTLS abort mode, it was necessary to concentrate on the lowest energy type of abort scenarios. Ultimately this forced the consideration of only performance related abort scenarios.

Since Boeing had completed its own model for a single LFBB engine out scenario, emphasis was placed on the Shuttle's main engines. The logical progression seemed to start with a single SSME engine out, work a solution and if time permitted, work the dual SSME abort scenario.

Information was then gathered about single SSME out scenarios. In studying the information collected from the various resources already mentioned, particular interest was paid to any information that dealt with SSME and LFBB thrust levels and their corresponding throttling capability. Information collected from Boeing stated each of the two LFBBs would be comprised of four booster engines for a total of eight. Each booster engine would have an approximate thrust capability of 1,000,000 $lbs._{vac}$. Nominal operation would be at 70%, with a throttle range from 50-100% [13;15;12]. NASA figures place the capability of the SSMEs at 470,000 lbs_{vac}, with a throttling capability of 50-104.5% for Block I SSMEs [17;28]. NASA planned to introduce a Block II SSME during 1999. The Block II engine would be capable of throttle settings from 50 - 109% [38].

2.4 Phase Two: Abort Reentry

2.4.1 Abort Trajectory Modeling.

This portion of the trajectory model would take over from the initial gravity turn method as soon as an abort scenario was initiated. The abort trajectory model only incorporates a few more items than the gravity turn portion of the model. Although, it should be pointed out that the items in question are the forces of lift and drag, and that even in the gravity turn portion of the model, their values were being calculated. But, as mentioned in Section 2.3.1, the influences of the lift and drag forces during the initial vertical flight are so insignificant as to be easily ignored. All other effects and influences described in the gravity turn section, including the six equations of motion and their modifications due to thrust and mass, apply to the abort portion of the trajectory model.

2.4.2 Modeling External Tank Separation Conditions.

During an abort attempt the separation of the External Tank (ET) would always be a point of concern for the crew. Any re-contact between the Shuttle and ET would almost guarantee disaster. NASA documentation gives 2 psf dynamic pressure at Mach 1.3, eight minutes after launch, as the desired ET separation conditions [28:17;287]. Dennis Bentley of NASA stated that during his work on ECAL, a contingency *high rate separation* for the ET was developed for both ECAL and RTLS aborts [3]. The values associated with this separation method were 9 psf dynamic pressure at Mach 5 with an angle of attack of –2°, and at an altitude of approximately 200,000 feet. He went on to state if a method was devised to control the flight of the ET during separation, it was theoretically feasible to separate the orbiter from the ET at around 300 psf going 300 knots. He based his statement on the Shuttle separation flight tests that were conducted off the back of a 747 jetliner in the early 1980's [3].

As with the ascent portion of the launch model, the abort reentry section keeps track of the dynamic pressure. When 2 psf is sensed upon reentering the atmosphere a check is made to see if the LFBBs are still attached, if they are, they are commanded to separate. A short time after LFBB separation the ET is commanded to separate. The model's 2 psf separation was conservative, but if the ET successfully separated, then it would ensure no re-contact would occur during an abort scenario while using this model.

2.4.3 Aerodynamic Force Modeling for Reentry.

Blanchard, Larman, and Moats' journal article titled Rarefied-Flow Shuttle Aerodynamics Flight Model [5], was the basis for the code included in the trajectory model that dealt with the lift and drag forces associated with reentry. Rarefied – flow is the transition region between free-molecule flow and the hypersonic continuum. It exists approximately from 60 – 160 km in altitude. These regions were determined based on the Knudsen number, Kn. Kn is the ratio of mean free path to the mean aerodynamic chord (MAC). The MAC for the Shuttle is 12.058 meters [5:553].

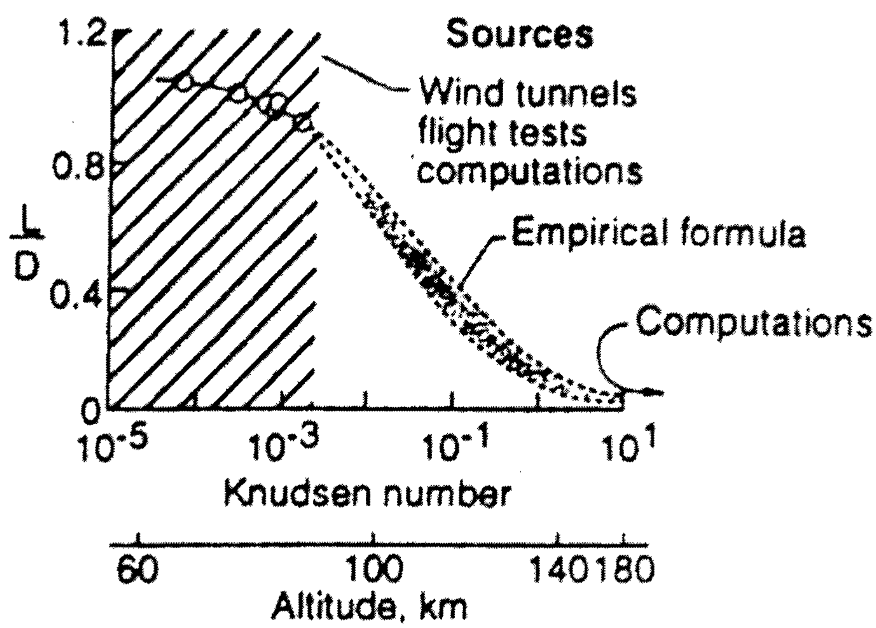

Figure 2-1. Knudsen Number: Orbiter Rarefied-Flow

Figure 2-1 shows the relationship between the different methods used over the years to derive the rarefaction parameter Kn, or the Knudsen number as it relates to the orbiter's lift to drag ratio (L/D), and altitude. The region $10^{-3} < Kn < 10$ is the rarefied-flow transition region. This region uses an empirical formula to bridge between the hypersonic continuum ($\sim Kn < 10^{-3}$) and the free molecule flow ($\sim Kn > 10$) regimes [5:550]. As can be seen in the figure, the Shuttle spends the majority of its time in the rarefied-flow region. The empirical formulae governing this regime are broken into three parts: Hypersonic continuum, free molecular flow, and a bridging formula. All three calculate C_N and C_A, which are aerodynamic coefficients in the normal and axial directions. These coefficients are functions of pitch angle α. Also, methods used to convert C_N and C_A into C_L and C_D will be shown in the methodology chapter, Chapter 3.

2.4.4 Final State Conditions.

The goal of this research was to attempt the elimination of the RTLS abort mode by landing the Shuttle and its crew safely at the first available landing site. In Section 1.3.5 Figure 1-2, a list of ECAL proposed landing sites and their corresponding runway lengths was given. The point being made by Dennis Bentley, when he first assembled this list, was a significantly long runway would be needed due to the high reentry speeds the Shuttle would be expected to have while attempting this type of abort landing. Mr. Bentley stated "12,000 feet is preferred, but in a pinch 8,000 would work with the drogue chute" [3].

For the purpose of this research, the first choice when it came to selecting a potential landing site was finding the closest major airport or military landing facility so the Shuttle could be on the ground in the shortest possible time. This abort model would use the same list of airports shown in Figure 1-2 with the addition of two others. The first, Jacksonville International Airport (JAX) located outside Jacksonville, Florida; possessed a 10,000 by 150-foot runway that could be approached from either 70° or 250°

azimuth. The second addition was Savannah International Airport (SAV) Savannah, Georgia. SAV had a 12,000-foot runway that headed directly east-west, as well as a 10,000 foot runway that headed directly north-south. Depending on where the abort trajectory finally ended up, other possible landing sites might also exist. The important point being made was that the potential runway had to be at least 8,000 feet long. This 8,000-foot value was the same minimum length used for TAL and ECAL abort landings [2:25].

Normal reentry interface occurs at 400,000 feet altitude and 5,063 miles from the landing site [28:12]. At this point, the slope of the flight path has at a –0.857° angle. When the orbiter crosses the Terminal Area Energy Management (TAEM) interface at 83,000 feet altitude and 59 miles out, this angle would have increased to 14.91°. Upon reaching the *Steep Outer Glide Slope* at 10,000 feet altitude and just 7.9 miles from the landing site, this angle would have reached approximately 20° [28:14].

Analyzing the nominal approach characteristics of the Shuttle aided the understanding of what type of trajectory modifications the abort trajectory model would have to incorporate. Characteristically, the closer to the launch pad the abort-landing site was, the steeper the angle of approach would be. The expected acceleration that would be experienced during a modified skip reentry was to become one of the key trajectory termination values. The modified skip reentry would be required so as to acquire the proper flight path angle for approach. NASA's information gave 5 g's, or 5 times the gravitational force of the earth, as the maximum the Shuttle structure could withstand before deforming [37:9].

At the end of chapter eight of Dr. Wiesel's book, Spaceflight Dynamics, a skip reentry equation exists [39:252]. This equation calculated the maximum lift acceleration experienced by a vehicle at the bottom of a pull-up maneuver while attempting a skip reentry. Since a primary goal of this research was getting the Shuttle down quickly, a higher than normal flight path angle for approach was expected. To compensate for this, and get the Shuttle onto the glide slope for approach, a modified version of the skip reentry was adopted. Once the initial pull-up was accomplished it was expected that the Shuttle crew would quickly pitch the nose back down, giving a negative angle of attack α, as the orbiter headed back up the other side of the pull-up trajectory. This would prevent skipping back out of the atmosphere, and would allow for the acquiring of the glide slope in a shorter period of time.

These then would become the primary criteria for declaring a successful landing attempt with this model: After getting into range of a potential landing site, could the orbiter pull out of a steep flight path angle before hitting the ground? And, could the orbiter acquire the glide slope for final approach without ripping the wings off by exceeding the 5 g limit set by NASA? In addition to these criteria, the *g-forces* could not exceed the 5 g limit during any part of the abort trajectory, this included: launch, skip reentry, and final TAEM interface. Also, to further qualify a success, the Shuttle had to be within range of the landing site. This meant at the TAEM point, the Shuttle's altitude and velocity had to be relatively close to the expected nominal values of 762 m/s at an altitude of 25 km. It was assumed the velocity and altitude could not be less than the values required at the TAEM point if the runway was to be acquired. Further, it was assumed that the velocity and altitude could exceed the values of the TAEM point with the excess energy being bled off with *S* maneuvers or possibly circling the runway if need be. With these criteria in mind, the thesis progressed into the programming stage, which is discussed next.

2.5 Modeling Techniques in Fortran

Multiple sources can be sited for the assistance that was provided in programming this trajectory model in Fortran. Foremost to be mentioned, was the expert advice offered by Dr. Wiesel. Numerous times his advice was sought on how best to handle a particular problem in Fortran. There were also a plethora of books available on the subject of programming in Fortran. Of the four sources sited in the Bibliography, two stood out as being the greatest help when it came to published programming techniques. The Essentials of Fortran was the most helpful of the programming resources. Never was the book found to be lacking for some kind of answer [31]. The second source of excellent information on Fortran programming was a book by Koffman and Friedman titled Fortran with Engineering Applications [18]. What was beneficial about this text was that it was quite contemporary and included many examples that were applicable today. With Fortran being a language that had its heyday in the 1970's and 1980's, it was surprising to find a book written in the 1990's that had so many practical applications and examples for problems that exist today. The book's engineering slant was most helpful with developing methods to deal with the multi-faceted problem of designing a rocket trajectory.

Probably the single most important routine in the trajectory model was the method used to integrate the incremental values for the equations of motion. This section of code called *haming*, was provided by Dr. Wiesel [40]. The core of a computer program that simulates a launch vehicle's trajectory is a method that numerically integrates the equations of motion for the flight under study [16:74].

Numerical integrators fall into numerous classes: predictors, extrapolators, and predictor-correctors. Extrapolators assume constant functions, the obvious problem with this is that the new value could diverge from the expected values. Higher order extrapolators can correct for this and provide better approximations but they still suffer from divergence. If data points for the state vector $\underline{\mathbf{x}}$ and its rate of change $\underline{\mathbf{f}}$, generated by the before mentioned extrapolator, are used as data points for polynomials that are run in time, then a predictor method exists. Both extrapolators and predictors step their way into the future using data from the current instant and the immediate past. Once new datum are created they can be evaluated in the equations of motion, the question of course is whether or not the results are accurate and if they improve the quality of the new state vector [40:119]. If a higher order polynomial is run through the previous data points, and the new equations of motion, now exists a predictor-corrector method. The advantage of the predictor-corrector method is that it need not suffer from divergence as extrapolators might.

Predictor-correctors are not perfect, the numerical analyst Hamming recognized that if dx/dt = 0 the predictor-corrector method would diverge exponentially. Hamming went on to note that a predictor-corrector method is a set of linear finite difference equations, whose forcing function is the actual system to be integrated. Just as with linear differential equations, the solution consists of a homogenous part and a particular part. If the homogenous system (the integrator algorithm alone) is itself unstable, then eventually the actual solution function is buried under the exponential divergence of the unstable homogenous part [40:119].

The sub-routine *Haming* includes the algorithm devised by Hamming to deal with the issue of creating a numerically stable numerical integration algorithm. One problem with using a predictor-corrector is that several points are required for them to begin, not just a set of initial conditions. Haming is a fourth order predictor-corrector so 4 *previous* points are required. Since only a single set of points is usually given by the initial conditions, another method must be utilized to initiate the Hamming algorithm. *Picard Iteration* was just such a method that could be used to initialize Haming. This is a slow and expensive initiation routine. It evaluates the first three steps in time in order to calculate the first 3 points for the state vector. After this initiation step, Haming has the 4 points it needs to begin. It is now ready to begin predicting future values for the equations of motion. This then is the integrator that would be used to evaluate the equations of motion that comprised the abort trajectory model [40].

2.6 Summary

This completes the description of the various elements that made up the ascent and abort-reentry phases of the trajectory model design. Along with the description of the elements, an attempt was made to point out the key contributors and any revealing insights they might have had for the development of this model. As seen from the element descriptions for each phase, many aspects overlap between the two phases. Differences were minimal and centered around the addition of lift and drag force evaluation as well as setting up the desired end conditions for the trajectory that if met, would indicate success.

In conclusion, this chapter helped in the understanding of the specific goals of this thesis by defining the problem and explaining the steps involved in developing an approach to a potential solution. The primary goal would be the elimination of RTLS by developing an alternative abort method that would allow for successful aborts to landing sites along the East Coast. This meant the alternative abort method would have to be capable of successfully completing an abort scenario that initiated anytime during the period from just after clearing the launch tower until TAL availability. As for the potential abort points between the two extremes, if successful abort landing conditions are met for the extremes, then theoretically the trajectories existing between these two points should have solutions as well. The secondary goal would be getting the Shuttle down in a shorter period of time than what TAL offers.

Equally important, this chapter helped define what a successful abort-landing attempt constituted. For a success, the end conditions would be those values for altitude and velocity that best match the values documented by NASA for the TAEM interface point. A check would be included for substantiating which trajectories qualify as successes. This check would test for maximum lift accelerations and would indicated if the structure of the Shuttle could withstand these forces. This chapter also discussed the incorporation of the modified skip reentry for proper glide slope acquisition. This reentry method would include a check for altitude that would answer the question: Does the Shuttle's flight path terminate at the TAEM interface or at ground level?

Chapter 3 will now describe the methodology used for the development of a nominal trajectory model. Chapter 3 then expounds on the modifications that would have to be made to this nominal model so as to create the abort trajectory version. This abort trajectory model would ultimately be used for evaluating potentially successful abort landing trajectories.

3. Methodology

3.1 Introduction

This chapter explains the methodology used to analyze the performance of the Shuttle Liquid Fly Back Booster (LFBB) launch system through the use of a simulation model of possible East Coast Abort (ECA) scenarios. There are two phases to the discussion. In phase one, the design and testing of the baseline simulation model are discussed as well as the parameters used to validate the model. In phase two, an abort variation of the nominal model is created. The abort model will include routines to test whether the enhanced performance characteristics of the Liquid Fly Back Booster (LFBB) will allow modifications to the in-flight trajectory, thus allowing for emergency landings at East Coast airports or landing facilities.

The abort version will be the centerpiece for this thesis's actual research work, which will be accomplished in Chapter 4. In Chapter 4, abort scenarios will be presented for a simulated 51.6° inclination launch. This inclination simulates the high number of expected Shuttle launches necessary for the building and servicing of the new International Space Station. Chapter 4 will then analyze the results from using various control variables to solve for successful abort trajectories, which are based on the models developed in this chapter.

During the development of the launch model, some concerns were kept at the forefront of the design process; Section 3.2 discusses the creation of the launch model and the philosophy used in balancing these concerns. In phase one, the design of the launch model consisted of three parts. The first part, covered in Section 3.2.1, discusses modeling the ascent of the vehicle as a gravity-turn trajectory. Modeling this portion of the launch as a gravity-turn trajectory allows simplification of the equations of motion by treating the earth as flat. This allows solving for common *aircraft-type* parameters such as altitude and downrange distance. Since the vehicle is initially moving slow relative to the earth, the effects of the spherical earth on its trajectory can be ignored. Parts two and three of the launch model design are contained in Section 3.2.3. In this section, the gravity turn trajectory is refined to include the effects of a rotating spherical earth. Part two of the model design deals with the inclusion of these effects into the equations of motion. Part three focuses on the equations of force that will be of paramount interest as the model moves into phase two. Although the influences of the equations of force are minimal during launch, since angle of attack and yaw are both zero, the equations of force will be the primary method by which phase two will alter the Shuttle's trajectory and provide solutions for abort landing trajectories.

The verification and validation of the simulation model are handled in Section 3.2.5. Sections 3.2.5.1 and 3.2.5.2 explaining how the initial conditions were derived and what final conditions were desired, respectively. Section 3.3 begins phase two. It addresses modifications the launch program had to undergo so as to create the abort version of the trajectory model. Minor modifications to the launch model were necessary so that the peculiarities of reentry were addressed. These modifications would be used in Chapter 4 to solve for solutions to abort trajectory scenarios. Section 3.2.4 explains how the changes, necessary to create the abort launch model, were included in the Fortran code. Finally, Section 3.4 summarizes the work done in this chapter.

3.2 Phase 1: Launch Model

In developing a model a balance must be struck by its creator. On one hand an infinite amount of detail will result in a model which possesses a high fidelity, but the amount of time and effort invested will also approach infinity. On the other hand, too little detail will have just the opposite affect; a poor fidelity model may result which may not simulate even the most basic of functions. Keeping this balance in mind, the goal in developing this Shuttle-LFBB launch model became trying to match the data, which was supplied by Boeing Defense Space Group, Downy CA., to the expected performance characteristics of the Shuttle-LFBB launch system.

3.2.1 Assumptions.

During the development of the launch model some assumptions had to be made in accordance with the balance discussed in the previous section. Even though some assumptions were made, they would have little impact on the results of this thesis.

During ascent, the launch model assumes a constant thrust. As long as the state vector elements created by this model show a close correlation to those values supplied by Boeing, then this assumption will have little detrimental effect. The second assumption deals with the launch model beginning just as the Shuttle clears the tower, at launch plus 9 seconds. The model initiates at the top of the launch tower because the calculations that have velocity terms in the denominator would have a tough time dealing with the initial velocity being zero. Efforts were made to calculate correct initial values for velocity, acceleration, atmospheric pressure, mass, and thrust at this altitude. Albeit, these changes are small compared to starting at the base of the launch tower. High levels of accuracy though, would pay off as the model propagates through the trajectory. Any error, no matter how small, could become substantial over time. How these initial conditions were derived is discussed in Section 3.2.5.1. A Third assumption is that the atmosphere behaves as depicted in the model developed by Regan and Anandarskarian. In their model the majority of the influential atmosphere exists below 50 km [29: Appendix A]. The fourth assumption deals with the aerodynamic forces the Shuttle experiences during its ascent. Since the most substantial portion of the atmosphere is traversed during the gravity turn portion of the launch trajectory, the values for C_D and C_L can be assumed to be one. This choice of $C_d = 1$ was based on comparing the values of C_d that both NASA and Boeing provided. NASA used a $C_d = 2.0$ while Boeing's averaged to a $C_d = 0.393$ from t_0 to launch plus 135.1 seconds, or when nominal LFBB separation would occur [15:27]. Again, this assumption is possible since the forces generated by these two coefficients are minimal during ascent. With the thrust and velocity vectors aligned during the gravity turn portion of the trajectory, no sizeable aerodynamic forces exist. Lastly, the model assumes that the spherical earth is rotating in such a manner that the coriolis acceleration term, $2\omega V$, is retained but the higher order terms such as $\omega^2 r$ can be neglected, ω is small by itself, when considered at a higher order it becomes insignificant.

3.2.2 Gravity Turn Trajectory.

Rockets are inherently designed to carry large axial loads. At the same time, rockets conserve structural mass by limiting support mass for transverse loads. Large rockets are incapable of flying through the atmosphere at an angle of attack, attempting this at several times the speed of sound would generate aerodynamic loads that could cause a catastrophic failure of the booster structure [39:217]. Keeping this in mind, this portion of the trajectory model keeps the thrust vector aligned with the velocity vector of the vehicle at all times, with one exception, which will be explained later. During flight to orbit, the vehicle must translate from a vertical orientation at lift-off to a horizontal one at burn out. This translation is accomplished using the inherent dynamics of the gravity-turn trajectory [39:217].

Figure 3-1 shows a rocket in flight over the earth with the corresponding forces acting upon it. Taking all these forces into account, Newton's second law, **F=ma,** can be written as

$$\mathbf{T} + \mathbf{D} + \mathbf{L} + (\mathbf{-mg}) = \mathrm{m}\mathbf{a}. \qquad (3\text{-}1)$$

T is thrust, **D** is drag, and **L** is lift. What is significant about equation (3-1) is that the force of gravity **(-mg)** in this instance is modified to account for *centrifugal force*, this centrifugal force is sometimes referred to as *centrifugal lift* in aircraft dynamics. This is not a force at all, but an inertial acceleration term on the wrong side of **F**=m**a** [39:21]. The importance of this is the model can now treat flight over a spherical earth as if it were flight over a flat earth. The only components of the state vector that are affected by doing this simplification are latitude and longitude, but by the use of spherical trigonometry these can be solved for easily as will be shown in Section 3.2.3.

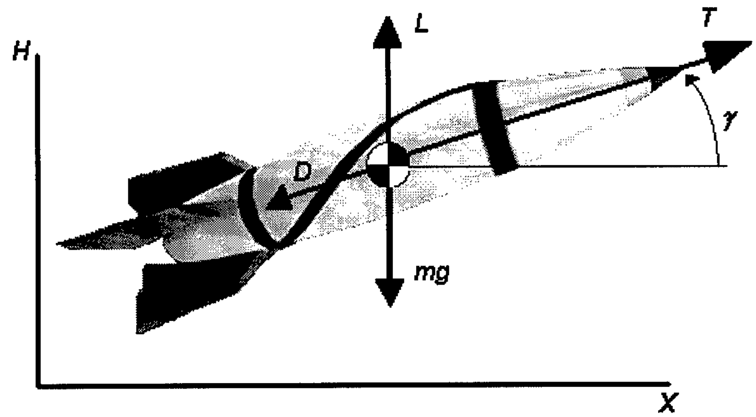

Figure 3-1. Forces acting on a rocket

The specific reason as to why it is preferred to treat the earth as being flat is that during the initial moments of launch, the vehicle is moving slowly as related to the rotating spherical earth. The components of the rockets velocity are small compared those of the earth's velocity. So in effect, the rocket is not countering the effect of the earth in the slightest. Thus the path of the rocket can be treated as if it were over a flat earth. As the components of the rocket's velocity increases, this simplification breaks down. The effects of the rotating spherical earth must then be taken into account. Also, at low altitudes such as those experienced during launch or reentry, it is more convenient to refer to coordinates such as altitude and horizontal distance flown when describing the flight path of a rocket. In Figure 3-1, **X** is the downrange distance flown, **H** is the altitude, and γ is the flight path angle. To begin describing the equations of motion for this rocket, a frame of reference must be chosen. Since the local horizon frame, or **h** frame, has its origin at the center of the earth, then the inertial origin for this research work will have its origin there as well. As the name implies, the **h** frame is local to the rocket, with $\mathbf{h}_1$ in the downrange direction, $\mathbf{h}_2$ in the local vertical or up direction, and $\mathbf{h}_3$

coming out of the page. Since the **h** frame is not an inertial frame, the angular velocity, ω^{hi}, must be taken into account so as to keep the rocket on the local vertical axis $\mathbf{h}_2$ as shown in Figure 3-2.

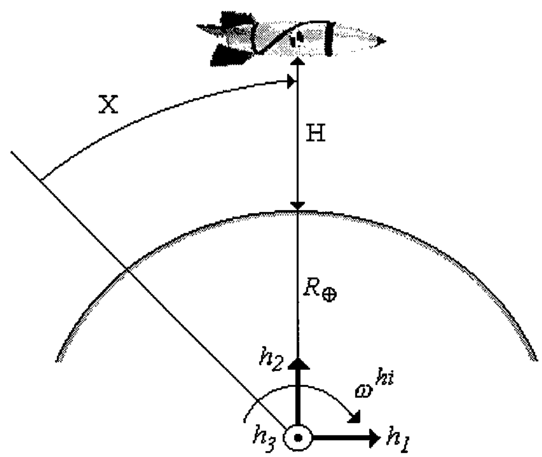

Figure 3-2. Local-horizon frame [39:20]

The position vector and angular velocity can now be written as

$$\mathbf{r} = (R_\oplus + H)\mathbf{h}_2 \tag{3-2}$$

and

$$\omega^{\mathbf{hi}} = -\frac{\dot{X}}{R_\oplus + H}\mathbf{h}_3 . \tag{3-3}$$

The minus sign comes from the right-handed rule for angular velocities and $R_\oplus$ is the radius of the earth [39:19]. Direct calculations of centripetal acceleration, which points back to the center of the earth, $\omega^{\mathbf{hi}}$ x ($\omega^{\mathbf{hi}}$ x **r**) yields

$$\omega^{\mathbf{hi}} \times (\omega^{\mathbf{hi}} \times \mathbf{r}) = -\frac{\dot{X}^2}{R_\oplus + H}\mathbf{h}_2 . \tag{3-4}$$

As explained previously this is the inertial acceleration component which, when taken to the other side of **F=ma,** becomes a *force* [39:217]. Expanding on what is shown in Figure 3-1, the downrange acceleration would be $\ddot{X}$ and the acceleration in the vertical direction would be $\ddot{H}$ but, as will be shown, it is more useful to describe the accelerations in their local components. Newton's second law as shown in equation (3-1) can now be further expressed as

$$\mathbf{T}+\mathbf{D}+\mathbf{L}-mg\mathbf{h}_2 = m\left(\ddot{X}\mathbf{h}_1 + \left\{\ddot{H} - \frac{\dot{X}^2}{R_\oplus + H}\right\}\mathbf{h}_2\right). \tag{3-5}$$

Taking the centripetal acceleration term to the other side of the equation yields

$$\mathbf{T}+\mathbf{D}+\mathbf{L}-m\left(g - \frac{\dot{X}^2}{R_\oplus + H}\right)\mathbf{h}_2 = m(\ddot{X}\mathbf{h}_1 + \ddot{H}\mathbf{h}_2)\,. \tag{3-6}$$

Referring back to Figure 3-1 the equations of motion, with velocity V aligned with thrust, become

$$\frac{dX}{dt} = V\cos\gamma \tag{3-7}$$

$$\frac{dH}{dt} = V\sin\gamma\,. \tag{3-8}$$

Since the launch model is based on the gravity-turn trajectory, the acceleration will not be broken down into its horizontal $\ddot{X}$, and vertical $\ddot{H}$ components. Keeping in mind that the local acceleration components of the rocket are described as being dV/dt along the vehicle axis parallel to the thrust vector, and Vdγ/dt, which is transverse to this axis, equations can be used to solve for Newton's second law. Specifically, Equations (3-7) and (3-8) can be used with Equation (3-6) to represent Newton's second law as,

$$\mathbf{T}-\mathbf{D}-\left(mg-\frac{m\dot{X}^{2}}{R_{\oplus}+H}\right)\sin\gamma=m\frac{dV}{dt} \tag{3-9}$$

$$-\left(mg-\frac{m\dot{X}^{2}}{R_{\oplus}+H}\right)\cos\gamma=mV\frac{d\gamma}{dt}. \tag{3-10}$$

But, this is only part of the solution; these equations of motion handle only the initial few moment of the ascent trajectory. They do not take into account the effects of a rotating spherical earth.

Before going on to the next section, where these basic equations of motion are expanded to include the effects of a rotating earth, a point made previously should be explained. The exception to the statement of how the velocity and thrust vectors are in constant alignment during the gravity turn trajectory will now be addressed. In equation (3-10), if the initial γ is 90°, then γ does not change, and the rocket would quickly revisit the spot from which it was launched. To prevent this unwanted re-visitation, soon after launch when the vehicle has cleared the tower, the vehicle is nudged away from its 90° attitude by a small change in γ, this is called the *pitch program* [39:218]. Once this occurs, as is apparent in equation (3-10), ${}^{d\gamma}\!/_{dt}$ is negative and the vehicle would begin to fall over. This is the basis of the gravity-turn trajectory. At high speed it is not the vehicle's axis that is being turned, as was stated this would lead to catastrophic transverse loads on the vehicle's structure, but its momentum vector. This rotation is performed by forces acting through the center of mass, so there is no torque on the booster and a relatively long period of time passes so as to transition from a γ of 90° to a γ of 0° for a circular orbit [39:218]. The trick of course is picking the initial value of γ such that at burnout, γ is 0°.

3.2.3 Gravity Turn Trajectory Refined.

The equations of motion generated in Section 3.2.2 work for the initial ascent portion of the launch trajectory, where drag is a relative minor influence because of the low initial velocity. Lift is non existent for much of the same reason. Now the equations of motion will be expanded to cover these forces, and will describe the time rate of change of radius, longitude, latitude, velocity, flight path angle, heading angle, and mass ($r, \theta, \phi, V, \gamma, \psi, m$). These additions, along with including the effects of a spherical earth on the trajectory model, will refine the model even further with the goal being to arrive at similar integrated values for the equations of motion as those supplied by Boeing, Defense Space Group. Section 3.2.3.1 will discuss the conventions used with respect to the spherical earth model, and Section 3.2.3.2 will derive the equations of motion. Finally, the equations of force will be explained in Section 3.2.3.3.

3.2.3.1 Conventions.

In Section 3.2.1 the local horizon frame, **h,** was defined for a rocket. To expand the equations of motion it is now necessary to define an earth fixed, earth centered inertial reference frame 0XYZ. This frame has 0 at the center of the gravitational field of a spherical earth with the 0XY plane in the equatorial plane of this same earth [34;21]. The 0XYZ reference frame is fixed with respect to the planet and is rotating with an angular velocity ω, which is constant and directed along the Z-axis Figure 3-3.

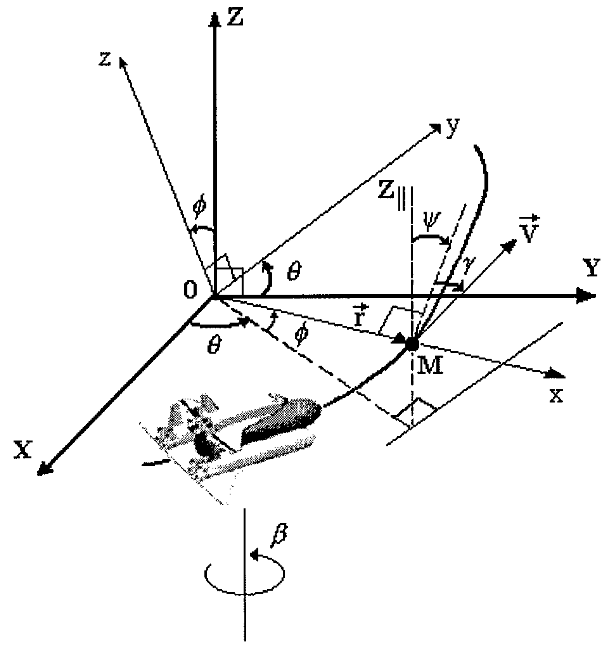

Figure 3-3. Coordinate Systems [34:22]

In Figure 3-3, with respect to the earth-fixed system 0XYZ, $\vec{\mathbf{r}}$ is the position vector indicating the radius to the point mass **M**, which represents the Shuttle-LFBB Launch System. Longitude θ, is measured from the X-axis, in the equatorial plane, with the positive direction being eastward. Latitude ϕ, is measured from the equatorial plane, along a meridian, positively northward [34:22]. With respect to the rotating coordinate system 0xyz, the **x**-axis is along the position vector $\vec{\mathbf{r}}$, the **y**-axis is in the equatorial plane positive toward the direction of motion and orthogonal to the **x**-axis, the **z**-axis completes the right-hand system. More concisely, the components of the rotating coordinate system can be described locally as; **x** up, **y** positive eastward, **z** north. The yaw angle is β, and is measured positive counter-clockwise from the Shuttle-LFBB local horizontal plane, the plane passing through the vehicle and orthogonal to the vector $\vec{\mathbf{r}}$, about the **x**-axis. The angle between the local horizontal plane and the velocity $\vec{\mathbf{V}}$ is γ. The angle γ, is positive when $\vec{\mathbf{V}}$ is above the horizontal plane. The heading angle ψ is the angle between a hypothetical axis $Z_{\parallel}$, which is parallel to the 0Z-axis and superimposed on the surface of the earth, and the projection of $\vec{\mathbf{V}}$ onto the horizontal plane. It is measured positive eastward from north, (negative x-axis via the right-hand rule). Not shown in Figure 3-3 is α this is known as the angle of attack (aoa), and is the angle measured between the velocity vector and the thrust vector. These conventions are summarized in the following table.

Table 3-1. Equations of Motion Conventions

Radius Vector	$\vec{\mathbf{r}}$
Longitude	θ
Latitude	ϕ
Flight Path Angle	γ
Heading Angle	ψ
Yaw	β
Velocity Vector	$\vec{\mathbf{V}}$
Angle of Attack	α

3.2.3.2 Basic Equations of Motion.

If **i**, **j**, and **k** are unit vectors in the rotating reference frame 0xyz respectively, then it can be shown that the position vector can be expressed in component form as

$$\vec{\mathbf{r}} = r\hat{\mathbf{i}}, \tag{3-11}$$

and with the help of Figure 3-3 and some spherical geometry [6:219], the velocity vector can be shown to be

$$\vec{\mathbf{V}} = (V \sin\gamma)\hat{\mathbf{i}} + (V \cos\gamma \sin\psi)\hat{\mathbf{j}} + (V \cos\gamma \cos\psi)\hat{\mathbf{k}}\,. \tag{3-12}$$

And, the angular velocity $\vec{\omega}$ can be represented by

$$\vec{\omega} = (\omega \sin\phi)\hat{\mathbf{i}} + (\omega \cos\phi)\hat{\mathbf{k}}\,. \tag{3-13}$$

With respect to the earth-fixed inertial system, Newton's second law **F=ma**, can be written in vector form as

$$\vec{\mathbf{F}} = m\frac{d\vec{\mathbf{V}}}{dt}, \tag{3-14}$$

and by using the operator relation [39:12]

$$\frac{{}^{i}d}{dt}(\;) = \frac{{}^{r}d}{dt}(\;) + \omega^{ri} \times (\;) \tag{3-15}$$

transformation of derivatives with respect to the rotating frame into derivatives with respect to the inertial frame is possible.

The time rate of change of the position vector with respect to the inertial frame can be expressed as

$$\frac{{}^{i}d\vec{\mathbf{r}}}{dt} = \frac{{}^{r}d\vec{\mathbf{r}}}{dt} + \vec{\omega} \times \vec{\mathbf{r}}\,, \tag{3-16}$$

which is an expression for the absolute velocity $\vec{\mathbf{V}}$. This then, using equation (3-15), and taking another time derivative gives

$$\frac{{}^{i}d\vec{\mathbf{V}}}{dt} = \frac{{}^{r}d}{dt}\left(\frac{{}^{r}d\vec{\mathbf{r}}}{dt} + \vec{\omega} \times \vec{\mathbf{r}}\right) + \omega^{ri} \times \left(\frac{{}^{r}d\vec{\mathbf{r}}}{dt} + \vec{\omega} \times \vec{\mathbf{r}}\right), \tag{3-17}$$

which expands to

$$\frac{{}^{i}d\vec{\mathbf{V}}}{dt} = \frac{{}^{r}d^{2}\vec{\mathbf{r}}}{dt^{2}} + 2\vec{\omega} \times \frac{{}^{r}d\vec{\mathbf{r}}}{dt} + \vec{\omega} \times (\vec{\omega} \times \vec{\mathbf{r}}), \tag{3-18}$$

and gives the absolute acceleration $d\vec{\mathbf{V}}/dt$.

Substituting equation (3-18) into equation (3-14) and, for convenience sake, changing the notation for the time derivative, Newton's second law becomes

$$m\frac{{}^{i}d\vec{\mathbf{V}}}{dt} = \vec{\mathbf{F}} - 2m\vec{\omega} \times \vec{\mathbf{V}} - m\vec{\omega} \times (\vec{\omega} \times \vec{\mathbf{r}}). \tag{3-19}$$

The velocity is with respect to the planet, and the time derivative is taken with respect to the earth-fixed axes [34:22]. The terms of equation (3-19) are summarized in Table 3-2.

Table 3-2. Newton's Second Law Equivalencies [34:22]

m is	Mass	(kilograms)
$\vec{\mathbf{V}}$ is	Absolute Velocity	(meters/second)
$\vec{\mathbf{F}}$ is	Σ all Forces (See Below)	(Newtons)
$\vec{\omega}$ is	Angular Velocity	(degrees/second)
$\vec{\mathbf{r}}$ is	Radius	(meters)

In reference to equation (3-19), $\vec{\omega} \times \vec{\mathbf{V}}$ and $\vec{\omega} \times (\vec{\omega} \times \vec{\mathbf{r}})$ can be derived from equations (3-12) and (3-13) as

$$\begin{aligned}\omega \times \vec{\mathbf{V}} = (\omega V \cos\gamma \cos\phi \sin\psi)\hat{\mathbf{i}} + \omega V(\sin\gamma \cos\phi + \cos\gamma \sin\phi \cos\psi)\hat{\mathbf{j}} - \\ \omega V(\cos\gamma \sin\phi \sin\psi)\hat{\mathbf{k}}\end{aligned}, \tag{3-20}$$

and

$$\vec{\omega} \times (\vec{\omega} \times \vec{\mathbf{r}}) = -(\omega^{2} r \cos^{2}\phi)\hat{\mathbf{i}} + (\omega^{2} r \sin\phi \cos\phi)\hat{\mathbf{k}}. \tag{3-21}$$

As for defining all forces and their components acting on the point mass **M** refer to Figure 3-4.

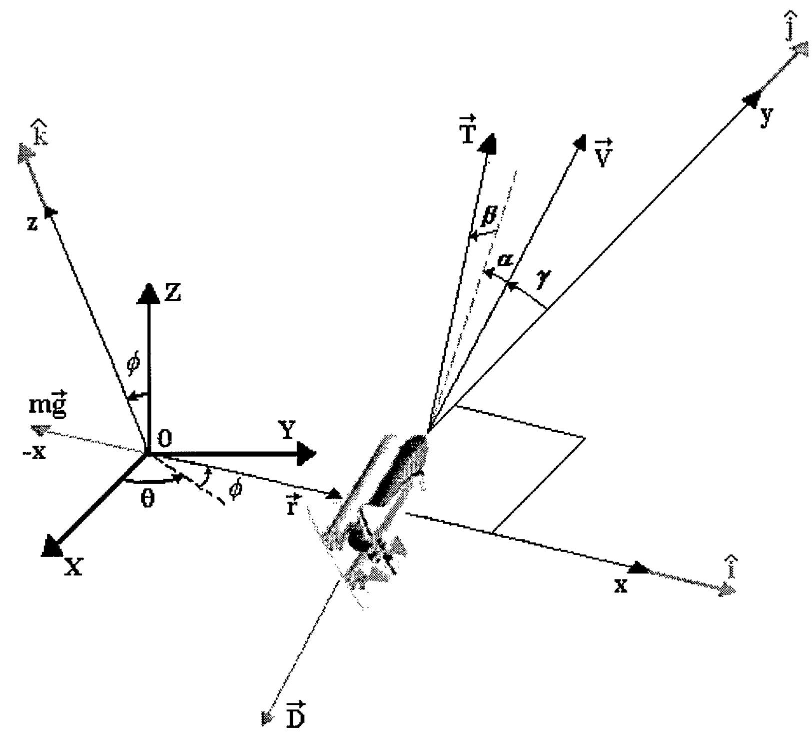

Figure 3-4. Force Components; Gravity, Aerodynamic and Thrust

In $\vec{\mathbf{F}}$, the gravity force is simply

$$m\vec{g} = -mg(r)\hat{\mathbf{i}}. \tag{3-22}$$

As for aerodynamic forces, this model considered the drag force $\vec{\mathbf{D}}$, which is opposite to the velocity vector $\vec{\mathbf{V}}$, and the lift force $\vec{\mathbf{L}}$, which acts transverse to the velocity vector. The equivalent drag force is

$$\vec{\mathbf{D}} = -\tfrac{1}{2} C_D A \sigma V^2 \hat{\mathbf{v}}_1. \tag{3-23}$$

The vector axis $\hat{\mathbf{v}}_1$ will be discussed below. The equivalent lift force is

$$\bar{\mathbf{L}} = \frac{1}{2} C_L A \sigma V^2 \hat{\mathbf{v}}_3. \tag{3-24}$$

Table 3-3 summarizes the drag and lift force components.

Table 3-3. Drag & Lift Components Defined [15]

C_D	Coefficient of Drag	= 1
C_L	Coefficient of Lift	= 1
A	Area (m^2)	= 249.90 m^2
σ	Atmospheric Density @ Altitude	= Kg/m^3 (Generated by Atm.for)
V^2	Velocity Squared (m^2/sec^2)	= m^2/sec^2 (Generated by Nominal.for)

The Shuttle's area A, was taken from Boeing data that was based on calculations derived from expected physical and performance attributes of the Shuttle-LFBB launch system [15].

The Coefficient of drag and lift are assumed to be 1 for the launch model as explained in this chapter's assumptions.

Initial velocity $V_{0,}$ was taken from data generated in section 3.2.5.1, and atmospheric density at altitude is calculated in the Fortran subroutine *Atm.for*" which is based on the atmospheric model found in the appendix of Regan and Anandarskarian [29]. The subroutine *Atm.fo*r, generates a value for density in kilograms per cubic meter given the altitude and ground level pressure.

Dynamic pressure S, is equivalent to

$$\mathrm{S} = -\frac{1}{2}\sigma V^2, \tag{3-25}$$

it has units Newtons per square meter. Dynamic pressure will be used in the trajectory model as a check to see if the atmosphere is too dense for any angle of attack α. Maneuvering the vehicle will become necessary as abort scenarios are initiated, hence necessitating an angle of attack. For reasons discussed in Section 3.2.1 air density must be at a minimum for maneuvers to be initiated in the event of an abort situation. Boeing cites 2 pounds per square foot as the maximum dynamic pressure where maneuvers could be successfully accomplished [14]. Using Equation (3-25) in Equations (3-23) and (3-24) yields

$$\vec{\mathbf{D}} = -C_D A S \hat{\mathbf{v}}_1 \tag{3-26}$$

and

$$\vec{\mathbf{L}} = C_L A S \hat{\mathbf{v}}_3 . \tag{3-27}$$

As for the different angles shown in Figure 3-4, γ or flight path angle is in the 0xy plane and is the angle between the local horizontal plane and the velocity vector. α or angle of attack, is the angle between the velocity vector and the thrust vector superimposed on the 0xy plane. Angle β is the amount of yaw, and is measured about the **x**-axis from the 0xy plane to the thrust vector.

The final force to be discussed from Figure 3-4 is the thrust force $\vec{\mathbf{T}}$. Since this model will handle the thrust force as a non-symmetric force, meaning the thrust vector will not always lie within the velocity-drag plane, it will be comprised of components along multiple axes. Consider a body reference frame v that is parallel to the rotating reference frame 0xyz and travels along at each point of the trajectory with the Shuttle-LFBB launch system. It is important to note which body axis corresponds to which rotating reference frame axis. Note that the $\mathbf{v}_1$-axis is parallel to the rotating reference frame's **y**-axis, the $\mathbf{v}_2$-axis is parallel to the **z**-axis, and the $\mathbf{v}_3$-axis is parallel to the **x**-axis. These relationships are shown in Figure 3-5.

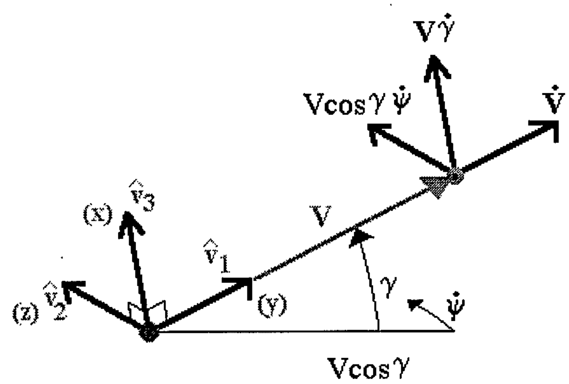

Figure 3-5. Body Reference Frame as it Relates to Rotating Frame 0xyz.

In Figure 3-5 some important points are made: any time rate of change to the flight path angle γ will affect any force component associated with the $\hat{v}_3$ axis. This relationship holds true for the heading angle as well, any time rate of change to ψ will affect components along the $\hat{v}_2$ axis, and finally any change to the magnitude of velocity will affect components along the $\hat{v}_1$ axis.

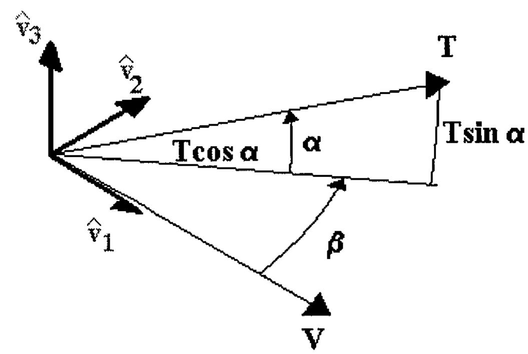

Figure 3-6. Body Reference Frame Thrust Components.

Figure 3-6 shows the components of thrust $\vec{\mathbf{T}}$, as they relate to the new body reference frame v. Thrust can now be fully expressed as

$$\vec{\mathbf{T}} = (T\cos\alpha\cos\beta)\hat{\mathbf{v}}_1 + (T\cos\alpha\sin\beta)\hat{\mathbf{v}}_2 + (T\sin\alpha)\hat{\mathbf{v}}_3 . \qquad (3\text{-}28)$$

Since the v reference frame axes and those of the rotating reference frame 0xyz are parallel, equation (3-28) can be written as

$$\vec{\mathbf{T}} = (T\sin\alpha)\hat{\mathbf{i}} + (T\cos\alpha\cos\beta)\hat{\mathbf{j}} + (T\cos\alpha\sin\beta)\hat{\mathbf{k}} . \qquad (3\text{-}29)$$

Having accomplished this, all vector terms in equation (3-19) have now been resolved into vector components along the rotating axis 0xyz. Now, the time derivatives of the vectors $\vec{\mathbf{r}}$ and $\vec{\mathbf{V}}$ must be taken with respect to the earth-fixed system 0XYZ. To do this the angular velocity vector $\tilde{\mathbf{U}}$ of the rotating axes must be evaluated. The 0xyz system results from the 0XYZ system by a rotation of θ about the positive Z-axis, and a rotation of ϕ about the negative Y-axis as seen in Figure 3-1 [34:24]. For the angular velocity of the rotating system 0xyz, this gives

$$\vec{\Omega} = \left(\sin\phi \frac{d\theta}{dt}\right)\hat{\mathbf{i}} - \left(\frac{d\phi}{dt}\right)\hat{\mathbf{j}} + \left(\cos\phi \frac{d\theta}{dt}\right)\hat{\mathbf{k}} \tag{3-30}$$

for the angular velocity. Using Poisson formulas [34:20] the time derivatives of $\vec{\mathbf{i}}$, $\vec{\mathbf{j}}$, and $\vec{\mathbf{k}}$ can be deduced as

$$\frac{d\vec{\mathbf{i}}}{dt} = \vec{\Omega} \times \vec{\mathbf{i}} = \left(\cos\phi \frac{d\theta}{dt}\right)\vec{\mathbf{j}} + \left(\frac{d\phi}{dt}\right)\vec{\mathbf{k}}, \tag{3-31}$$

$$\frac{d\vec{\mathbf{j}}}{dt} = \vec{\Omega} \times \vec{\mathbf{j}} = -\left(\cos\phi \frac{d\theta}{dt}\right)\vec{\mathbf{i}} + \left(\sin\phi \frac{d\theta}{dt}\right)\vec{\mathbf{k}}, \text{ and} \tag{3-32}$$

$$\frac{d\vec{\mathbf{k}}}{dt} = \vec{\Omega} \times \vec{\mathbf{k}} = -\left(\frac{d\phi}{dt}\right)\vec{\mathbf{i}} - \left(\sin\phi \frac{d\theta}{dt}\right)\vec{\mathbf{j}}. \tag{3-33}$$

Taking the time derivative of $\vec{\mathbf{r}}$ in equation (3-11) while using equation (3-31) for the time derivative of $\vec{\mathbf{i}}$ yields

$$\frac{d\vec{\mathbf{r}}}{dt} = \left(\frac{dr}{dt}\right)\hat{\mathbf{i}} + \left(r\cos\phi \frac{d\theta}{dt}\right)\hat{\mathbf{j}} + \left(r\frac{d\phi}{dt}\right)\hat{\mathbf{k}}. \tag{3-34}$$

Setting equation (3-34) equal to equation (3-12) and solving for individual $\vec{\mathbf{i}}$, $\vec{\mathbf{j}}$, and $\vec{\mathbf{k}}$ components along with some simplification, yields the first three equations of motion

$$\frac{dr}{dt} = V\sin\gamma, \tag{3-35}$$

$$\frac{d\theta}{dt} = \frac{V\cos\gamma\sin\psi}{r\cos\phi}, \tag{3-36}$$

$$\frac{d\phi}{dt} = \frac{V\cos\gamma\cos\psi}{r}. \tag{3-37}$$

These equations of motion are the kinematic equations, or the equations describing the motion of the Shuttle-LFBB model. With them the model arrives at values for the time rate of change of altitude r, longitude θ, and latitude ϕ [19:27]. These equations will be placed into a Fortran subroutine called Rhs.for from which incremental values will be calculated.

3.2.3.3 Equations of Force.

Finding these first three equations of motion completes the second step in developing a trajectory model, the first being section 3.2.1, which dealt with the gravity-turn portion of the initial launch. The third step involves deriving three other equations of motion that calculate values for the time rate of change of velocity $\vec{\mathbf{V}}$, flight path angle γ, and heading angle ψ. These equations of motion are derived by taking the basic vector equation (3-19) and substituting in equations (3-20), (3-21), (3-22), and (3-28), and then solving for the derivatives dV/dt, dγ/dt, and dψ/dt. This yields the following scalar equations:

$$\frac{dV}{dt} = \frac{T}{m}\cos\alpha\cos\beta - g\sin\gamma - \frac{\mathbf{C}_d As}{m} \tag{3-38}$$

$$\frac{d\gamma}{dt} = \frac{\left(-(g\cos\gamma) + \dfrac{V^2\cos\gamma}{r} + \dfrac{T}{m}\sin\alpha - 2\omega V\cos\phi\sin\psi + \dfrac{\mathbf{C}_l As}{m}\right)}{V} \tag{3-39}$$

$$\frac{d\psi}{dt} = \frac{\left(\dfrac{T}{(m\cos\gamma)}\cos\alpha\sin\beta - 2\omega V(-\sin\phi + \cos\phi\cos\psi\tan\gamma) + \dfrac{V^2}{r}\cos\gamma\sin\psi\tan\phi\right)}{V} \tag{3-40}$$

These three force equations [34:27] will complete the equations of motion that appear in the Fortran subroutine Rhs.for. This now gives a total of six equations of motion. Table 3-4 lists the terms of the equations of force.

Table 3-4. Force Equations of Motion Term Summary

T	Thrust (Newtons)	α	Angle of Attack (degrees)
m	Mass (kilograms)	β	Yaw (degrees)
r	Radius (meters)	γ	Flight Path Angle (degrees)
V	Velocity (meters/second)	ϕ	Latitude (degrees)
g	Gravity (meters/second^2)	θ	Longitude (degrees)
C_d	Coefficient of Drag	ω	Angular Velocity (deg/second)
A	Surface Area (meters^2)	ψ	Heading Angle (degrees)
s	Dynamic Pressure (N/meters^2)		

At this point all required parts for the baseline model have been derived, the next section will explain how the different components were combined by coding the model into a Fortran program called *Nominal.for*. This program will simulate the baseline case of a nominal launch, an abort derivative of this code was then made so that the performance characteristics of the Shuttle-LFBB launch system could be studied.

3.2.4 Launch Model Fortran Code Development.

Fortran 77 was chosen due to its powerful number crunching ability and it is also currently supported and utilized by Boeing in their LFBB code development [32].

Nominal.for is comprised of three sub-routines called *Atm*, *Rhs*, and *Haming*. *Atm* is a routine that calculates the atmospheric density for a particular altitude given an initial atmospheric pressure at height H_0. *Atm* is based on the atmospheric model of Regan and Anandarskarian [29], and the calculations of this routine are used, among other things, in calculating the dynamic pressure. Section 3.2.5.1 sets this initial pressure at 99621 Newtons per square meter, which is the atmospheric pressure at the top of the launch tower where the model begins. *Rhs* is the sub-routine that contains the six equations of motion that were generated previously in sections 3.2.3.2 and 3.2.3.3, these equations of motion provide the incremental values for the initial conditions as time is advanced, this process creates the simulated trajectory model. To get the incremental values that are added to the initial conditions, which generate the trajectory model, some method of numerical integration must be employed. *Haming* is the sub-routine that does just that, it handles the integration of the equations of motion contained in *Rhs*.

Almost any technique used to generate equations of motion will lead to the system of equations

$$\dot{\mathbf{x}} = \mathbf{f}(x,t)), \tag{3-41}$$

in first order form. Second order equations can be reformulated into first order form and implemented as in equation (3-41). The requirements for using any numerical integrator are to have a main program that sets up initial conditions, controls the input and output, and sets a time step. A subsequent requirement is to have a sub-routine that contains the actual equations of motion, as mentioned *Rhs* does this for this model. Given the state vector of the rocket, *Rhs* calculates the right hand sides of the equations of motion, hence the name *Rhs*. It is worthwhile to note that the units need not be the same on each element of the state vector. However, it is much safer numerically if all the state vector elements have the same characteristic order of magnitude [40:118]. This is the specific reason why dimensionless units were used throughout *Nominal.for* and its subroutines. Information for converting to dimensionless units can be found in Bate, Mueller, and

White's Fundamentals of Astrodynamics Appendix A [1;429]. Table 3-5 list the units that were used to convert the variables used throughout the different subroutines.

Table 3-5. Dimensionless Units and Their Values.

Time Unit, TU			=	806.8118744	seconds/TU
Distance Unit, DU	=	1 Earth Radii	=	6378145.0	meters/DU
Mass Unit, MU	=	1 Earth Mass	=	5.976 E+24	kilograms/MU
Angular Rotation, $\omega_{\oplus}$			=	0.0588336565	radians/TU
1 radian			=	57.2957795131	degrees
1 degree			=	0.0174532925199	radians

The next section discusses the methods employed, while executing the nominal model, to achieve the desired final conditions. Achieving the desired final conditions would verify that this was a valid baseline model of the Shuttle-LFBB launch system.

3.2.5 Launch Model Validation.

With the different components of the nominal model having been derived, it was possible to complete phase one of the thesis. Phase one provided a *baseline* model of a nominal 51.6° trajectory for the Shuttle-LFBB launch system by combining the information generated in Sections 3.2.2 and 3.2.3 into a Fortran program called *Nominal.for*. A copy of the code can be found in Appendix 1.

Before running *Nominal.for* for the first time, it is necessary to calculate the initial conditions that would be used to verify that this launch model behaves in a realistic manner. Also, the desired final conditions must be presented so that the results can be compared to Boeing's existing performance data in hopes of validating the nominal model. The next two sub-sections, 3.2.5.1 and 3.2.5.2, will explain how the initial conditions were derived as well as the final conditions the model hoped to achieve. A description of the steps taken for verification of the nominal model will follow in sub-section 3.2.5.3.

3.2.5.1 Gravity Turn Initial Conditions.

The definition of initial conditions begins with the understanding that this is a boundary-value problem, several conditions are known at each end of the trajectory. At the pitch-over point an initial altitude, H_0, must be calculated so that the vehicle will be clear of the tower prior to starting the pitch-program. For the Shuttle-LFBB launch system, Boeing gave a time of nine seconds as the amount of time necessary to clear the tower and begin the pitch-program [14]. All calculations to follow were based on the SI system except for Table 3-6 which includes some values from the original Boeing data in U.S. units for comparison.

Using the performance parameters depicted in Table 3-6, the following initial values were calculated assuming constant acceleration: For the initial mass at the beginning of the model, which takes place at launch plus nine seconds, M_0 was calculated to be

$$\mathbf{M}_0 = \mathbf{M}_{sl} - (\dot{m} * Time). \tag{3-42}$$

The mass flow rate $\dot{m}$, was derived from the propellant weight flow rate at sea level. Weight flow rate or $\dot{w}$, is a proportionality constant. This means no matter what the altitude, $\dot{w}$'s value remains constant. Thrust and Isp are varied so as to maintain $\dot{w}$'s value [22:8]. This relationship can be seen in the following equation that solves for $\dot{w}$,

$$\dot{w} = \frac{Thrust_{Tot-Vac} * PowerLevel}{Isp_{Tot-Vac}}. \tag{3-43}$$

Now

$$\dot{m} = \dot{w} * gravity_{SL} \tag{3-44}$$

therefore,

$$\mathbf{M}_0 = 2{,}134{,}000 - (9{,}470 * 9\,\sec s) \tag{3-45}$$

Table 3-6. Shuttle-LFBB Baseline Dual RS-76 w/RTLS Eliminated [13]

Mass	
Shuttle + ET + PayLoad	1,933,000 lbs or 877,000 Kg
LFBB (incl. 8 RS-76 Boosters)	2,773,000 lbs or 1,258,000 Kg
Total Launch Mass (SL)	**4,706,000 lbs or 2,135,000 Kg**
Performance	
Thrust Vaccum (Rated)	
Shuttle 3 Main Engs (492k lb ea @104.5%)	1,476,000 lbf or 6,569,000 N
LFBB RS-76 x 8 (750k lb ea)	6,000,000 lbf or 26,689,000 N
Total Vac	**7,476,000 lbf or 33,258,000 N**
Thrust Sea Level (Rated)	
SME (398.3k lb ea @104.5% PL)	1,194,000 lbf or 5,313,000 N
LFBB (649.8k lb ea @75% PL)	5 ,198,000 lbf or 23,123,000 N
Total SL	**6,392,000 lbf or 28,436,000 N**
Thrust @ L+9 secs (Interpolated Boeing Data)	
SME (399,500 lb ea. @104.5% PL)	1,198,500 lbf or 5,331,000 N
LFBB (651,487.5 lb ea. @75% PL)	5,211,900 lbf or 23,183,000 N
Total @ L+9 secs (Tower Clear)	**6,410,400 lbf or 28,514,000 N**
Isp Vaccum	
Shuttle Main Engines	453.2 secs
LFBB (RS-76 type)	340.5 secs
Weight Flow Rate, $\dot{w}$	
Total (Vac or SL)	**92,878 N/sec**
Mass Flow Rate, $\dot{m}$	
Total (Vac or SL)	**9,470 kg/sec**
Engine Nozzle Exit Area (in^2)	
Shuttle	6,405 (in^2) * 3 = 19,215 (in^2)
LFBB	6,818 (in^2) * 8 = 54,544 (in^2)
Atmospheric Pressure @ L+9 secs (Tower Clear)	
Interpolated value from Boeing Data	14.45 lb/in^2 or 99621 N/m^2

or

$$\mathbf{M}_0 = 2{,}049{,}400\,\text{kg at L} + 9\,\text{seconds}. \tag{3-46}$$

As for the initial atmospheric pressure P_0, using the interpolated value for LFBB thrust at altitude and the value for LFBB thrust in a vacuum from Table 3-6 in equation (3-47), it was possible to solve for P_0. Equation (3-47) shows how thrust for a given altitude was calculated based on the value of thrust in a vacuum, for a particular engine, and the atmospheric pressure experienced at that altitude [14].

$$\text{Thrust}_{@\,\text{Alt}} = \text{Thrust}_{\text{Vac}} - \left(\text{Pressure}_{@\,\text{Alt}} * \text{Engine Nozzle Exit Area}\right). \tag{3-47}$$

Rearranging equation (3-47) and solving for Pressure $_{@\,\text{Alt}}$ gives

$$\text{Pressure}_{@\,\text{Alt}} = \frac{\text{Thrust}_{\text{Vac}} - \text{Thrust}_{@\,\text{Alt}}}{\text{Engine Nozzle Exit Area}}. \tag{3-48}$$

So

$$\text{Pressure}_{@\,\text{Alt}} = \frac{750{,}000\,\text{lbf} - 651{,}500\,\text{lbf}}{6818\,\text{in}^2}, \tag{3-49}$$

or

$$\text{Pressure}_{@\,\text{Alt}} = 14.45\,{}^{\text{lb}}\!/_{\text{in}^2} \text{ or } 99621\,{}^{\text{N}}\!/_{\text{m}^2}. \tag{3-50}$$

This then is the pressure at the top of the launch tower or P_0. H_0, calculated using sea-level values, was

$$\text{H}_0 = {}^1\!/_2 * \mathit{Acceleration} * \mathit{Time}^2, \tag{3-51}$$

where

$$Acceleration = \frac{Thrust_{\text{Tot SL}}}{Mass_{\text{Tot SL}}} - gravity, \tag{3-52}$$

or

$$Acceleration = \frac{28{,}436{,}000\ Newtons}{2{,}135{,}000\ Kg} - 9.807\ {}^{m}\!/_{s}. \tag{3-53}$$

Therefore

$$Acceleration = 3.515\ {}^{m}\!/_{s^2}. \tag{3-54}$$

With the pitch-program beginning at launch +9 seconds, equation (3-51) gives an H_0 of

$$\mathrm{H}_0 = {}^{1}\!/_{2} * 3.515\,{}^{m}\!/_{s^2} * 9\sec s^2,\ \text{or} \tag{3-55}$$

$$\mathrm{H}_0 = 142\, meters. \tag{3-56}$$

This correlated with the documented NASA time of seven seconds to climb to an altitude of 122 meters [26], and the Boeing's interpolated value of 143 meters at L+9 seconds. The current Shuttle configuration has a slightly higher acceleration; this explains the difference between the values [2]. The LFBB with its slower acceleration needs the two extra seconds to build up enough altitude and velocity to ensure the tower/lightning arrestor is cleared by a substantial margin. V_0, the initial velocity, was calculated as

$$\mathrm{V}_0 = Acceleration * Time \tag{3-57}$$

$$\mathrm{V}_0 = 3.515\,{}^{m}\!/_{s^2} * 9\sec s \tag{3-58}$$

$$\mathrm{V}_0 = 31.635\ {}^{m}\!/_{s}. \tag{3-59}$$

Again, this is a close correlation to similar performance characteristics of the current Space Shuttle launch system and Boeing LFBB performance data [2:13]. Initial latitude, ϕ, and longitude, θ, for the Kennedy Space Center's Complex 39 pad B were taken off a

Garmin-III Global Positioning System (GPS) unit. The values were compared to the values used by Boeing for its model [13]. Mass is the gross mass of the Shuttle-LFBB launch system with a hypothetical 40,000 kg payload and all necessary fluids such as fuel, coolant, and hydraulics necessary for a typical mission.

The initial value for flight path angle γ, was arrived at by using an initial guess of 89.9°, a value slightly off from vertical, in the nominal trajectory model. Section 3.2.5 discusses the refinement of this in detail. Understanding that allowable East Coast launches fall within azimuths ranging from 35° northeast to 120° southeast for inclinations of 57° to 39° respectively [28], the initial azimuth or heading angle ψ was calculated using spherical trigonometry [4:128]. Like γ, this would be an approximation, using the spherical trigonometric equation (3-60) and Figure 3-7,

$$\cos(Inclination) = \cos(Latitude) * \sin(Azimuth) \text{ [4]}, \tag{3-60}$$

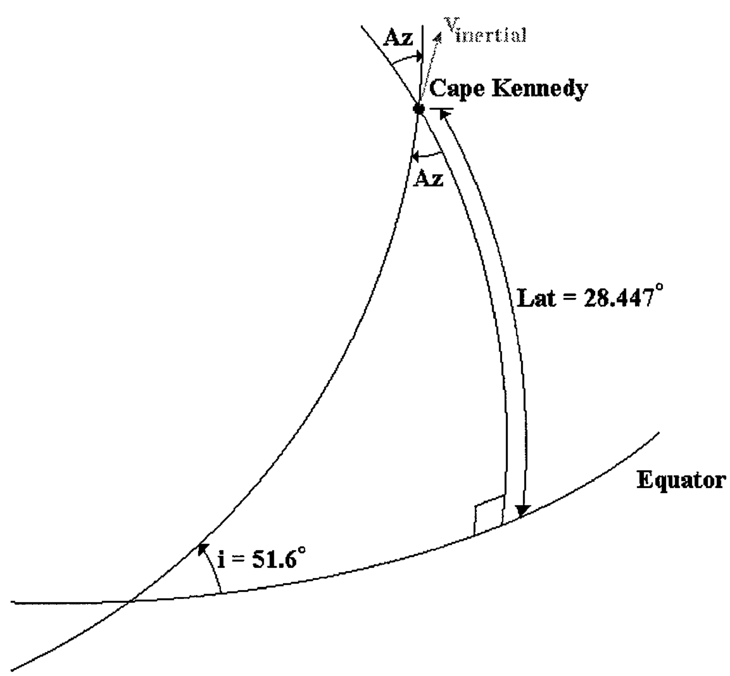

Figure 3-7. Initial Azimuth, ψ (Non-Inertial) Defined.

the initial value for ψ was calculated to be

$$Initial\ Azimuth = \arcsin\left(\frac{\cos(Inclination)}{\cos(Latitude)}\right) \tag{3-61}$$

$$Initial\ Azimuth = \arcsin\left(\frac{\cos(51.6)}{\cos(28.447)}\right) \tag{3-62}$$

$$Inital\ Azimuth = 44.946 \text{ degrees}. \tag{3-63}$$

The nominal model was initialized with this initial value for ψ. The model then calculated the final inclination based on an inertial ψ, inertial in the sense that it included the effects of angular velocity ω, from the rotating earth. The inertial ψ is based on Figure 3-8 and was calculated with Equations (3-64) and (3-65).

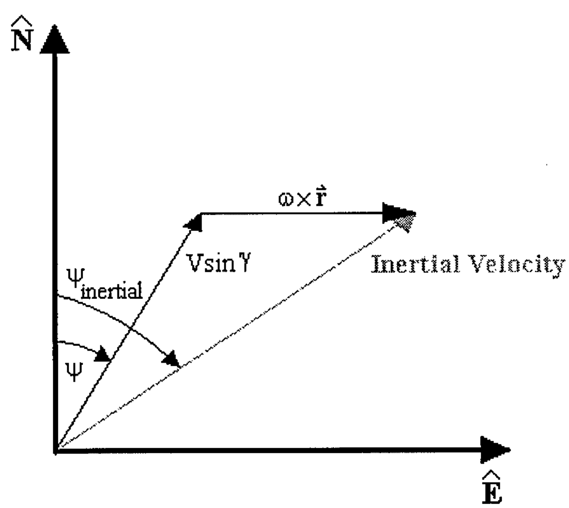

Figure 3-8. Inertial Azimuth.

$$\mathbf{V}_{Inertial} = (V \sin\gamma \cos\psi)\hat{\mathbf{N}} + ((\omega \times \vec{\mathbf{r}}) + V \sin\gamma \sin\psi)\hat{\mathbf{E}}. \tag{3-64}$$

$$\psi_{Inertial} = \arctan\left(\frac{(\omega \times \vec{\mathbf{r}}) + V \sin\gamma \sin\psi}{V \sin\gamma \cos\psi}\right). \tag{3-65}$$

The iterations of different ψ values for the model will be discussed in Section 3.2.5, all initial conditions are listed in Table 3-7.

Table 3-7. Launch Model Initial Conditions

Mass	M_0	2,049,000	kilograms
Atmospheric Pressure	P_0	99621	Newtons/meter^2
Altitude	H_0	142	meters
East Longitude	θ	279.395	degrees
Latitude	ϕ	28.447	degrees
Velocity	V_0	32	meters/second
Flight Path Angle	γ	89.881	degrees
Heading Angle	ψ	44.625	degrees

3.2.5.2 Gravity Turn Final Conditions.

At the other end of the trajectory the altitude for a particular circular orbit is usually known, H_f, as well as the burnout velocity, V_f, needed to attain it. But, the purpose of this model is to look at abort situations that could occur with the LFBBs still attached and thrusting prior to attaining the desired final orbit parameters. So, disregarding H_f and V_f, the final conditions that are important for this model are the state vector components: altitude, longitude, latitude, velocity, flight path angle, heading, mass, and inclination or r, θ, ϕ, V, γ, ψ, m, and i respectively. It is important that these components match as close as possible the values provided by Boeing for a nominal

LFBB separation at launch plus 135.1 seconds in order to validate the model. Desired Boeing state components are shown in Table 3-8 [13].

Table 3-8. LFBB State Vector Components at Nominal Separation L+135.1 sec's [13].

Altitude	r	48,826	meters
East Longitude	θ	279.8	degrees
Latitude	ϕ	28.8	degrees
Velocity	V	1728	meters/second
Flight Path Angle	γ	26	degrees
Heading Angle	ψ	43	degrees
Mass	m	902,000	kilograms
Inclination	i	51.6	degrees

3.2.5.3 Verification.

Having an understanding of the initial and final conditions the model has to work with, it was now possible to proceed with the verification of the nominal model. In completing this verification it was necessary to perform iterations where different values of γ and ψ were added to the input file *nom.in*, which would be read in by *Nominal.for*. The code was iterated so as to reach the final conditions presented in Section 3.2.5.2, and shown in Table 3-8.

Once the final conditions for γ and ψ were attained, the output files *nom_dyn.dat* and *nom_H_X.dat*, which recorded dynamic pressure and altitude/range data respectively, were compared to similar Boeing performance data for the Shuttle-LFBB launch system. The results of both output files are displayed in Figure 3-9 and Figure 3-10.

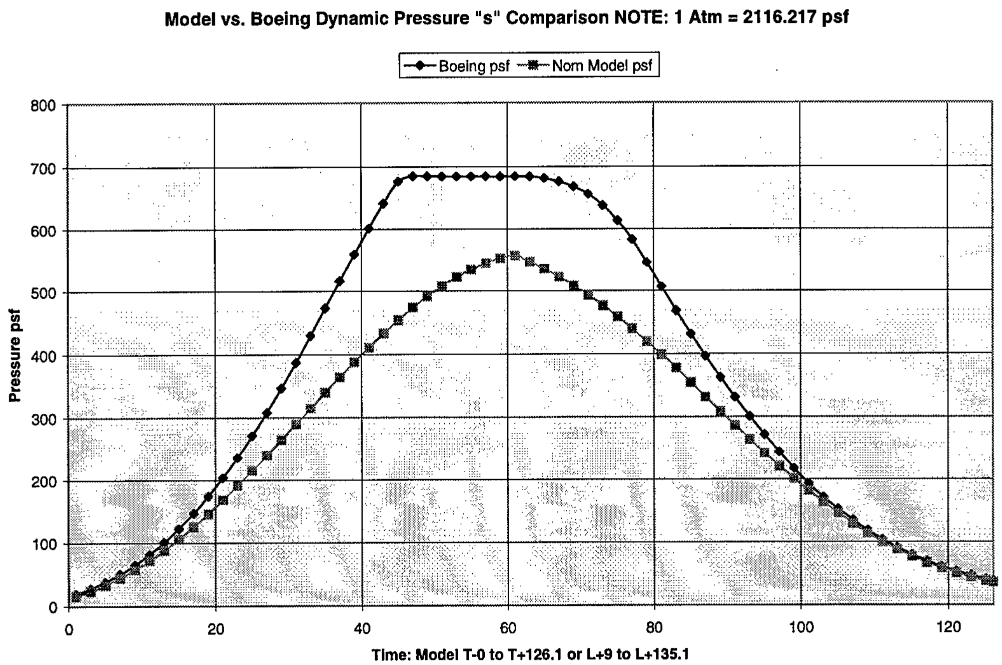

Figure 3-9. Boeing vs. Nominal Model: Dynamic Pressure "s" Curves.

As can be seen in Figure 3-9, the percent error between the peaks of the two corresponding curves at the 60-second point can easily be calculated. At the 60-second point the Boeing value for dynamic pressure is 684 psf, while the nominal model gives a value of 557 psf. The percent error for dynamic pressure is then

$$\%error_s = \frac{(684-557)}{684} \tag{3-66}$$

or,

$$\%error_s = 18\%\,. \tag{3-67}$$

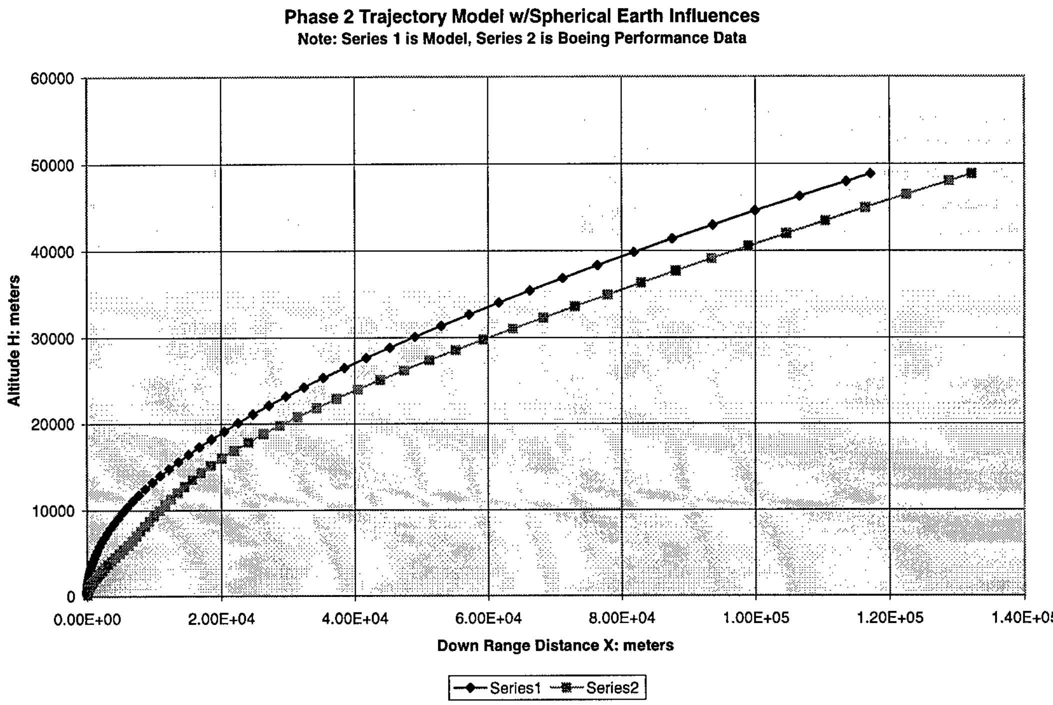

Figure 3-10. Boeing vs. Nominal Model: Trajectory Comparison.

In Figure 3-10, the percent error between the two curves for Altitude vs. Downrange distance at the 60-km downrange point was

$$\%error_{Alt/Dis} = \frac{(33-29.8)}{33} \tag{3-68}$$

or,

$$\%error_{Alt/Dis} = 9.7\% \,. \tag{3-69}$$

After consulting with Boeing and discussing the percent error results obtained from the nominal trajectory model for *dynamic pressure* and *Altitude vs. Downrange distance*, the level of agreement represented by these equations was considered acceptable. Boeing verified that the nominal model produced results comparable to the expected performance capabilities of the Shuttle-LFBB launch system [13]. This verification validated the trajectory model and showed that this model would be a good baseline representation of a nominal launch trajectory.

In concluding phase one, the trajectory model outputs to *nom_st_v.dat,* the state vector for each corresponding second of the Shuttle-LFBB nominal launch trajectory. This output covers the period from liftoff until the LFBB separation time of 135.1 seconds. Table 3-9 shows the output files and the data stored in each for the nominal trajectory model. At this point, phase two was entered into.

Table 3-9. Nominal Model Data Files & Information Stored.

DATA FILE	DATA STORED
Nom_dyn.dat	Time, Altitude, s (Dynamic Pressure)
Nom_h_x.dat	Time, Altitude (H), Downrange Distance (X)
Nom_st_v.dat	Time, 7 Element State Vector (r, θ, ϕ, v, γ, ψ, m), t0 & tf (Initial & Final Times), Nstp & Nskp (Integration Steps)

3.3 Phase 2: Abort Model

Phase two dealt with the modifications that would be made to the nominal trajectory model so that an abort version could be created. This abort trajectory model would simulate aborts at different times along the nominal trajectory. The abort model would then solve for the proper combinations of thrust, α, and β that would allow for successful abort landings to the East Coast. Chapter 4 discusses the results, which were collected from the abort model.

3.3.1 Assumptions.

For the abort model, some of the assumptions made for the nominal model still held. The first assumption dealt with the modeling of the atmosphere. Again, the atmosphere was thought to behave exactly as depicted in the model atmosphere of Regan and Anandarskarian [29]. The majority of the atmosphere was thought to exist below 50 km. Another assumption seen in the nominal model dealt with the rotating earth. Again the coriolis acceleration term, $2\omega V$, was retained but the higher order terms were neglected. The last assumption to be made, before modifications to the nominal model are discussed, concerns the theoretical modeling of winged vehicles through the atmosphere at hypersonic speeds. This aerodynamic model of the Orbiter is called the Rarefied-Flow Shuttle Aerodynamics Flight Model. It is based on data collected over 12 Shuttle re-entries. This model provides C_L and C_D values for varying altitudes, speeds, and is a function of α. Even though this is a highly theoretical area, the data provided by this model was based on actual data collected from the Shuttle during re-entry. This model is the best approximation of how a winged vehicle reacts when traveling through the atmosphere at hypersonic speeds. The Rarefied-Flow model will be discussed in the next section as well as during the discussion of how dynamic values for C_D and C_L were derived.

3.3.2 Nominal Model Modifications.

The Coefficients of drag and lift are generated by a subroutine called *Aero.for*. *Aero.for* calculates the two coefficients for a given angle of attack and a given Knudsen number that occur for a specific altitude. The Knudsen number is the ratio of mean free path to the mean aerodynamic chord (MAC). Section 2.4.3 discusses in detail the use of the *Rarefied-Flow* aerodynamic model, and how it involves the use of this Knudsen number. In short, the rarefied-flow is the transition region between free molecular flow and hypersonic continuum. This transition region occurs between 60 and 160 km, which is the primary region where the Shuttle operates [5:550]. With this being the case, it will be necessary to use the empirical equations that relate to this region. The equations are split into three parts: *hypersonic continuum*, *free molecular flow*, and a *bridging formula*. All three segments calculate C_N and C_A, which are aerodynamic coefficients in the normal and axial direction. Both C_N and C_A are functions of angle of attack α, and can be easily converted into C_L and C_D.

The hypersonic continuum equations for the normal and axial coefficients, as a function of angle of attack, are

$$C_{Nc,\alpha} = -9.25704\times10^{-5}\alpha^2 + 5.23808\times10^{-2}\alpha - 0.839782, \tag{3-70}$$

and

$$C_{Ac,\alpha} = 5.86689\times10^{-7}\alpha^3 - 6.72027\times10^{-5}\alpha^2 + 3.32044\times10^{-3}\alpha - 0.0086314. \tag{3-71}$$

These equations are for the Shuttle at an angle of attack envelope of 35 $<\alpha<$45 deg.

The free molecular flow equations for C_N and C_A as a function of α are:

$$C_{Nf,\alpha} = -7.16528\times10^{-6}\alpha^3 + 9.66197\times10^{-4}\alpha^2 + 9.18422\times10^{-3}\alpha - 1.58739\times10^{-3} \tag{3-72}$$

$$C_{Af,\alpha} = -1.17117\times10^{-5}\alpha^3 + 5.92205\times10^{-4}\alpha^2 + 0.0164864\alpha + 0.751105. \tag{3-73}$$

These equations are for the Shuttle at an angle of attack envelope of 0 $<\alpha<$60 deg.

Bridging the hypersonic continuum to the free-molecule flow regime is accomplished by use of the following *bridging formulae*:

$$\overline{C}_N = \exp\left[-0.29981(1.3849 - \log_{10} Kn)^{1.7128}\right] \qquad (3\text{-}74)$$

if $\log_{10} Kn < 1.3849$,

otherwise $\overline{C}_N = 1.0$.

$$\overline{C}_A = \exp\left[-0.2262(1.2042 - \log_{10} Kn)^{1.8410}\right]. \qquad (3\text{-}75)$$

if $\log_{10} Kn < 1.2042$,

otherwise $\overline{C}_A = 1.0$.

By using Equations (3-70) through (3-75), the re-entry aerodynamic coefficients can be calculated. Equations (3-76) and (3-77) show this relationship.

$$C_N = C_{Nc} + (C_{Nf} - C_{Nc})\overline{C}_N. \qquad (3\text{-}76)$$

$$C_A = C_{Ac} + (C_{Af} - C_{Ac})\overline{C}_A. \qquad (3\text{-}77)$$

Now it is possible to convert these aerodynamic coefficients into C_L and C_D. Equations (3-78) and (3-79) show how this is accomplished via the α of the orbiter, which is in degrees.

$$C_L = -C_A Sin(\alpha) + C_N Cos(\alpha). \qquad (3\text{-}78)$$

$$C_D = C_A Cos(\alpha) + C_N Sin(\alpha). \qquad (3\text{-}79)$$

The coefficient functions presented in Equations (3-70) through (3-77) were derived from curves fitted to Shuttle data that was collected from STS-61C [5:552].

3.3.3 Abort Model Fortran Code Additions.

Since the abort trajectory model is an expansion of the nominal trajectory model, the Fortran code was simply copied to an abort version with a single addition being made. The addition centered on the previous section's discussion of the Rarefied-Flow model, and the derivation of the C_D and C_L coefficients. The subroutine *Aero.for* was added to the abort model's Fortran code. A call to this subroutine would pass values for α and the *Kn*, and in return *Aero.for* would provide values for C_D and C_L for a given α and altitude.

3.4 Summary

This concludes phase one and phase two of the discussion on the methodology employed to create both the nominal and abort trajectory models. This chapter discussed the methods for developing a nominal trajectory model, which would simulate a 51.6° inclination launch up the East Coast of the United States. Output from this model included the state vector for each second of the trajectory. This information would then be used as input to the abort trajectory model.

This Chapter discussed the modeling of the initial portion of the launch trajectory as a gravity turn trajectory so that the thrust vector could be kept aligned with the velocity vector, and thus avoid potentially disastrous transverse aerodynamic forces to the launch structure. The gravity turn trajectory was then refined by the inclusion of effects from a rotating spherical earth. In anticipation of using the nominal model as the core structure for the abort model, routines were added for varying the values of C_D and C_L, and the number of equations of motion were increased to six. These additional equations of motion allow for the variation of thrust, α, and β, which in turn allow for solving for specific values of dV, $d\gamma$, and $d\psi$. In solving for specific values of dV, $d\gamma$, and $d\psi$, the abort model will generate trajectories that could potentially lead to successful abort landings along the East Coast.

The next chapter will take the abort model generated in this chapter and will run various abort scenarios through it. Solutions to the equations of motion will be solved for by the use of control variables. These control variables will directly affect how thrust, α, and β are varied. The primary purpose of the next chapter will be the elimination of the need for the RTLS abort mode when the Shuttle experiences a single SSME out abort scenario. The secondary purpose will be to see if an abort trajectory exist that will allow for quicker abort landings than what the TAL abort mode currently offers.

4. Analysis

4.1 Introduction

This chapter focuses on the results obtained by the abort trajectory program. It analyzed the end conditions to see if a landing facility could successfully be reached during the single SSME out abort scenario. As defined in Section 1.9 of this thesis, meeting the end conditions would indicate a successful abort landing had been accomplished.

The abort model tested two scenarios. In the first, an abort was simulated to have occurred just one second after clearing the launch tower. The second abort scenario was set to initiate at launch plus 119 seconds (L+119), just prior to TAL availability. In both cases, the initial state vector for initiating the abort was obtained from the nominal trajectory run discussed in Section 3.2.5. The nominal run simulated a typical 51.6° inclination launch with state vectors for every second of the nominal trajectory being stored in the file *nom_st_v.dat*. The collected state vectors covered the ascent period from liftoff until LFBB separation at L+135 seconds. To prevent confusion it should be pointed out that both the nominal and abort models started the simulation at L+9 seconds, at the top of the launch tower, and ran for 126 seconds until LFBB separation at L+135. The reason for this initial start occurring at L+9 seconds was due to the gravity turn portion of the initial ascent not functioning properly with an initial velocity of 0. To continue, the state vector for the one-second-abort scenario, derived from the nominal case, can be seen in Table 4-1. Table 4-2 contains the state vector for the 119-second abort scenario as well.

Table 4-1. 1 Second Abort Initiation State Vector

1.000027603394376	4.876375247312332	4.993081693153024E-001
4.521228749522317E-003	1.566322763194498	8.325507229168753E-001
3.413484164565641E-019		
1.240429996817460E-003	1.562941795990000E-001	
126.000000	30.000000	

Table 4-2. 119 Second Abort Initiation State Vector

1.005668904744045	4.879500322815867	5.024450999202795E-001
1.501080746434611E-001	6.442927216051213E-001	7.180002122034157E-001
1.683184621520560E-019		
1.364472996499195E-001	7.43667785562000E-001	
245.000000	50.000000	

Table 4-3 shows what state vector values are represented by the numbers seen in Table 4-1 and Table 4-2. The state vectors represented by these files would become the input values for the abort trajectory model. Most of the variables in Table 4-3 are self-explanatory. The *initial* and *final times*, along with *Nstp* would be used to calculate the time step of the trajectory model. The time step would dictate how often the model would calculate data for each trajectory point. For the two abort scenarios addressed by this research the time step was set to two, this would allow the model to generate a trajectory point every two seconds. *Nstp* and *Nskp* are used in the integration routine *Haming* where they calculate the range of integration.

Table 4-3. State Vector Dimensionless Represented Values

Radius from earth's Center	Longitude	Latitude
Velocity	Flight Path Angle	Azimuth Angle
Mass		
Initial Time of State Vector	Final Time of State Vector	
Nstp	Nskp	

4.2 Control Variables

To achieve the desired trajectory end conditions, control variables were used. Equations (3-38), (3-39), (3-40) were manipulated to solve for *Thrust*, α, and β. The control variables either represent these terms directly, or through surrogates so that desired results for the equations of motion were reached. Equations (4-1) through (4-6) show how *T*, α, and β were derived. The angle of attack α was represented by

$$\alpha = \tan^{-1}\left[\cos(\beta)\left(\frac{mg\cos(\gamma)r - mV^2\cos(\gamma) + 2m\omega V\cos(\phi)\sin(\psi)r + m\text{dgam}r + C_L Asr}{rmg\sin(\gamma) + rC_D As + mr\text{dV}}\right)\right], \tag{4-1}$$

while the yaw angle was

$$\beta = \tan^{-1}\left[m\left(\frac{-2\omega Vr\cos(\phi)\sin(\phi)\cos(\gamma) + 2\omega Vr\cos(\phi)^2\cos(\psi)\sin(\gamma) - V^2\cos(\gamma)^2\sin(\psi)\sin(\phi) + \cos(\gamma)\text{dpsi}r\cos(\phi)}{r\cos(\phi)mg\sin(\gamma) + r\cos(\phi)C_D As + r\cos(\phi)m\text{dV})}\right)\right]. \tag{4-2}$$

Solving for *Thrust* involves finding its magnitude as express by the components *A*, *B*, and *C*. This was calculated as,

$$Thrust = \sqrt{A^2 + B^2 + C^2} \tag{4-3}$$

where,

$$A = \left(mg\sin(\gamma) + C_D As + m\text{dV}\right) \tag{4-4}$$

$$B = \left(mg\cos(\gamma) - \frac{mV^2\cos(\gamma)}{r} + m2\omega V\cos(\phi)\sin(\psi) + m\text{dgamV} + \text{C}_\text{L}\text{As}\right) \tag{4-5}$$

$$C = \left[\left[\left(m\cos(\lambda)2\omega V\right)\left(-\sin(\phi) + \cos(\phi)\cos(\psi)\tan(\gamma)\right)\right] - m\frac{V^2}{r}\cos(\gamma)^2\sin(\psi)\tan(\phi) + m\cos(\gamma)\text{dpsiV}\right]. \tag{4-6}$$

Table 4-4 lists the control variables and the ranges used to manage the Shuttle's trajectory.

Table 4-4. Abort Model Control Variables and Effective Ranges.

CONTROL VARIABLE	EFFECTIVE RANGE OF VALUES
maxT ($\dot{V}$ Increments of LFBB/Shuttle Combo)	0.0 to 1.0244 Max Q = 750 psf at $\dot{V}$ of 1.0244
num (LFBB/Shuttle Throttle Increments)	2 to 40
dv_max ($\dot{V}$ Increments Shuttle Only)	0.1 to 1.5 SSME Capability Exceeded if $\dot{V}$ >1.5
nnn_stp (Shuttle-only Throttle Increments)	2 to 40
d_gam	(0.88 to 1.1) * (-1° per Time Unit)
d_psi	(0.05 to 0.54) *(-1° per Time Unit)
Δdgam	(-0.5 to-49.9) * (d_gam)
γdot -> 0	0 to –15° (γ)
α, Angle of Attack	0 to 45° (Defined)

The control variables *maxT*, *num*, *dv_max*, and *nnn_stp* all dealt with the simulation of dynamic throttling. Specifically, *maxT* and *num* dealt with the period when the LFBB and Shuttle would be attached together; it would represent the combined $\dot{V}$ the engines could provide. When dynamic pressure was low enough to allow an angle of attack or a change in yaw, the value of $\dot{V}$ given by *maxT* and *num* would be used in equations that solved for specific values of α, β, and thrust. These values, which give the desired $\dot{V}$, would in turn be used in the equations of motion to achieve the desired affect. The variable *num*, would be used to calculate the number of increments in the throttling profile, it would be user defined in the input file *abort.in*. The control variables *dv_max* and *nnn_stp* were the equivalent values for the time after the LFBB had separated and the Shuttle was thrusting on its own. Loops would be created in the model's code to evaluate each and every possible combination of LFBB/Shuttle and Shuttle-only thrust levels.

The controls *d_gam* and *d_psi*, were each used to correct the trajectory's path. Both are modifiers in that they are multiplied with the value (-14.0815236524d0). This dimensionless term represents a change of one degree per time unit. For example, if the heading must be corrected by two degrees over some period of time then *d_psi* would be assigned the value of 2.d0 and multiplied by this constant. Both *d_gam* and *d_psi* find their ways into the equations for α, β, and thrust in order to solve for specific values of dV , $d\gamma$, and $d\psi$, which are then used in the equations of motion.

The control variable *Δdgam* would be used to flip the sign and modify the magnitude of the flight path angle γ. This control would be activated by γ either going to zero or some predefined angle. *Δdgam*'s purpose was to modify the magnitude that γ was changing by and in what direction.

The variable *γdot-> 0* is more of a description of an event that would trigger other control variables such as *Δdgam*. As just stated above, if it was desired to flip the sign or affect the magnitude of γ, then when *γdot* went to 0 the action would take place. *γdot->*

could signify the achievement of different ranges of angles, for the purpose of this research it represented 0 to -15° with respect to γ.

The final control used in the abort model was angle of attack or α. Depending on how this control was used, small or large effects on the trajectory could be created. It could be used to slow or speedup the velocity of the orbiter depending if more or less range was needed to make an intended target.

The control variables were varied until values that approximated the TAEM end conditions were achieved. The first iterations of the abort program set a flag to limit the amount of output data. End condition data was sent to a file and analyzed to see if favorable conditions had been met. Also, any trends were noted that could aid in understanding of which control variables should be adjusted. Once desired end conditions were obtained, the abort model was executed a final time with a flag set for maximum data output. A discussion of the process involved with evaluating the effects of the various control variables and the ultimate discovery of the solutions to the abort trajectories will be discussed next.

4.3 Test Procedures and Evaluation

4.3.1 Method.

The first step in deciding how an abort landing might be best attempted was getting an idea of the specific elements that could have an adverse affect on the vehicle when the trajectory started to change. It became clear that one of the greatest influences would stem from dynamic pressure. Not until the dynamic pressure was low enough could the trajectory of the vehicle be altered. The vehicle would experience destructive forces if allowed to change its trajectory before dynamic pressures had dropped to low enough levels. Boeing stated 2 psf as being the level of dynamic pressure at which the vehicle could begin altering its trajectory without concern from lateral forces [14]. As for the other end of the trajectory, when the Shuttle would be making its approach to a landing site, dynamic pressure would again be a matter of concern. Here, dynamic pressure drives the time when the ET must separate. If allowed to get too large, the aerodynamic forces would drive the ET back into the orbiter. NASA resources quoted ET separation occurring at 2 psf, but for anomalous situations a *high rate separation* could be initiated with the dynamic pressure going as high as 9 psf [3;9;38]. Also, just as a maximum dynamic pressure or *max Q* is experienced during ascent, it is also experienced during landing. For ascent, the Shuttle is certified for 375 psf, but NASA Officials stated that theoretically, the Shuttle could sustain 800 psf. For reentry, NASA officials further stated that a *max Q* ranging from 375 to 750 psf could be feasible [3].

Another key element that had to be tracked during the entire abort trajectory was *g- force*, or the force of gravitational acceleration experience by the Shuttle and its crew. At any point in the trajectory, this value could not exceed 5 times the earth's gravitational acceleration, or 5 g's. The two critical times that g's approach dangerous levels were during launch and glide slope acquisition. The pull-up from the modified skip reentry maneuver produces substantial g's when reentering the atmosphere and attempting acquisition of the glide slope.

The next consideration was what would the pilot do if he had flight control and a SSME failed? With limited pilot experience the researcher concluded the first instinctive reaction would be to throttle up to compensate for the loss of power. Upon investigating this method, it became apparent that the LFBB could loft the Shuttle to a high enough altitude, but the LFBB would have used up its booster propellant in the process. If this occurred, then the LFBB could not further help shape the trajectory. Also, this may leave the Shuttle in an unrecoverable situation.

Since the purpose of this research was to understand how the performance capabilities of the LFBBs could aid in the successful accomplishment of an abort landing, another method was developed. As stated in Section 1.9, the initial approach to finding solutions for East Coast abort landings would entail flying the Shuttle like a helicopter. This would mean thrusting vertically to maintain altitude, while using the LFBB's throttling capability to shape the trajectory to the desired final conditions. The researcher felt the main contributor to successful abort trajectories would be the thrust provided by the LFBBs. With this in mind, LFBB throttling would have to be incorporated into the model. As a side note, a benefit of this research was the identification of the maximum $\dot{V}$, or velocity's time rate of change, for which the equation for thrust could solve. This was important not only because it would ensure that *max Q* stayed below the 750 psf maximum, but also because it would limit the thrust levels and keep them to realistic performance values. The maximum dimensionless value for $\dot{V}$ during ascent would be 1.0244. Since the dimensionless quantity for acceleration is one g, this then represents 1.0244 g's. This is the maximum acceleration the vehicle can attain during ascent without violating the 750 psf *max Q* constraint.

Using these findings, a routine was developed in Fortran to evaluate maximum thrust after an abort was declared. Values for maximum and minimum thrust, and weight flow rate $\dot{\omega}$, appear in Table 4-5. A complete listing of the abort model's Fortran code can be found in Appendix 2.

Table 4-5. Min & Max Values for Thrust and $\dot{\omega}$ for 1 SSME Out Scenario.

2xSSME & 8xLFBB	**Max Thrust**	**Min Thrust**	**Max $\dot{\omega}$**	**Min $\dot{\omega}$**
Shuttle/LFBB lbf	8.9846 E6	4.5919 E6	25.6672E3lbf/s	13.1416E3lfb/s
Newtons	39.9653 E6	20.4257 E6	114.173E3 N/s	58.4568E3 N/s
Dimensionless	6.8253 E-19	3.4883 E-19	1.5732 E-18	8.0547 E-19
Shuttle-Only lbf	9.8456 E5	4.7108 E5	2.1724 E3 lbf/s	1.0394 E3 lbf/s
Newtons	4.3795 E6	2.0955 E6	9.6632 E3 N/s	4.6235 E3 N/s
Dimensionless	0.7479 E-19	0.3579 E-19	1.3315 E-19	0.6371 E-19

Two other loops were added to simulate the throttling that would occur. The first loop would simulate the throttling down of the LFBB/Shuttle combined thrust that occurs prior to LFBB separation. The second loop would simulate Shuttle-only thrust levels. During abort model execution, the control variables *num* and *nnn_stp* are read in and represented the number of throttling increments for the LFBB/Shuttle and Shuttle-only segments, respectively. The larger *num* and *nnn_stp* were the smaller the throttle increments. The Fortran code was designed so that the inner throttle loop dealt with the combine thrust prior to LFBB separation, while the outer loop provided Shuttle-only thrust levels. The loops, when executed, ran through every value of LFBB/Shuttle combined thrust for each given value of Shuttle-only thrust. In this manner, every possible combination of thrust level was attained. Any successful combinations, whose end conditions met the requirements defined as a success, were recorded in the output file *ag.dat*. Data concerning g-forces was output to *Gs.dat*, and data concerning dynamic pressure was output to *adyn.dat.* Information concerning time and altitude were included in both *Gs.dat* and *adyn.dat* to aid in evaluating possible solutions for abort landing trajectories. Also included in this output, was information concerning which combination of Shuttle-only and LFBB/Shuttle thrust levels were used for the successful trajectory model.

It was mentioned previously that the reason for limiting the control value *maxT* to 1.0244 was so the value for *max Q,* 750 psf, was not violated during ascent. What was not explained was the reason for limiting the Shuttle-only equivalent *dv_max*, to 1.5. This was to avoid violating the SSME's maximum thrust capability. These requirements drove the addition of a further check being added to the code. This check would ensure that the thrust levels did not violate their respective performance envelopes. In the subroutine *Rhs.for,* when the code was calculating the values of thrust T, angle of attack α, and yaw β,, which would give specific values for the rate of change of velocity dV, flight path angle $d\gamma$, and heading angle $d\psi$; the check would verify that the values for thrust had not been violated. If a violation occurred the thrust levels would be reset to their respective maximums. This *forcing* had no ill effects on the trajectory since the code would continue its iterations, trying to achieve the desired values for dV, $d\gamma$, and $d\psi$.

4.3.2 Initial Abort Model Execution.

The test case for the abort model would be the evaluation of a simulated abort one second after clearing the tower. The model evaluated the entire RTLS window for promising throttle down times. The program *abort.for*, was initialized with the outermost loop set to evaluate the abort trajectory from L+10 seconds to L+119 seconds. Again, the model picks up the abort just one second after the Shuttle clears the tower at L+9. This is the reason for initializing the model at L+10 seconds. For this first run no steering was involved. Instead, it was desired to record what throttle down times allowed successful achievement of the TAEM end conditions for altitude and velocity. Besides these checks for velocity and altitude, additional code was included to calculate the maximum lift accelerations. The Shuttle would experience these accelerations during ascent, and the pull-up that would occur after ET separation during the modified skip reentry. This pull-up would be necessary in order for the Shuttle to acquire the correct glide slope for the abort landing approach. Equation (4-7) lists the equation added to the code to calculate the number of *g's* the Shuttle would experience. H_o is the scale height of the atmosphere, which is approximately 23,000 ft or 7010.4 m [39:252].

$$g's = \left(\frac{Vel^2}{H_0}(1 - Cos\gamma)\right) \Big/ 9.80655\, m/s^2 \tag{4-7}$$

Equation (4-8) shows the calculations involved with conducting the modified skip reentry maneuver. Equations (4-9) & (4-10) derive the components for Equation (4-7) [39:241].

$$H = H_0 \ln\left(\frac{K_L H_0}{\cos\gamma + B}\right) \tag{4-8}$$

where

$$B = K_L H_0 e^{-H_i H_0 /} - \cos\gamma_i , \tag{4-9}$$

and

$$K_L = \frac{C_L A \rho_0}{2m} . \tag{4-10}$$

K_L is the lift constant, B is the constant of integration evaluated at the atmospheric entry point, and C_L is the vehicle's *coefficient of lift* at that point in time. By ignoring the relatively small exponential term an indication that the final pullout height occurs above ground is;

$$K_L > \frac{1}{H_0}\left(1 - \cos\gamma_i\right). \tag{4-11}$$

This assumes a γ of zero degrees at the bottom of the pull-out. If this holds true, then further efforts using Equations (4-8), (4-9), and (4-10) can be accomplished to find the exact height at which this occurs. Data concerning the final pullout height was stored in the output file *Pull.dat.* Data was only sent to this file after the ET had separated. Also, the final pullout height was calculated for every second after ET separation. This would give a complete picture as to what points of the trajectory would allow for favorable end conditions to be met, thus signifying a successful abort landing.

4.4 Analysis

4.4.1 Modifications.

As mentioned, once all the code was set, an initial run was accomplished to get an idea of when in the trajectory possible solutions existed. Solutions in this case referred to candidate times when throttling down would lead to the attainment of favorable end conditions. Initially, these end conditions were limited to altitude, velocity and to a lesser extent flight path angle. No steering was involved. The amount of g's during the trajectory as well as the levels of dynamic pressure were set as the filters for this test case. Analysis would concentrate on the output data stored in *Gs.dat* and *adyn.dat.* With respect to the one-second abort case, there were two clusters of possible throttle times. Again, these clusters stemmed from the different throttle settings. Groups of 8 and 9 were located about the 43 and 50-second points, respectively. These then would be the initial points where the throttling down would occur.

Table 4-6 represents a sampling of the data that was collected from this initial run. Upon analyzing this data it was concluded, besides indicating the times to investigate throttling down, some additions to the abort model were in order. First, the model treated the Shuttle as being ballistic in nature, this would not do. More robust routines were needed so that the control variables would have a more realistic impact on the model's trajectory. This entailed including routines to handle adjusting the angle of attack α. A lower α would provide more forward velocity, while a higher α, had a braking affect. Also lacking was a method for checking the distance to a particular target, and the heading correction needed to get there. This would aid modifying the control variables for the optimum trajectory solution. To do this, spherical trigonometry was employed, as it had in Figure 3-7 of Section 3.2.5.1, for defining inertial azimuth. This would enable the calculation of the arc length that connected the Shuttle to the target location.

Table 4-6. Sample of Initial Throttle Point Data.

```
Shuttle dV after LFBB Sep =   1.600000000000000
Combined dV this run was =    1.920750000000000E-001

Distance to Runway in miles =       1083.657181839782000
Heading correction (act-this)degs =   18.180044436394500
ET Sep s>20 begin pull up
g s at bottom of pull up =             1.740252620895963E-001

r is       25115.974758880260000 meters
theta is     288.690652190518800 degrees
phi is        43.822618427369530 degrees
v is         420.744277726209900 meters/sec
gamma is     -21.185274775953910 degrees
psi is        30.244389583444540 degrees
aoa is        40.000000000000000
beta is   0.000000000000000E+000
m is      104326.245099784800000 kgs mass
Seconds into flight        1319
Dyn Press psf is             74
Throttle Down Time =         53
Attained 25 km!

****************************************************
Shuttle dV after LFBB Sep =   4.266666666666667E-001
Combined dV this run was =    3.414666666666667E-002

Distance to Runway in miles =        135.734336979271600
Heading correction (act-this)degs =   32.177274720866300

r is       25516.823516234610000 meters
theta is     280.552786054698200 degrees
phi is        30.114503977147290 degrees
v is        1516.989689924279000 meters/sec
gamma is     -56.476926541162290 degrees
psi is        24.431676131829990 degrees
aoa is        40.000000000000000
beta is   0.000000000000000E+000
m is      104326.245099784800000 kgs mass
Seconds into flight         266
Dyn Press psf is            905
Throttle Down Time =         53
Attained 25 km!
```

Distance to target was calculated in the Fortran code as,

```
dist_go = dacos(dcos((pi/2.d0)-lat_land)*dcos((pi/2.d0)-
          lat_shut)+ dsin((pi/2.d0)-lat_land)*
          dsin((pi/2.d0)-lat_shut)*dcos(long_shut -
          long_land)).
```
(4-12)

The heading correction needed to get to the target was defined as,

```
psi_cor =(-1.d0*(dasin((dsin(pi/2.d0)-lat_land)*
         (dsin(long_shut - long_land)/dsin(dist_go))))) [39].
```
(4-13)

Two other routines were added to enhance the control of the model. The first checked for when the flight path angle γ had gone to zero. Once this occurred, it would force $\dot{\gamma}$ to zero, and would provide the helicopter effect mentioned previously. This would allow the maintaining of altitude, while increasing forward velocity and downrange distance. Data concerning the γ angle was stored in the file *agam.dat*, and analysis of this data helped in the development of this routine. Second, a routine was put in place that would only execute if the LFBB had separated and the orbiter was coming *down* the far side of the *semi-ballistic* trajectory. This routine would use the control variable *Δdgam*. Runs of the trajectory model were analyzed for the points where the LFBB had already separated as well as the altitude where the trajectory had peaked. The model was coded so that once the trajectory had met these two conditions, LFBB had separated and the Shuttle was coming down from the peak, *Δdgam* would reverse the sign and change the magnitude of the flight path angle γ. In this manner *Δdgam* would expanded the *helicopter* technique. The primary purpose was to limit the magnitude of $-\gamma$ as the Shuttle came down the backside of the trajectory peak. If the angle was permitted to become too great, then the Orbiter would stand little chance of surviving the pull-up from the modified skip reentry maneuver.

4.4.2 Abort Model Execution.

Having made the enhancements to the model, each abort scenario was run. For each scenario, if any trends were noted during the initial collection of the data, minor modifications were made to the control variables involved. Effort was made to identify the primary control variable responsible for a specific trend. After the model was executed, the output data was analyzed for any constraints whose values might have been exceeded. The numbers of g's and dynamic pressure were the two primary filters used in the initial analysis. If these two could not be met, then none of the other constraints mattered. If a constraint had been violated, the values of *num* and *nnn_stp* responsible for the constraint violation would not be used in future iterations. Again, any trends were noted and the process was repeated.

During this iterative process, the control variable most responsible for the error in question was identified if possible. It was then modified to improve results and then reloaded into the model for another cycle. This process was repeated numerous times. Any function that could be absorbed by the computer was coded into the model. During the initial runs of the model, prior to any combinations of *maxT* or *dv_max* being discarded, simulations could exceed 10 hours in duration. The amount of data being evaluated was massive; this was apparent since these long duration's took place on a Pentium II, 350 MHz computer.

4.5 Results

4.5.1 One Second Abort Scenario.

The use of the control variables combined with the numerous iterations performed by the simulation eventually produced solutions. For the first abort scenario, the model simulated an abort just one second after the orbiter had cleared the launch tower. The control variables and their respective values that were responsible for this solution are depicted in Table 4-7.

Table 4-7. One Second Abort Control Variable Solutions.

CONTROL VARIABLE	EFFECTIVE SOLUTION VALUE
maxT	0.320125d0
num	5
dv_max	1.3d0
nnn_stp	13
d_gam (Initial)	0.1d0 * (-1° per Time Step)
d_psi	0.53 *(-1° per Time Step)
Δdgam (Shut&ET; Alt below 85 km)	-23d0 * (d_gam)
γdot -> 0	+0.25° (γ)
α, Angle of Attack (Initial at ET Sep)	42°
α, (Shuttle Only, No ET; Below 34 km)	33°

Table 4-8 lists the input file used for this solution. The top half contains the seven-element state vector of the vehicle. This was obtained from the output provided from the nominal model. Following these seven are the initiation time of the abort and the maximum simulation time. The maximum time was set to 600 seconds, this could be easily adjusted if necessary. The final two items in the top half of the input file are *Nstp* and *Nskp*. These terms set the range for the integration routine *Haming*. The lower half contains from left to right starting with the top row: *minT*, *maxT*, *num*, and on the bottom row; *dv_min*, *dv_max*, and *nnn_stp*. The number 4 that is shown, is the output flag for maximum data output. The *Time-step* selected for this model was two seconds. This would limit the amount of data output without sacrificing the resolution of the trajectory.

Table 4-8. One Second Abort Input File.

```
1.000027603394376        4.876375247312332    4.993081693153024E-001
4.521228749522317E-003   1.566322763194498    8.325507229168753E-001
3.413484164565641E-019
1.240429996817460E-003   7.43667785562000E-001
299.000000              50.000000

      0.d0      1.0244d0      16
      0.1d0     1.5d0         16
      4
c Above is abort 1 SSME failure at L+10 seconds
```

Table 4-10 and Table 4-13 lists output showing the conditions which existed when the LFBB and ET separated from the Shuttle during the L+10 and L+119 second abort scenarios, respectively. Each separation was comprised of three parts. With respect to the LFBB separation portion of Table 4-10, the first section gives the particular values for weight flow rate *wdotshut,* and *Combined dV* the Shuttle was experiencing at the moment of LFBB separation. The term *Shuttle dV*, is the value for the rate of change in velocity the Shuttle would experience after LFBB separation when it would be thrusting with just its remaining SSMEs. *Thrust* is the specific value of thrust at the time of separation and was used as a check to ensure the model was performing within the Shuttle's performance envelope. The second part deals with the orbiter's relative position to the targeted landing site. Lastly, the third part deals with the specific state conditions of the orbiter during separation. The state conditions included the Shuttle's altitude r, longitude theta, latitude phi, velocity v, angle of attack aoa, yaw angle beta, mass, time into flight and the dynamic pressure experienced.

Various data was used for the analysis of this research work. Data collected included information concerning the Shuttle's altitude and velocity, the dynamic pressures and g-forces experienced, as well as the times when these events took place. Table 4-9 shows the various output files and the related data that was stored in each.

Table 4-9. Abort Model Data Files & Information Stored.

DATA FILE	DATA STORED
Agam.dat	Time, Altitude, γ (Flight Path Angle)
Adyn.dat	Time, Altitude, s (Dynamic Pressure)
Ags.dat	Time, Altitude, g's Experienced
Apull.dat	Altitude, Height @ Bottom of Pull-up, Kl, B, Cl, aoa, Gamma, v, θ, ϕ, Shuttle dV, Combined dV, wdot, Time of Event
Ag.dat	Shuttle dV, Combined, Thrust, Distance to Runway in miles, Heading correction, g's bottom of pull-up r (Altitude), θ, ϕ, v, γ, ψ, aoa, beta, m, Seconds into flight, Dynamic Pressure, Throttle Down Time

The first chart, shown in Figure 4-1, depicts the Shuttle's altitude for a given time. As shown by this chart, the modified skip reentry takes place at approximately L+475 seconds with the vehicle leveling off at around 25 km. This trajectory shows that a part of the required end conditions, altitude at TAEM interface, has been met.

Further conditions are met in Figure 4-2. This chart shows the level of g-forces experienced during the abort-landing trajectory. It should be noted that the 5-g constraint was not violated. The first peak in the amount of *g's* experienced occurred during the initial ascent. Here, the acceleration force reached 3.22 g's. The second peak occurred during the modified skip reentry where the maximum value reached 4.54 g's. The control variables primarily responsible for the level of g's experienced were *maxT*, *dv_max*, and *Δdgam*. *maxT* and *dv_max* were responsible for the level of g's experienced during the initial moments of liftoff. *Δdgam* was the primary control variable that affected the amount of g's experienced during reentry. Not only could the value for *Δdgam* be varied, but also the altitude at which it took place.

Table 4-10. 1 Second Abort Scenario: LFBB & ET Separation Conditions.

```
*****************************************************
 LFBB SEPARATION FOR 1 SECOND ABORT
 wdotshut    1.201271170693062E-019     Throttle down time = 43 secs
 Shuttle dV for this run  =             1.300000000000000
 Combined dV this run was =             3.201250000000000E-001
 Thrust =                               1.807410999005179E-019
 Distance to Runway in miles =        240.318751289409100
 Heading correction (act-this)degs =    4.847315180001611
 r, Alt is  49860.152156924120000 meters
 v is         637.128106740850500 meters/sec
 theta is      80.532265769428800 Degrees West Longitude
 phi is        28.690945415062010 degrees Latitude
 gamma is      62.752268843636980 degrees
 psi is        36.887205156411550 degrees
 aoa is         0.000000000000000E+000
 beta is        0.000000000000000E+000
 m is      898178.793598172100000 kgs mass
 Seconds into flight          127
 Dynamic Pressure, psf is      4
****************F********************************
 ET SEPARATION FOR 1 SECOND ABORT
 wdotshut    1.201271170693062E-019     Throttle down time = 43 secs
 Shuttle dV for this run  =             1.300000000000000
 Combined dV this run was =             3.201250000000000E-001
 Thrust =                               0.000000000000000E+000
 Distance to Runway in miles =        129.529804364116100
 Heading correction (act-this)degs =  12.355226183837430
 r, Alt is  68766.470192886850000 meters
 theta is      80.288266238911530 Degrees West Longitude
 phi is        30.414406850330130 degrees Latitude
 v is        1528.069987343930000 meters/sec
 gamma is     -18.911079026122010 degrees
 psi is       -13.861049474617740 degrees
 aoa is        42.000000000011450
 beta is        0.000000000000000E+000
 m is      104326.245099784800000 kgs mass
 Seconds into flight          391
 Dynamic Pressure, psf is      2
```

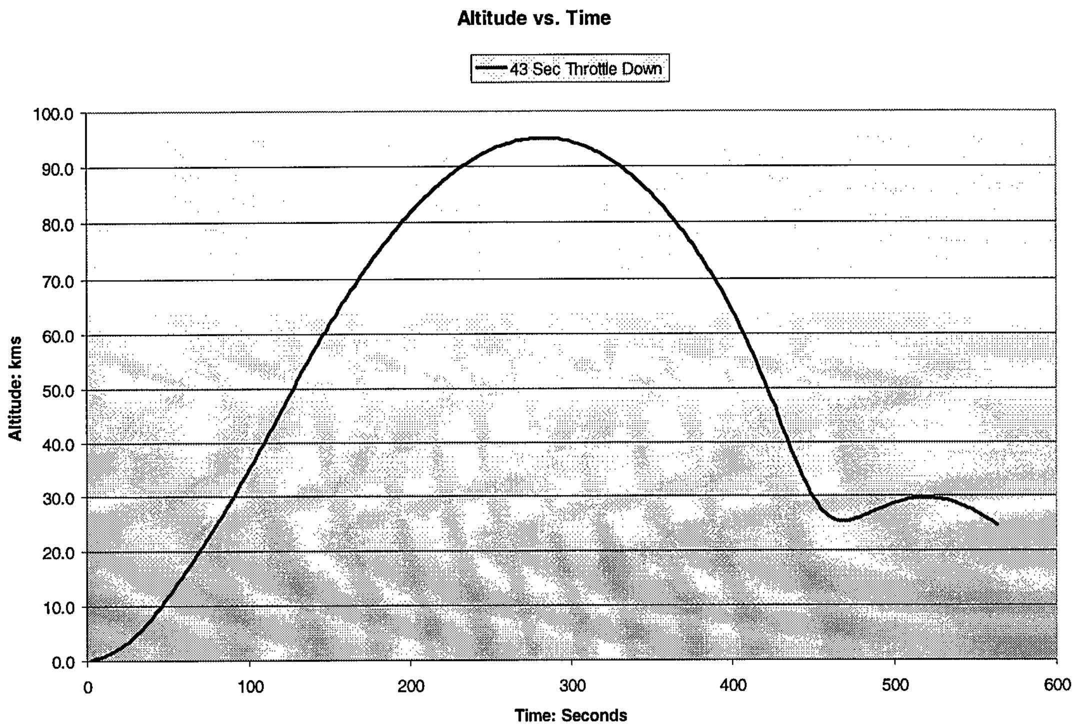

Figure 4-1. One Second Abort: 43 Second Throttle-Down Altitude vs. Time.

Analyzing data related to the value set for *Δdgam*, and the location where the control variable was initiated, showed that the original helicopter approach had to be modified. The model, once it detected a flight path angle γ equal to 0°, would try to maintain this 0° by forcing $\dot{\gamma}$ to 0. This was accomplished by adjusting thrust, α, and β. In doing this, the model would over compensate by vectoring all the thrust into the vertical direction. This would cause the trajectory to deteriorate too quickly. Once the LFBB's propellant was spent, the vehicle would reenter the atmosphere with too great of a flight path angle γ. A solution was derived to modify this helicopter approach by allowing the orbiter to crest the trajectory's peak, build up horizontal velocity, and then modify $\dot{\gamma}$ prior to ET separation. The solution was based on a smaller rate of change to $\dot{\gamma}$. Table 4-7 lists the value of $\Delta dgam$ for the one second abort as (-23d0*(d_gam)). Here d_gam was initially set to -0.1 degrees per two seconds of flight.

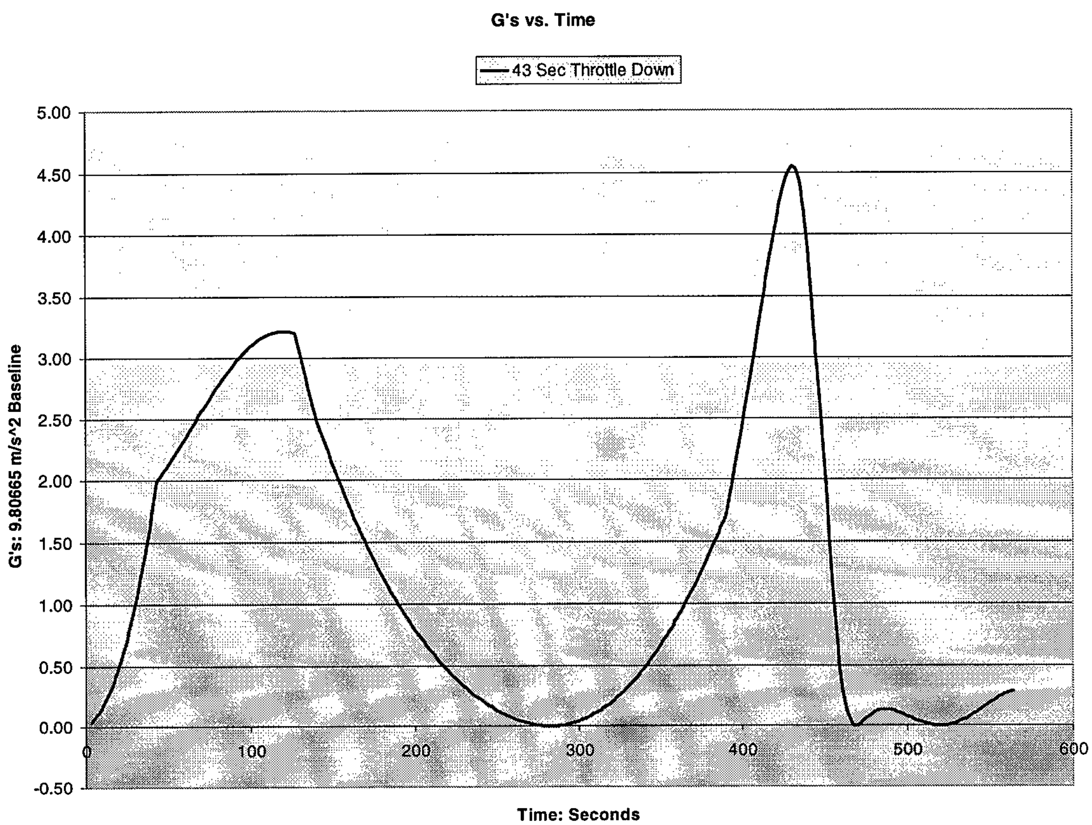

Figure 4-2. One Second Abort: G-Force vs. Time.

The final chart for the one-second abort, shown in Figure 4-3, relates how dynamic pressure varies with time during the different phases of the abort-landing trajectory. As predicted by this researcher's calculations, the largest value for dynamic pressure occurs during launch and was 748.8 psf. A lesser spike during reentry of 501.3 psf followed this.

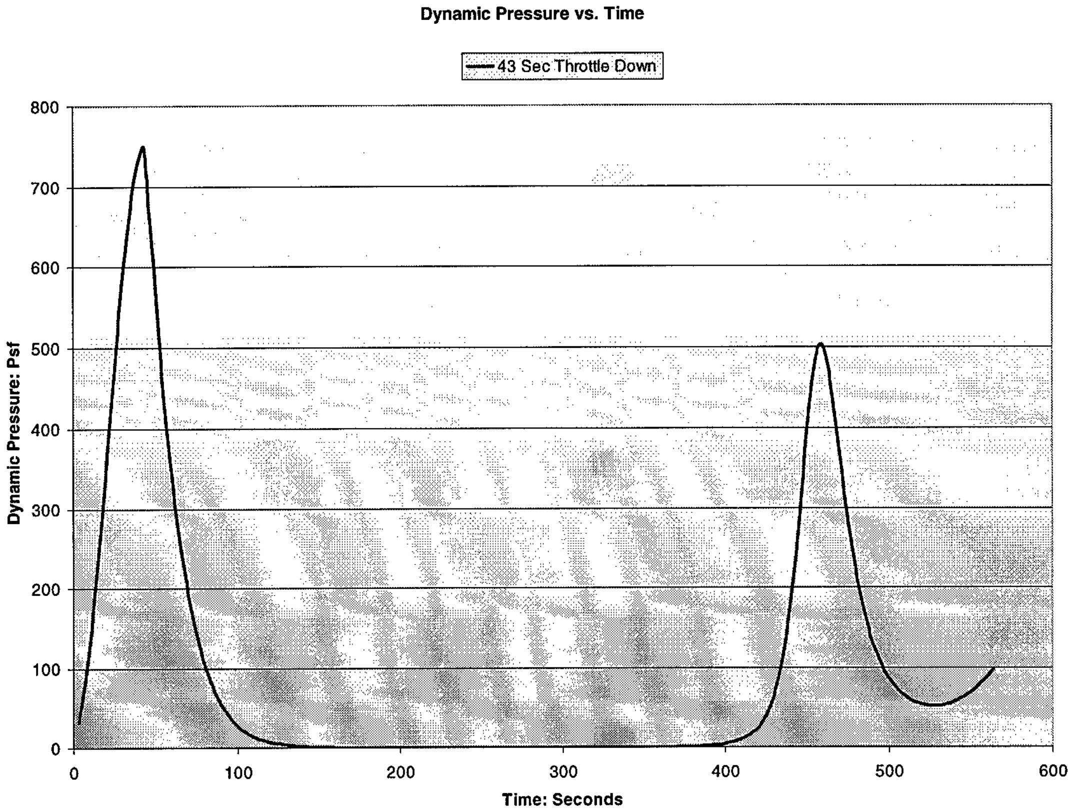

Figure 4-3. One Second Abort: Dynamic Pressure (psf) vs. Time.

The control variables primarily responsible for affecting the levels of dynamic pressure were *maxT* for the ascent, where thrust levels directly affect the vehicles velocity and hence the value of *max Q*. The altitude and value of *Δdgam*, which was used to modify the flight path angle γ, also affected dynamic pressure.

4.5.2 119 Second Abort Scenario.

The major difference for this abort scenario, as compared to the one-second abort, was that the Shuttle would not be as concerned with the amount of g's and dynamic pressure related to the ascent portion of the trajectory. In this scenario, the LFBB was just 16 seconds from separation. As mentioned previously, this was just prior to where the TAL abort mode could begin initiation. Table 4-11 lists the control variables and their corresponding values, which led to the second successful abort-landing trajectory. As noted in Table 4-12, the large initial value for d_psi had little affect on the trajectory, this was attributed to the high velocities the orbiter was experiencing at the time of this abort. By the time the LFBB could influence the trajectory laterally it had already separated.

Table 4-11. 119 Second Abort Scenario Control Variable Solutions.

CONTROL VARIABLE	EFFECTIVE SOLUTION VALUE
maxT	1.0244d0
num	16
dv_max	1.5d0
nnn_stp	15
d_gam (Initial)	0.1d0 * (-1° per Time Step)
d_psi (Initial) (No Benefit Even w/this Large of a Value)	10.9 *(-1° per Time Step)
Δdgam (Shut&ET; Alt below 74 km)	-6.9d0 * (d_gam)
d_psi (During Δdgam Mods at 74 km)	8.43d0*(-14.0815236524d0)
γdot -> 0 (Solution Did Not Use This CV)	+0.25° (γ)
α, Angle of Attack (Initial at ET Sep)	38°
α, (Shuttle Only, No ET; Below 34 km)	32°

Key to attaining the necessary constraints for this successful scenario were the control variables related to thrust and the change of the heading angle. After LFBB separation had occurred the greatest influence of d_psi was experienced by the trajectory. Due to the location of the orbiter during abort initialization, maximum values for *maxT* and *dv_max* were necessary to gain enough height so as to allow successful completion of the modified skip reentry. Table 4-12 list the input file used for the 119-second abort scenario. Of interest is the modification to *Nstp*. It was changed to 245.0 to keep the *time step* set to two-second intervals. Table 4-13 list the 119-second conditions for the LFBB and ET separation points.

Table 4-12. 119 Second Abort Input File

```
1.005668904744045        4.879500322815867         5.024450999202795E-001
1.501080746434611E-001   6.442927216051213E-001    7.180002122034157E-001
1.683184621520560E-019
1.364472996499195E-001    7.43667785562000E-001
245.000000        50.000000

      0.d0      1.0244d0      16
      0.1d0     1.5d0         16
      4
c Above is abort 1 SSME failure at L+119 seconds
```

Analysis for the L+119 second abort was carried out in much the same manner as was done for the one-second-abort scenario. Of interest in Figure 4-4, is that the modified skip reentry concludes its pull-up maneuver at approximately 25 km. Also, as can be seen in the figure, the Shuttle maintains approximately 25 km for over 100 seconds. This is attributed to the control variable α. Once the ET is separated, α is initially set to 38°. This ramps down to 32° by the time the orbiter is at the TAEM interface. Normally, this angle is closer to 14° at this point. But, due to the steepness of the reentry flight path angle γ, a higher than normal α allows excess energy to be bled off and thus allows for the most optimistic TAEM interface conditions. While the Shuttle is oscillating about the 25-km altitude range it is loosing energy. If banking is necessary to better align the Shuttle with the runway's heading alignment circles, the excess α angle could be decreased to make up for the loss in energy. This decrease in α would ensure that the Shuttle's flight path would not end up short of the runway.

Table 4-13. 119 Second Abort Scenario: LFBB & ET Separation Conditions.

```
**************************************************
 LFBB SEPARATION FOR 119 Second Abort
 wdotshut    1.331471924030000E-019      Throttle down time = 111 secs
 Shuttle dV for this run  =              1.600000000000000
 Combined dV this run was =              1.024400000000000
 In 100, prior to LFBB Sep
 Thrust =                                2.333798385024705E-019

 Distance to Runway in miles =         279.811546626525400
 Heading correction degrees  =          -1.964621508184954

 r, Alt is   43257.536257864410000 meters
 theta is       80.357355612175740 Degrees West Longitude
 phi is         28.856014282922050 degrees Latitude
 v is         1286.870360158173000 meters/sec
 gamma is       36.172806068010490 degrees
 psi is         41.144466151708220 degrees
 aoa is          0.000000000000000E+000
 beta is         0.000000000000000E+000
 m is       889367.833829346200000 kgs mass
 Seconds into flight          120
 Dynamic Pressure psf is       44

*****************F*******************************
 ET SEPARATION FOR 119 Second Abort
 wdotshut    1.331471924030000E-019     Throttle down time =  111 secs
 Shuttle dV for this run  =             1.600000000000000
 Combined dV this run was =             1.024400000000000
 Thrust =                               0.000000000000000E+000

 Distance to Runway in miles =        131.262041450512900
 Heading correction degrees  =         11.719085978444140

 r, Alt is   68700.019393986430000 meters
 theta is       79.127234374809750 Degrees West Longitude
 phi is         31.155882497868720 degrees Latitude
 v is         1496.380812467676000 meters/sec
 gamma is      -13.667163290946160 degrees
 psi is         -5.584126877298266E-001 degrees
 aoa is         37.999999999988540
 beta is         0.000000000000000E+000
 m is       104326.245099784800000 kgs mass
 Seconds into flight          340
 Dynamic Pressure psf is        2
```

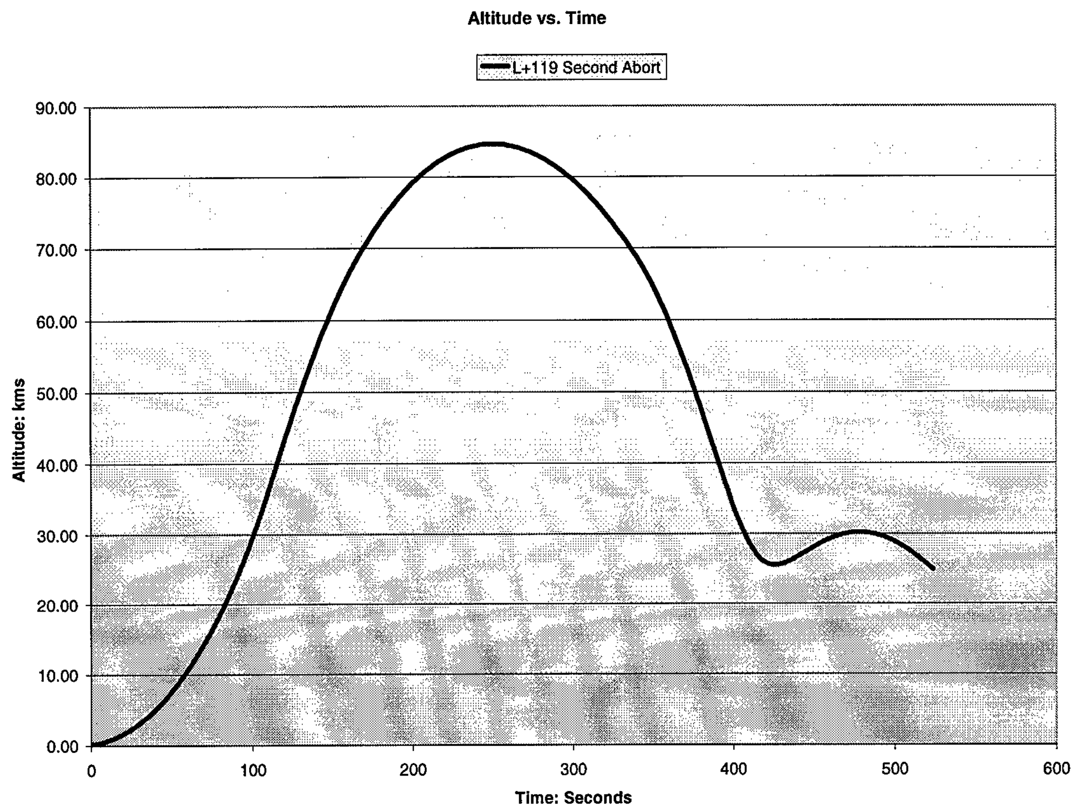

Figure 4-4. 119 Second Abort with Full Throttles Altitude vs. Time.

Figure 4-5 shows that the g's experienced were less than those for the one-second abort. This correlates well with the calculations of the atmospheric model, since the Shuttle is above a good portion of the atmosphere at abort initiation. Again, both acceleration force spikes are within the maximum set by NASA [37]. The first spike occurs during the Shuttle's climb for altitude and reaches 4.08 g's. The second spike occurs during the Shuttle's reentry into the atmosphere and completion of the modified skip reentry maneuver. Here, the Shuttled experienced a maximum of 3.79 g's. During this abort scenario, the control variable *Δdgam* was most responsible for influencing the amount of g's experienced by the Shuttle. If the flight path angle was allowed to become too large, then the acceleration forces would approach and possibly exceed the constraints for TAEM interface and a successful abort landing.

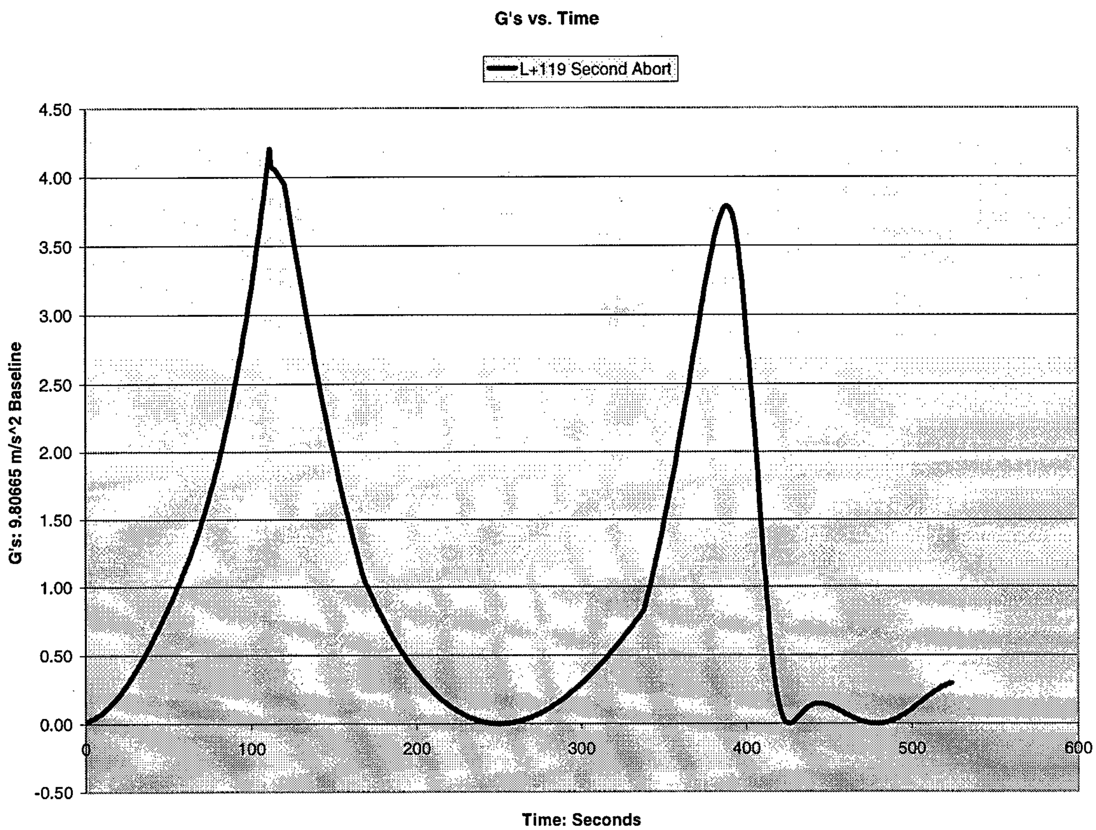

Figure 4-5. 119 Second Abort with Full Throttles G-Force vs. Time.

The values for dynamic pressure depicted in Figure 4-6 were also within the performance envelope of the Shuttle. As expected, *maxQ* during ascent was less than the one-second abort, since the Shuttle was following a nominal throttle profile as it climbed through the atmosphere. The value for dynamic pressure during ascent reached 557-psf, the value was taken from the data file *Nom_dyn.dat*. During reentry the high values for dynamic pressure were related to the higher than normal approach velocity. The peak value of 505.4 psf still falls within the tolerances given by NASA [37]. The control variable most affecting the level of dynamic pressure experienced during reentry was again α. The trade-off was, any lowering of α meant an increase in velocity, which was directly related to an increase in dynamic pressure.

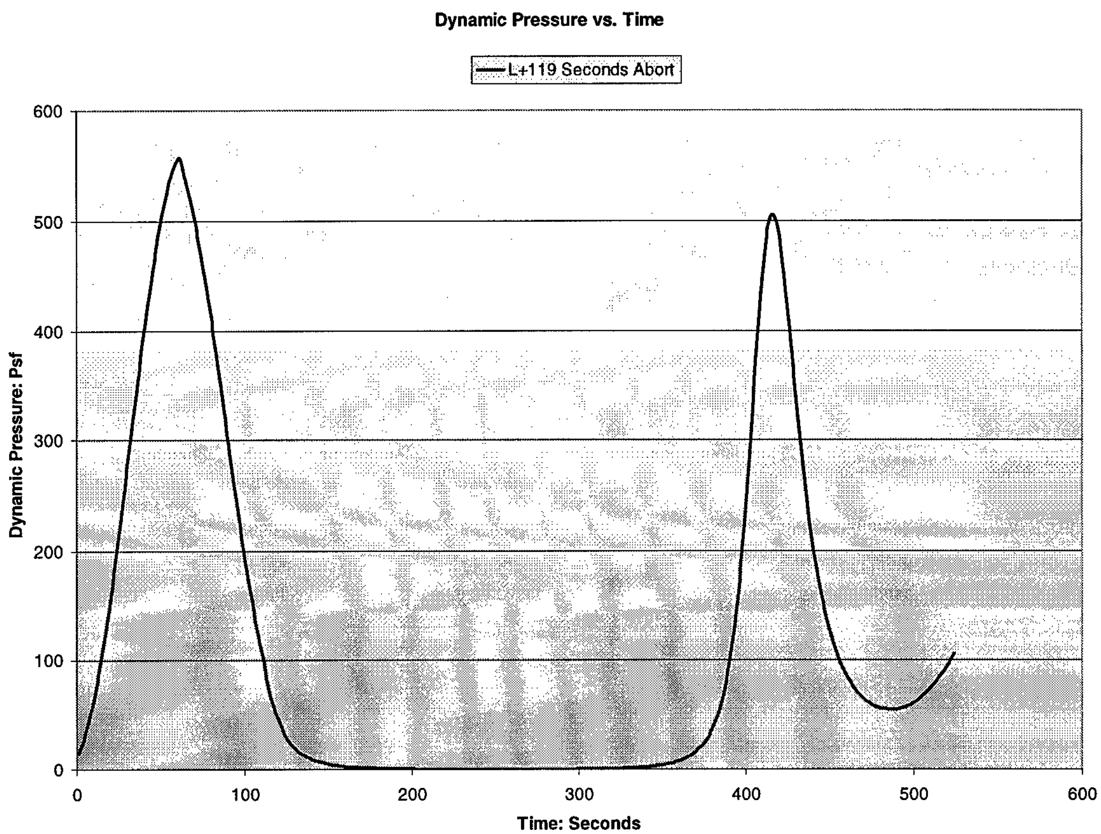

Figure 4-6. 119 Second Abort with Full Throttles Dynamic Pressure (psf) vs. Time.

4.5.3 Abort Trajectory Plots.

Once a successful abort run with its corresponding data had been collected, the latitude and longitude data was extracted from *ag.dat* for plotting. The latitude and longitude points for the successful trajectory were then input into a *Garmin III GPS Receiver.* This receiver was capable of displaying a topographic map and allowed the input of waypoints, which represented user-defined points along the trajectory. These waypoints were then uploaded to *Street Atlas USA 6.0* so that the trajectories could be displayed and printed. With this level of graphic detail the 45th Space Wing's Range Safety Office could easy note if any destruct line boundaries had been violated with a particular abort-landing trajectory. This could aid in understanding of which, if any, Launch Constraint Criteria (LCC) may need to be waived so as to save a Shuttle and its crew. Figure 4-7 and Figure 4-8 show the two abort trajectories along with the separation points depicting where the ET and LFBB were dropped from the Shuttle. Figure 4-7 shows the one-second-abort trajectory from initiation until TAEM interface for the Savannah International Airport. Output from this abort scenario, shown in Table 4-10, indicates that the LFBB separated at an altitude of 49 km with a velocity of Mach 2, and at 2 psf dynamic pressure.

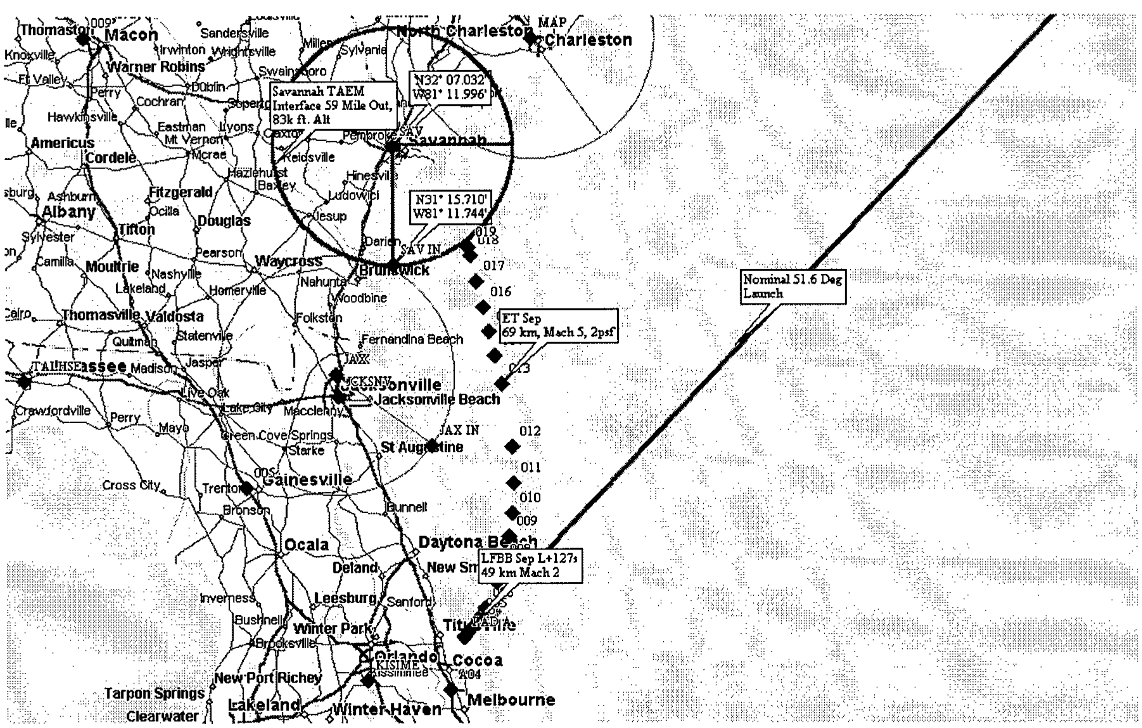

Figure 4-7. 1 Second Abort Landing Trajectory, Target: SAV TAEM.

Figure 4-8 shows the abort trajectory for the 119-second abort scenario. The orbiter follows the nominal trajectory at first. After abort initiation, the orbiter deviates from the nominal trajectory. As shown in the figure, the model modifies the Shuttle's trajectory so that it intersects the TAEM interface point for the Charleston International Airport.

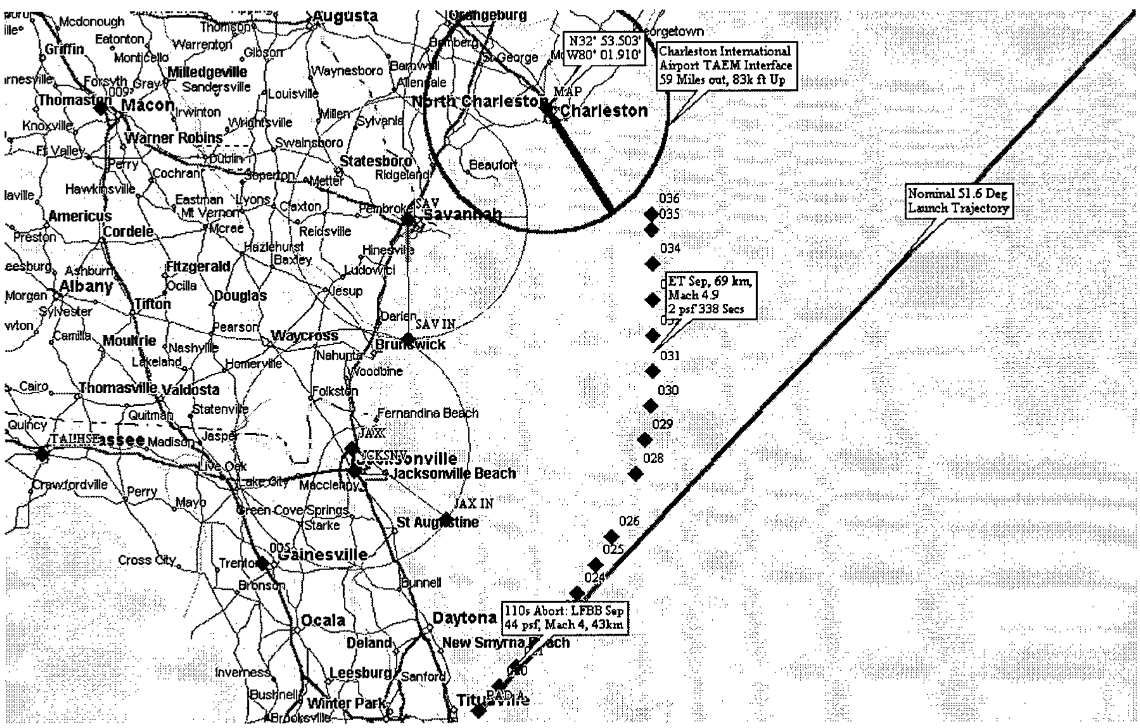

Figure 4-8. 119 Second Abort Landing Trajectory, Charleston TAEM

Figure 4-9 is a close-up of the point where the LFBB separated during the 119-second abort scenario. As shown, the LFBB separated 120 seconds after launch at an altitude of 43 km at Mach 4.2, and at a dynamic pressure of 44 psf.

With the information provided by Boeing, both of these abort scenarios show that it may be possible to not only recover the Shuttle and its crew, but also the LFBBs. At the locations given for LFBB separation, the LFBBs should have enough performance capability to allow for successful return to the Cape Canaveral's *Skid Strip*, or Kennedy Space Center's *Shuttle Landing Facility (SLF)*.

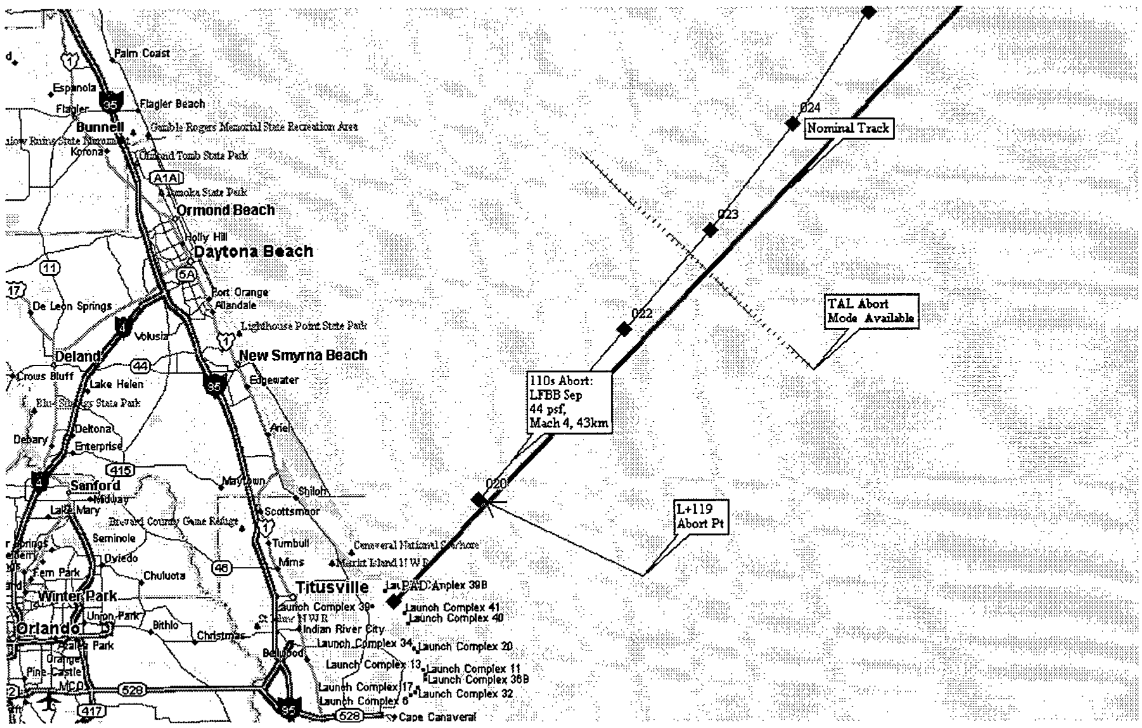

Figure 4-9. 119 Second Abort: LFBB Separation Conditions & TAL Availability Line.

4.6 Summary

In summary, this chapter has presented two abort scenarios, which occur at either end of the RTLS abort window. In both cases solutions have been found that allow for the successful completion of an abort landing to a southern East Coast airport. For each success the control variables responsible for attainment of the end conditions were specified. Also specified was which element of the trajectory was most affected by a particular control variable.

This chapter also provided the input values necessary to replicate the results, along with explanations of what each input value represented. These input values, when combined with the abort model Fortran code listed in Appendix 2, and information concerning when the abort scenario would take place, and specific throttle down times, will allow replication of the successful abort landing trajectories presented in this thesis.

5. Conclusions and Recommendations

5.1 Restatement of Research Goal

The LFBB is an intended upgrade for the Space Transportation System. The LFBB design will replace the solid rocket motors the Space Shuttle now uses with a liquid engine booster that is capable of being throttled. The design will also make use of conventional jet engines to facilitate powered recovery of the LFBB back to the Kennedy Space Center for reuse.

A key benefit of the LFBB is that it may eliminate the Return to Launch Site (RTLS) abort mode for Shuttle emergencies. Currently, if an abort occurs within the first 150 seconds of launch the only intact abort alternative is the risky RTLS maneuver. Astronauts are quick to point out that this is the least favored abort option. With the enhanced performance characteristics of the LFBB, East Coast aborts to commercial or military airfields may be possible.

With NASA placing requirements upon Boeing to investigate the *TAL-from-the-pad* capability in its design of the LFBB, situations may still arise when the Shuttle and its crew need to get back on the ground quicker than the 25-30 minutes that TAL offers.

The goal of this research was to show that it might be possible to eliminate the RTLS abort procedure, and attempt landings along the East Coast of the United States. With the enhanced performance capabilities of the LFBB, coupled with those of the Space Shuttle, it should be possible to cover the entire RTLS window, which extends from launch until L+150 seconds where the TAL abort mode becomes available. Also, if

TAL-from-the-pad becomes a specific NASA requirement, then work from this research would provide an alternative method for getting the Shuttle down quickly.

5.2 Conclusions

This abort trajectory model does indeed show that the LFBB could be the key to the future elimination of the RTLS abort mode. The researcher believes that this thesis succeeded due to the enhanced capabilities liquid booster engines offer over conventional solid rocket boosters. Supporting this was the fact that the liquid boosters, modeled by this thesis, allowed for modifications to be made to the Shuttle's trajectory. These modifications thus made successful abort landings possible. The current SRBs used by the Shuttle would not allow this type of early abort landing to occur. Current Shuttle crews have their hands tied until SRB separation. By the time this occurs at L+120 seconds, many potential early abort sites have been passed.

In conclusion, with the increased number of flights planned for building, and then servicing the new International Space Station, a safe and cost effective upgrade is needed for the aging Shuttle fleet. With the results of the abort trajectory model supporting the capabilities of the LFBB, the LFBB proves to be a viable next step in the evolution of the Shuttle Program.

5.3 Significant Results of Research

Significant results of the research include proving the feasibility of East Coast landings to southern U.S. landing facilities. The research results also show that the abort landing trajectories generated by this model provide a quicker means of getting the Shuttle back down on the ground than what the TAL abort mode can offer. This last point will become increasingly important if NASA adopts the *TAL-off-the-pad* requirement for the Shuttle-LFBB launch system. Also, the research clearly points out that elimination of the RTLS abort mode should be considered. With the use of liquid booster engines the potential loss of another Shuttle could be avoided. If no other point is made, this thesis clearly shows that another method, other than RTLS, is available for the successful recovery of the Shuttle and its crew.

5.4 Recommendations for Future Research

Due to the unlimited possibilities associated with a research topic of this nature, it was necessary to create some sort of ranking as to the importance of work that should follow this research. Logically, further work should be done with this model to simulate worse scenarios than those already presented. An important area that did not get enough attention during the completion of this research was the 2 SSME out abort scenario. The probabilistic model presented by Hage [11] showed a very small percentage of occurrence for this failure mode. But, it is a potential danger. Further study in this area could possibly seal the fate of the RTLS abort mode. With the increased capabilities of the LFBB, successful recovery from a 2 SSME out abort scenario should be possible.

Also, as discussed in the introduction, since the scope of this research was limited to looking at the extremes of the RTLS window, efforts should be made to evaluate the time between tower clear and TAL availability. This would ensure a thorough understanding of what the LFBB is and is not capable of. It would also point out any gaps in abort coverage that may be a potential concern.

Appendix 1. Launch Model Fortran Code

This Appendix contains the Fortran code for the Nominal Launch Trajectory Model. Files for this model include: Nominal.for, Nom_rhs.for, Nom.in, Haming.for, and Aero.for. See Appendix 2 for the code for Atm.for, which is also part of the Nominal Launch Model.

NOMINAL.FOR

```
c     16 Oct mods T Miller
c
      program nominal
c
c     input file for this program is:
c       a 7 component state vector,
c       initial, and final times (t0,tf)
c       integration steps (nstp,nskp)
c
c
      implicit double precision (a - h)
      implicit double precision (o - z)
c
      common /ham/ t,x(7,4),f(7,4),err(7),n,h,mode
      double precision t,x,f,err,hh,h
c
      common /amat/ a(15,15),hamil,ithrot,igt,omega
      double precision a,hamil,omega
c
      common /ctrl/ aoa,beta,mdot
      double precision aoa,beta,mdot
c
      common /maxq/s

      double precision t0,tf
      double precision xic(7)

      common /debug/ idebug,ig
      common /Thrust/ Thrust
c
      real nstp,nskp
c

c     output file
c
      open(1,FILE='nom_dyn.dat',STATUS='UNKNOWN')
      open(2,FILE='nom_St_v.dat',STATUS='UNKNOWN')
      open(3,FILE='nom_H_X.dat',STATUS='UNKNOWN')

c     read in initial state vector
c
```

```
      read (*,*) (xic(ii),ii=1,7)
      write (*,*) 'Initial State Vector Shuttle/LFBB'
      write (*,*) 'r is     ',((xic(01)*6378145)-6378145), ' meters'
      write (*,*) 'theta is',(xic(02)/.0174532925199), ' degrees'
      write (*,*) 'phi is   ',(xic(03)/.0174532925199), ' degrees'
      write (*,*) 'v is     ',(xic(04)*7905.36828), ' meters/sec'
      write (*,*) 'gamma is',(xic(05)/.0174532925199), ' degrees'
      write (*,*) 'psi is   ',(xic(06)/.0174532925199), ' degrees'
      write (*,*) 'm is     ',(xic(07)*5.976D+24), ' kgs mass'
      write (*,*) ' '

      write (*,*) 'Dimensionless State Vector Shuttle/LFBB'
      write (*,*) 'r is     ',xic(01)
      write (*,*) 'theta is',xic(02)
      write (*,*) 'phi is         ',xic(03)
      write (*,*) 'v is           ',xic(04)
      write (*,*) 'gamma is',xic(05)
      write (*,*) 'psi is   ',xic(06)
      write (*,*) 'm is           ',xic(07)
      write (*,*) ' '
      write (*,*) 'Initial Vac Total Thrust = 5.67994603143d-19'
      write (*,*) ' '
c
c     read in times and steps, tf 135.1 sec LFBB sep
c
      read (*,*) t0,tf
      read (*,*) nstp,nskp
c
c     number of ode's, initial angle, step
      n = 7
      do 10 i = 1,7
          x(i,1) = xic(i)
   10 continue

      t = t0

c     time incr 135.1/(50*50)= .33775 secs print 2.702 secs

      h = (tf-t0)/dble(nstp*nskp)

c     nxt = 0 is hamings initialization flag...
      nxt = 0

c     initalize haming (we hope)
      call haming(nxt)

c     check!
      if(nxt .eq. 0) stop 909
c     if we are still alive, then...

c     Do the integration!
      do 20 i = 1,nstp
              do 19 j = 1,nskp
                      call haming(nxt)
   19 continue
c     double loop structure keeps haming from burying
```

```
c     us in output

c     Time, alt, Pres N/m^2(s*MU/TU^2*DU), Pres lbf/ft^2(N to lbf
c      then m^2 to ft^2); 1 atm = 2116.2166 lb/ft^2 = 101325 N/m^2

      write (1,*) (t*806.8118744d0),((x(1,nxt)*6378145d0)-6378145d0),
     + (s*30.0618114811D+9)

c     Write out current time's state vector

      write (2,*) x(1,nxt),x(2,nxt),x(3,nxt),x(4,nxt),x(5,nxt),
     + x(6,nxt),x(7,nxt)
      write (2,*) t,tf
      write (2,*) nstp,nskp
      write (2,*) ' '
      write (2,*) t*806.8118744d0, ' seconds'

c   This outputs to a file alt_range.dat secs,alt m, down range m

      write (3,*) t*806.8118744d0, ((x(1,nxt)*6378145d0)-
     - 6378145d0), ((x(4,nxt)*dcos(x(5,nxt))*
     - dsin(x(6,nxt)))*6378145d0*t)
      write (*,*) ' '
      write (*,*) 'Thrust = ',Thrust
      write (*,*) 'State Vector = '
      write (*,*) (x(k,nxt),k=1,7)
      write (*,*) ' '
      write (*,*) 'r is',((x(1,nxt)*6378145d0)-6378145d0), ' meters'
      write (*,*) 'theta is',(x(2,nxt)/.0174532925199d0), ' degrees'
      write (*,*) 'phi is  ',(x(3,nxt)/.0174532925199d0), ' degrees'
      write (*,*) 'v is    ',(x(4,nxt)*7905.36828d0), ' meters/sec'
      write (*,*) 'gamma is',(x(5,nxt)/.0174532925199d0), ' degrees'
      write (*,*) 'psi is  ',(x(6,nxt)/.0174532925199d0), ' degrees'
      write (*,*) 'm is    ',(x(7,nxt)*5.976D+24), ' kgs mass'

c    Inclination based on inertial psi, takes rotation of earth into
account
      write (*,*) 'Inclination is ',datan(((7.292116d-05*806.8118744d0)*
     + x(1,nxt)*dcos(x(3,nxt))+x(4,nxt)*dcos(x(5,nxt))*dsin(x(6,nxt)))/
     + (x(4,nxt)*dcos(x(5,nxt))*dcos(x(6,nxt))))/.0174532925199d0,
     + ' degrees'
      write (*,*) 'Current time= ', t*806.8118744d0
      write (*,*) 'Current Dyn Press in psf= ', (s*30.0618114811D+9)

   20 continue
      stop
      end
c
$INCLUDE: 'haming.for'
$INCLUDE: 'nom_rhs.for'
$INCLUDE: 'atm.for'
$INCLUDE: 'aero.for'
```

NOM_RHS.FOR

```
      subroutine rhs(nxt)
```

```
c
c     equations of motion and variation for launch problem
c
c     AERODYNAMICS OFF!!!!
c
      implicit double precision (a - h)
      implicit double precision (o - z)
c
      common /debug/ idebug,ig
c
      common /Thrust/ Thrust
      double precision Thrust,Isp,m,msq,mcu,wdot
c
      common /ham/ t,x(7,4),f(7,4),err(7),n,h,mode
      double precision t,x,f,err,hh,h
c
      common /ctrl/ aoa,beta,mdot
      double precision aoa,beta,mdot
c
      common /amat/ a(15,15),hamil,ithrot,igt,omega
      double precision a,hamil,omega
c
      common /maxq/s
      double precision s
c
      double precision P0,ALT,DALT,TALT,dDdr,d2Ddr,sonic,dmfpdr
      double precision mfp,PALT
c
      double precision Kn, Cd, Cl, dcdda, dclda
      double precision aoap,faoa,faoap,dfda,delal
      double precision aoalo,aoahi,faoalo,faoahi
c
      data istart /0/
c
c
c     extract state vector
c
C DEBUG
      if(idebug .ne. 0) then
         write (*,*) 'enter rhs, nxt ',nxt
      endif
C END
      r = x(1,nxt)
      theta = x(2,nxt)
      phi = x(3,nxt)
      V = x(4,nxt)
      gamma = x(5,nxt)
      psi = x(6,nxt)
      m = x(7,nxt)

      aoa = 0.d0
      beta = 0.d0

c     write (*,*) 'mass is ',m

c     calculate common auxillary quantities
```

```
c
      sinbeta = dsin(beta)
      cosbeta = dcos(beta)
      sinaoa = dsin(aoa)
      cosaoa = dcos(aoa)
      cosgam = dcos(gamma)
      singam = dsin(gamma)
      secgam = 1.d0/ dcos(gamma)
      tangam = dtan(gamma)
      cosphi = dcos(phi)
      sinphi = dsin(phi)
      secphi = 1.d0/ dcos(phi)
      tanphi = dtan(phi)
      cospsi = dcos(psi)
      sinpsi = dsin(psi)
      Vsq = V*V
      msq = m*m
      rsq = r*r
      sphisq = secphi*secphi
      tphisq = tanphi*tanphi
      sgamsq = secgam*secgam
      tgamsq = tangam*tangam
      sphicu = secphi*secphi*secphi
      sgamcu = secgam*secgam*secgam
      Vcu = V*V*V
      rcu = r*r*r
      mcu = m*m*m
c
c     calculate aerodynamic garbage P0 in N/m^2
c
c      P0 = 101325.d0
      P0 = 99621.5573252d0
      ALT = (r - 1.d0)* 6378145d0
c DEBUG
      if(idebug .ne. 0) then
         write (*,*) 'alt, meters',ALT
      endif
C END
      call ATM(ALT,P0,PALT,TALT,DALT,dDdr,d2Ddr,sonic,mfp,dmfpdr)
c
c     convert units on rho etc
c
      rho = DALT*((6378145d0**3.d0)/5.976d24)
      drhodr = dDdr*((6378145d0**4.d0)/5.976d24)
      d2rhodr2 = d2Ddr*((6378145d0**5.d0)/5.976d24)
      Kn = mfp/(21.02d0/3.048d0)

      PALT = PALT*(6378145d0*806.8118744d0**2.d0)/(5.976d24)
c
      g = (9.80665d0*806.8118744d0**2.d0)/(r*6378145d0)
c
c     Define some constant terms
c
c     Thrust, 8 LFBB's @ 75% + 3 SSME's @ 104.5% Vac Thrust
c     Thrust at alt = Thrust Vac tot - (Press @alt * nozzel area)
```

```
      Thrust = (5.67994603143d-19)-
     - (PALT*(47.58635644d0/(6378145d0**2.d0)))

      omega = 7.292116d-05*806.8118744d0

c     wdot is the weight flow rate, Thrust over Isp proportional
c      term same for vac or sl here it is 92.877532297d+3 N/sec
c      wdot is mdot*g

      wdot = 1.27974232928d-18

      Isp = (Thrust/wdot)*806.8118744d0
      Isp = Isp/806.8118744d0

      gsea = 1.d0
      mdot = -(Thrust/(gsea*Isp))
c
c     check for vanishing vehicle
c
c      if(m+mdot*hh .lt. 0.d0) then
c       write (*,*) 'vehicle is about to go away, t=',t
c      endif
c

c     Calculate Drag acceleration (/m)
c     drag=.5*CdArhoV^2, s=.5rhoV^2; A=surface area, Sarea
c     area 2690 ft^2 or 249.9091776 m^2

      Cd = 1d0
      s = .5d0*rho*Vsq
      Sarea = (249.9091776d0/6378145d0**2.d0)

c     10 Dec, calc T,aoa, beta for constant V,gam,psi

c aoa
c      aoa = datan(((r*g*m*cosgam*cosbeta)+(2.d0*omega*V*r*m*cosphi*
c     - sinpsi*cosbeta)-(Vsq*m*cosgam*cosbeta))/
c     - ((g*r*m*singam)+(Cd*r*Sarea*s)))

c      write (*,*) 'aoa = ',aoa/.0174532925199d0

c beta
c      beta = datan(((V*m*2.d0*omega*cosphi*cosphi*cospsi*singam*r)-
c     - (V*m*2.d0*omega*sinphi*cosgam*cosphi*r)-
c     - (Vsq*m*cosgam*cosgam*sinpsi*sinphi))/((cosphi*r*g*m*singam)+
c     - (cosphi*r*Cd*Sarea*s)))

c      write (*,*) 'beta = ',beta/.0174532925199d0

c Thrust single pass stop after this single calc

c      Thrust = dsqrt((((g*m*singam)+(Cd*Sarea*s))**2.d0)+(((g*m*cosgam)+
c     - (m*2.d0*omega*V*cosphi*sinpsi)-((m*Vsq*cosgam)/r))**2.d0)+
c     - (((2.d0*omega*V*m*cosphi*cospsi*singam)-(2.d0*omega*V*m*
c     - sinphi*cosgam)-((Vsq*m*cosgam*cosgam*sinpsi*tanphi)/r))**2.d0))
```

```
c        write (*,*) 'Thrust = ',Thrust

c        stop

c       calculate the equations of motion
c

        f(1,nxt) = V*singam
        f(2,nxt) = V*cosgam*secphi*sinpsi/r
        f(3,nxt) = V*cosgam*cospsi/r

        f(4,nxt) = Thrust*cosaoa*cosbeta/m - g*singam - (Cd*Sarea*s/m)

        f(5,nxt) = (-(g*cosgam) + Vsq*cosgam/r +
     -  Thrust*sinaoa/m - 2.d0*omega*V*cosphi*sinpsi)/V

c        write (*,*) 'Change in gam (rads) is = ', f(5,nxt)

        f(6,nxt) = (Thrust*cosaoa*sinbeta/(m*cosgam) -
     -  2.d0*omega*V*(-sinphi + cosphi*cospsi*tangam) +
     -  Vsq*cosgam*sinpsi*tanphi/r)/V

        f(7,nxt) = mdot


        return
        end
```

NOMINAL.IN

```
1.0000223180900    4.8763752299700    0.4993081557320    4.00171110043D-03
    1.5675374480100    .854798459035      3.429358472300D-19
0.0D0 0.743667785562d0
200 50

c 126.1 secs tf = 1.5629417959900D-1
INITIAL CONDITIONS NOMINAL MODEL PAD A
c gam = 89.81328 = 1.5675374480100
c psi = 48.9763440371 = .854798459035c

Thrust =    4.886402159338922E-019
Initial State Vector Shuttle/LFBB
r is          142.348014143586600 meters
theta is      279.395722163719000 degrees
phi is         28.608250221514440 degrees
v is           31.634999535114840 meters/sec
gamma is       89.813280695407310 degrees
psi is         48.976344416380610 degrees
m is     2049384.623046480000000 kgs mass

FINAL CONDITIONS
Thrust =    5.679176775602426E-019
State Vector =
       1.007655076737892        4.881694186245593    5.046423886511117E-001
  2.099236787992863E-001   5.538951019324256E-001   7.191380183123132E-001
  1.429195697770823E-019

r is        48825.189420403640000 meters
theta is      279.700473746117200 degrees
phi is         28.913879033181010 degrees
v is         1659.523991600786000 meters/sec
gamma is       31.735851633775240 degrees
psi is         41.203573336799470 degrees
m is      854087.348987843800000 kgs mass
Inclination is         51.604145012591810 degrees
Current time=       126.100000000078300
Current Dyn Press in psf=       35.045760082614790
```

INTEGRATOR: HAMING.FOR

```
c     14 Oct mods T Miller
c
      subroutine haming(nxt)
c
c     haming is an ordinary differential equations integrator
c     it is a fourth order predictor-corrector algorithm
c     which means that it carries along the last four
c     values of the state vector, and extrapolates these
c     values to obtain the next value (the prediction part)
c     and then corrects the extrapolated value to find a
c     new value for the state vector.
c
c     the value nxt in the call specifies which of the 4 values
c     of the state vector is the "next" one.
c     nxt is updated by haming automatically, and is zero on
c     the first call
c
c     the user supplies an external routine rhs(nxt) which
c     evaluates the equations of motion
c
      common /ham/ x,y(7,4),f(7,4),errest(7),n,h,hh,mode,loop(7,1)
      double precision x,y,f,errest,h,hh,xo,tol,loop
c      double precision yerr
c
      common /debug/ idebug,ig,iidebug
c
c     all of the good stuff is in this common block.
c     x is the independent variable ( time )
c     y(7,4) is the state vector- 4 copies of it, with nxt
c             pointing at the next one
c     f(7,4) are the equations of motion, again four copies
c             a call to rhs(nxt) updates an entry in f
c     errest is an estimate of the truncation error - normally not
c             used
c     n is the number of equations being integrated - 7 ,7 no mass here
c     h is the time step
c     mode is 0 for just EOM, 1 for both EOM and EOV
c
c      write(*,*)'f in haming ',f
      tol = 0.000000001d+00
c     switch on starting algorithm or normal propagation
      if(nxt) 190,10,200

c
c     this is hamings starting algorithm....a predictor - corrector
c     needs 4 values of the state vector, and you only have one- the
c     initial conditions.
c     haming uses a Picard iteration (slow and painfull) to get the
c     other three.
c     if it fails, nxt will still be zero upon exit, otherwise
c     nxt will be 1, and you are all set to go
c
   10 xo = x
```

```
      write (*,*) 'ham init ok, nxt=0'
      hh = h/2.0d+00
      call rhs(1)
      do 40 l = 2,4
      x = x + hh
      do 20 i = 1,n
   20 y(i,l) = y(i,l-1) + hh*f(i,l-1)
      call rhs(l)
      x = x + hh
      do 30 i = 1,n
   30 y(i,l) = y(i,l-1) + h*f(i,l)
   40 call rhs(l)
      jsw = -10
   50 isw = 1
      do 120 i = 1,n
      hh = y(i,1) + h*( 9.0d+00*f(i,1) + 19.0d+00*f(i,2)
     1          - 5.0d+00*f(i,3) + f(i,4) ) / 24.0d+00
      if( dabs( hh - y(i,2)) .lt. tol ) go to 70
      if(y(i,2) .ne. 0.d0) then
        if( dabs( (hh-y(i,2))/y(i,2) ) .lt. tol ) go to 70
      endif
c temp
      if(idebug .ne. 0) then
          write(*,*) 'problem state variable is', i
      endif
c end temp
      isw = 0
   70 y(i,2) = hh
      hh = y(i,1) + h*( f(i,1) + 4.0d+00*f(i,2) + f(i,3))/3.0d+00
      if( dabs( hh-y(i,3)) .lt. tol ) go to 90
      if( y(i,3) .ne. 0.d0) then
        if( dabs( (hh-y(i,3))/y(i,3) ) .lt. tol ) go to 90
      endif
c temp
      if(idebug .ne. 0) then
          write(*,*) 'problem state variable is', i
      endif
c end temp
      isw = 0
   90 y(i,3) = hh
      hh = y(i,1) + h*( 3.0d+00*f(i,1) + 9.0d+00*f(i,2) + 9.0d+00*f(i,3)
     1          + 3.0d+00*f(i,4) ) / 8.0d+00
      if( dabs(hh-y(i,4)) .lt. tol ) go to 110
      if( y(i,4) .ne. 0.d0 ) then
         if( dabs( (hh-y(i,4))/y(i,4) ) .lt. tol) go to 110
      endif
c temp
      if(idebug .ne. 0) then
          write(*,*) 'problem state variable is', i
      endif
c end temp
      isw = 0
  110 y(i,4) = hh
  120 continue
      x = xo
      do 130 l = 2,4
```

```
      x = x + h
  130 call rhs(1)
      if(isw) 140,140,150
  140 jsw = jsw + 1
      if(jsw) 50,280,280
  150 x = xo
      isw = 1
      jsw = 1
      do 160 i = 1,n
  160 errest(i) = 0.0
      nxt = 1
      go to 280
  190 jsw = 2
      nxt = iabs(nxt)
c
c     this is hamings normal propagation loop -
c
  200 x = x + h
      np1 = mod(nxt,4) + 1
      go to (210,230),isw
c     permute the index nxt modulo 4
  210 go to (270,270,270,220),nxt
  220 isw = 2
  230 nm2 = mod(np1,4) + 1
      nm1 = mod(nm2,4) + 1
      npo = mod(nm1,4) + 1
c
c     this is the predictor part
c
      do 240 i = 1,n
      f(i,nm2) = y(i,np1) + 4.0d+00*h*( 2.0d+00*f(i,npo) - f(i,nm1)
     1          + 2.0d+00*f(i,nm2) ) / 3.0d+00
  240 y(i,np1) = f(i,nm2) - 0.925619835*errest(i)
c
c     now the corrector - fix up the extrapolated state
c     based on the better value of the equations of motion
c
      call rhs(np1)
      do 250 i = 1,n
      y(i,np1) = ( 9.0d+00*y(i,npo) - y(i,nm2) + 3.0d+00*h*( f(i,np1)
     1          + 2.0d+00*f(i,npo) - f(i,nm1) ) ) / 8.0d+00
      errest(i) = f(i,nm2) - y(i,np1)
  250 y(i,np1) = y(i,np1) + 0.0743801653 * errest(i)
      go to (260,270),jsw
  260 call rhs(np1)
  270 nxt = np1
  280 continue
c
      return
      end
```

SUBROUTINE AERO.FOR

```
c
      subroutine aero( alpha, Kn, Cd, Cl, Cdp, Clp )
c
c     aerodynamic model for the shuttle, Blanchard et al, JSR 31, 550
c     converted to angle of attack alpha in radians; outputs Cd, Cl not
c     Ca, Cn; also returns first derivatives.
c
      double precision alpha,Kn,Cd, Cl, Cdp, Clp
      double precision Cnc, Cac, Cnf, Caf, Cncp, Cacp, Cnfp, Cafp
      double precision Cnbar, Cabar,alpha2,alpha3
      double precision Ca,Cn,Cap,Cnp,Cdf,Clf
c
      alpha2 = alpha * alpha
      alpha3 = alpha2 * alpha
c
c     hypersonic continuum
c
      Cnc = -0.839782d0 + 3.0012d0 * alpha - 0.303891d0 * alpha2
      Cac = -0.0086314d0 + 0.190247d0 * alpha - 0.220613d0 * alpha2
     1     + 0.110351d0 * alpha3
c
      Cncp = 3.0012d0 - 0.607781d0*alpha
      Cacp = 0.190247d0 - 0.441227d0 * alpha + 0.331053d0 * alpha2
c
c     free molecular flow
c
      Cnf = 0.00158739d0 + 0.526217d0 * alpha + 3.17184d0 * alpha2
     1     - 1.34772d0 * alpha3
      Caf = 0.751105d0 + 0.944601d0 * alpha + 1.94409d0 * alpha2
     1     - 2.20399d0 * alpha3
c
      Cnfp = 0.526217d0 + 6.34367d0 * alpha - 4.04317d0 * alpha2
      Cafp = 0.944601d0 + 3.88819d0 * alpha - 6.61198d0 * alpha2
c
c     bridging coefficients
c
      if( dlog10( Kn ) .lt. 1.3849d0 ) then
          Cnbar = dexp( -0.29981d0 * ( 1.3849d0 - dlog10( Kn ) )**
     1                1.7128d0 )
      else
          Cnbar = 1.d0
      endif
c
      if( dlog10( Kn ) .lt. 1.2042d0 ) then
          Cabar = dexp( -0.2262d0 * ( 1.2042d0 - dlog10( Kn ) )**
     1                1.8410d0 )
      else
          Cabar = 1.d0
      endif
c
c     merged normal and axial coefficients
c
      Cn = Cnc + ( Cnf - Cnc )*Cnbar
```

```
      Ca = Cac + ( Caf - Cac )*Cabar
c
      Cnp = Cncp + ( Cnfp - Cncp ) * Cnbar
      Cap = Cacp + ( Cafp - Cacp ) * Cabar
c
C DEBUG
c     write (1,*) alpha*57.29577d0, Cn
c     write (2,*) alpha*57.29577d0, Ca
C END
c
c     convert to Cl, Cd
c
      Cd =  Ca * dcos(alpha) + Cn * dsin(alpha)
      Cl = -Ca * dsin(alpha) + Cn * dcos(alpha)
c
      Cdp = Cap * dcos(alpha) + Cnp * dsin(alpha)
     1      -Ca * dsin(alpha) +  Cn * dcos(alpha)
      Clp = -Cap* dsin(alpha) + Cnp * dcos(alpha)
     1      -Ca * dcos(alpha) - Cn  * dsin(alpha)
c
      return
      end
```

Appendix 2. Abort Model Fortran Code

This Appendix contains the Fortran code for the Abort Trajectory model. Depicted is the solution for the 1 second abort (L+10). Target: Savannah International Airport. Files listed for this model include: Abort.for, Abo_rhs.for, Atm.for, and Abort.in. See Appendix 1 for the code listings for Haming.for and Aero.for, which are also part of the Abort Model.

ABORT.FOR

```
c Capt. Thomas Miller, 19 Feb 1999

      program abort

c*************************************c
c Input file for this program is:     c
c  a 7 component state vector,        c
c  initial, and final times (t0,tf)   c
c  integration steps (nstp,nskp)      c
c*************************************c


      implicit double precision (a - h)
      implicit double precision (o - z)

      common /flags/ mass_flag,thrst_flag,d_psi,d_gam,massref,wdotref
      double precision mass_flag,thrst_flag,d_psi,d_gam,massref,wdotref

      common /flags2/ thrstref,sref,timeref,aoaref,betaref,dv_max2
      double precision thrstref,sref,timeref,aoaref,betaref,dv_max2

      common /flags3/ psi_flag,spin_flag,spin,betapsi,dif,pi
      double precision psi_flag,spin_flag,spin,betapsi,dif,pi

      common /flags4/ shut_prt,lfsh_prt,prt_all,dist_go,psi_cor
      double precision shut_prt,lfsh_prt,prt_all,dist_go,psi_cor

      common /flags5/ long_land,long_shut,lat_land,lat_shut
      double precision long_land,long_shut,lat_land,lat_shut

      common /ham/ t,x(7,4),f(7,4),err(7),n,h,hh,mode,loop(7,1)
      double precision t,x,f,err,hh,h,loop

      common /amat/ a(15,15),hamil,ithrot,igt,omega,gam_flag
      double precision a,hamil,omega,gam_flag
```

```
      common /ctrl/ aoa,beta,mdot,dV,wdotLFBB,wdotshut,numstep,n_stp
      double precision aoa,beta,mdot,dV,wdotLFBB,wdotshut,numstep,n_stp

      common /maxq/ s,wdot_lf,wdot_sh,shut_dV,dv_min,dv_max,nnn_stp
      double precision s,wdot_lf,wdot_sh,shut_dV,dv_min,dv_max

      double precision t0,tf,gs
      double precision xic(7)

      common /debug/ idebug,ig,iidebug

      common /Thrust/ Thrust,wdot_flag,wdot,minT,maxT,num,orig_dV,dgam
      double precision Thrust,wdot_flag,wdot,minT,maxT,orig_dV,dgam

      common /pullup/ Ht,Kl,Ho,B,rho,Sarea,Cl,m
      double precision Ht,Kl,Ho,B,rho,Sarea,Cl,m

      real nstp,nskp

c**************************c
c Output files             c
c**************************c

      open(1,FILE='c:\adyn.dat',STATUS='UNKNOWN')
      open(2,FILE='c:\ags.dat',STATUS='UNKNOWN')
      open(3,FILE='c:\aHX.dat',STATUS='UNKNOWN')
      open(4,FILE='c:\agam.dat',STATUS='UNKNOWN')
      open(5,FILE='c:\apull.dat',STATUS='UNKNOWN')
      open(6,FILE='c:\ag.dat',STATUS='UNKNOWN')

      write (6,*) 'Shuttle/LFBB RTLS Abort Code'
      write (6,*) 'Capt. Thomas L. Miller Jr.  '
      write (6,*) ' '

c*********************************c
c Read in initial state vector    c
c*********************************c

      read (*,*) (xic(ii),ii=1,7)

c**************************************c
c Read in times and steps, tf 600.0 sec c
c**************************************c

      read (*,*) t0,tf
      read (*,*) nstp,nskp

c****************************************************************c
c minT & maxT are for dV = 0 and dV = 1.0244 for LFBB & Shut      c
c combined thrust levels min/max along with number of iterations. c
c****************************************************************c

      read (*,*) minT,maxT,num
      write (6,*) 'Combined Thrust Vdot; minT, maxT, # of increments'
      write (6,*) minT,maxT,num
```

```
      read (*,*) dv_min,dv_max,nnn_stp
      write (6,*) 'Shuttle-Only Vdot; Vdot min,Vdot max, # increments'
      write (6,*) dv_min,dv_max,nnn_stp

      read (*,*) iidebug
      write (6,*) 'Debug set to ',iidebug
      write (6,*) ' '

      if (iidebug .eq. 4) then
      write (6,*) '+++++++++++++++++++++++++++++++++++++++++++++++'
      write (6,*) 'Initial State Vector Shuttle/LFBB for Abort'
      write (6,*) '-----------------------------------------------'
      write (6,*) 'r is    ',((xic(01)*6378145)-6378145), ' meters'
      write (6,*) 'theta is',(xic(02)/.0174532925199), ' degrees'
      write (6,*) 'phi is  ',(xic(03)/.0174532925199), ' degrees'
      write (6,*) 'v is    ',(xic(04)*7905.36828), ' meters/sec'
      write (6,*) 'gamma is',(xic(05)/.0174532925199), ' degrees'
      write (6,*) 'psi is  ',(xic(06)/.0174532925199), ' degrees'
      write (6,*) 'm is    ',(xic(07)*5.976D+24), ' kgs mass'
      write (6,*) '+++++++++++++++++++++++++++++++++++++++++++++++'
      write (6,*) ' '
      endif

c***********************************c
c Outer Loop for Throttle up time   c
c***********************************c

c******************c
c Savannah Solution c
c******************c

      do 160 n_time = 43,43


      write (6,*) 'Throttle Time set to, ',n_time
      write (6,*) ' '
      dV = maxT

c*******************c
c Initial Conditions c
c*******************c

      gam_flag = 0.d0
      d_gam = 0.1d0
      d_psi = 0.53d0*(-14.0815236524d0)
      wdot_flag = 0.d0
      mass_flag = 0.d0
      thrst_flag = 0.d0
      psi_flag = 0.d0
      spin_flag = 0.d0
        pi=3.14159265359d0

c***********************************c
c Initally set wdot's to max.       c
```

```
c**************************************c

      wdotLFBB = 1.5731759133d-18
      wdotshut = 1.33147192403d-19


c******************************************c
c Number of ode's, initial angle, step  c
c******************************************c

      n = 7
        do 10 i = 1,7
          x(i,1) = xic(i)

   10 continue

      t = t0

c**************************************************c
c Time incr 600/(299*50) to eval every 2 secs  c
c**************************************************c

      h = (tf-t0)/dble(nstp*nskp)

c**************************************************c
c nxt = 0 is hamings initialization flag...    c
c**************************************************c

      nxt = 0

c*********************c
c Initalize haming  c
c*********************c

      call haming(nxt)
      if(nxt .eq. 0) go to 160
        do 50 i = 1,nstp
          do 19 j = 1,nskp
            call haming(nxt)
   19 continue

c************************************************************************c
c  Time, alt, Pres lbf/ft^2(N to lbf then m^2 to ft^2);               c

c 1 atm = 2116.2166 lb/ft^2 = 101325 N/m^2                             c
c  Various output files.  Also check acceleration force or g's        c
c************************************************************************c

      write (1,*) (t*806.8118744d0),((x(1,nxt)*6378145d0)-
     1 6378145d0),(s*30.0618114811D+9)

      gs = ((((x(4,nxt)**2)/1.09912835158d-3)*(1.d0-dcos(x(5,nxt))))/
     1 1.00085427871d0)
      write (2,*) (t*806.8118744d0),((x(1,nxt)*6378145d0)-
     1 6378145d0),gs
```

```
c****************************************c
c Write out current time's state vector   c
c****************************************c

c       write (2,*) t*806.8118744d0, ' seconds'
c       write (2,*) x(1,nxt),x(2,nxt),x(3,nxt),x(4,nxt),x(5,nxt),
c      + x(6,nxt),x(7,nxt)
c       write (2,*) t,tf
c       write (2,*) nstp,nskp
c       write (2,*) ' '

c************************************************************c
c This outputs to a file alt_range.dat secs,alt m, down range m   c
c************************************************************c

c       write (3,*) t*806.8118744d0, ((x(1,nxt)*6378145d0)-
c      - 6378145d0), ((x(4,nxt)*dcos(x(5,nxt))*
c      - dsin(x(6,nxt)))*6378145d0*t)

c***********c
c Gamma out  c
c***********c

      write (4,*) (t*806.8118744d0),((x(1,nxt)*6378145d0)-
     - 6378145d0), (x(5,nxt)/.0174532925199d0)

c****************************************************c
c If Intercept Ground or Calc Neg Thrust Loop to Start c
c****************************************************c

      if (x(1,nxt) .lt. 1.d0) then
      go to 160
      endif

      if (Thrust .lt. 0.d0) then
      go to 160
      endif

c**********************************************c
c Set up to calc distance and heading to target. c
c**********************************************c

      lat_shut = x(3,nxt)
      long_shut = x(2,nxt)

c**********************************************************c
c Savannah runway Lat = 32.1172 deg, Long = 278.800066667 deg c
c**********************************************************c
      lat_land = 0.56055088652d0
      long_land = 4.8659791181d0

c**********************************************************c
c Charleston S.C. runway Lat = 32.8916 N, Long = 279.967 deg E c
c**********************************************************c
c       lat_land = 0.574066716248d0
c       long_land = 4.88636281844d0
```

```
c************************c
c JAX, Jacksonville, FL. c
c************************c
c       lat_land = 0.531979264889d0
c       long_land = 4.85759746526d0

c*****************c
c BGR, Bangor, ME. c
c*****************c
c       lat_land = 0.78205382166d0
c       long_lang = 5.08213116464d0

      dist_go = dacos(cos((pi/2.d0)-lat_land)*cos((pi/2.d0)-lat_shut)+
     1 sin((pi/2.d0)-lat_land)*sin((pi/2.d0)-lat_shut)*cos(long_shut -
     2 long_land))

      psi_cor = (-1.d0*(dasin((sin(pi/2.d0)-lat_land)*(sin(long_shut-
     1 long_land)/sin(dist_go)))))

      if (iidebug .eq. 4) then
      write (6,*) '****************************************************'
      write (6,*) 'Shuttle dV for this run   = ',dv_max
      write (6,*) 'Combined dV this run was = ',maxT
      write (6,*) ' '
      write (6,*) 'Thrust = ',Thrust
      write (6,*) ' '
      write (6,*) 'Distance to Runway in miles = ',((dist_go*6378145d0)/
     1 1609.344d0)
      write (6,*) 'Heading correction (act-this)degs = ',(-1.d0*psi_cor/
     1 .0174532925199d0)
      if (wdot_flag .eq. 2.d0) then
      if ((s*30.0618114811D+9) .gt. 20.d0)then
      write (6,*) 'ET Sep s>20 begin pull up'
      gs = ((((x(4,nxt)**2)/1.09912835158d-3)*(1.d0-dcos(x(5,nxt))))/
     1 1.00085427871d0)
      write (6,*) 'g s at bottom of pull up = ',gs
      endif
      endif
      write (6,*) ' '
      write (6,*) 'r is     ',((x(1,nxt)*6378145d0)-6378145d0), ' meters'
      write (6,*) 'theta is',(x(2,nxt)/.0174532925199d0), ' degrees'
      write (6,*) 'phi is   ',(x(3,nxt)/.0174532925199d0), ' degrees'
      write (6,*) 'v is     ',(x(4,nxt)*7905.36828d0), ' meters/sec'
      write (6,*) 'gamma is',(x(5,nxt)/.0174532925199d0), ' degrees'
      write (6,*) 'psi is   ',(x(6,nxt)/.0174532925199d0), ' degrees'
      write (6,*) 'm is     ',(x(7,nxt)*5.976D+24), ' kgs mass'
      write (6,*) 'Seconds into flight ',int((t*806.8118744d0))
      write (6,*) 'Dyn Press psf is     ',int((s*30.0618114811D+9))
      write (6,*) ' '
      endif

      if ((t*806.8118744d0) .ge. n_time) then
      if ((s*30.0618114811d+19) .gt. 2.d0) then
      thrst_flag = 0.d0
      goto 100
```

```
      endif
      endif

      if ((t*806.8118744d0) .ge. n_time) then
      if ((s*30.0618114811d+19) .lt. 2.d0) then
      thrst_flag = 1.d0
      goto 100
      endif
      endif

      if ((s*30.0618114811d+19) .lt. 2.d0) then
      thrst_flag = 1.d0
      go to 100
      endif

c**************************************************c
c                                                  c
c STAGING when mass reaches 901,696kg LFBB is empty  c
c                                                  c
c**************************************************c
      if (mass_flag .eq. 0.d0) then
      if(x(7,nxt)+mdot*hh .lt. 1.50886257082d-19) then
      wdot_flag = 1.d0
      mass_flag = 1.d0
      write (6,*) 'Staged lfbb, mass_flag = ', mass_flag
      write (6,*) (x(7,nxt)*5.976D+24), ' kgs mass'

      do 20 k = 1,6
          x(k,1) = x(k,nxt)
   20 continue

c*****************************************c
c Re-initalize Haming for staging event   c
c*****************************************c

      nxt = 0

c***************************c
c New Shuttle/ET-only mass   c
c***************************c

      x(7,1) = 1.13322121821d-19

      call haming(nxt)

      if (nxt .eq. 0) then
      write (6,*) 'nxt is zero 919',nxt
      endif

      if(nxt .eq. 0) go to 160

c*******************************************c
c 1st loop Shuttle Sep but before time jump c
c*******************************************c

c  Integrating...
```

```
      do 25 l = 1,nstp
         do 22 m = 1,nskp
            call haming(nxt)
   22 continue

c****************************************************c
c If alt or thrust goes negative restart with new value c
c****************************************************c

      if (x(1,nxt) .lt. 1.d0) then
      go to 160
      endif

      if (Thrust .lt. 0.d0) then
      go to 160
      endif

      if ((t*806.8118744d0) .ge. n_time) then
      if ((s*30.0618114811d+19) .gt. 2.d0) then
      thrst_flag = 0.d0
      goto 100
      endif
      endif

      if ((t*806.8118744d0) .ge. n_time) then
      if ((s*30.0618114811d+19) .lt. 2.d0) then
      thrst_flag = 1.d0
      goto 100
      endif
      endif

      if ((s*30.0618114811d+19) .lt. 2.d0) then
      thrst_flag = 1.d0
      go to 100
      endif

   25 continue
      endif
      endif

   50 continue

      stop
c*******************************************************************c

c*******************************************************************c
c Throttle down: Shuttle dV outer loop, LFBB/SHUT dV inner loop.       c
c Size of Shuttle dV slice.  Max to min value of shut_dV.              c
c*******************************************************************c

c*******************************************************************c

  100 n_stp = ((dv_max+dv_min)/dble(nnn_stp))

c*********************************************c
```

```
c Tools to help re-initiate vals as loop cycles   c
c************************************************c

      wdot_lf = (1.5731759133d-18 - 8.05465584758d-19)/dble(num)
      wdot_sh = (1.33147192403d-19 - 0.637067906233d-19)/dble(num)

      wdotref = wdot_flag
      massref = mass_flag
      thrstref = thrst_flag
      timeref = t
      sref = s
      aoaref = aoa
      betaref = beta

      do 105 ik = 1,7
       loop(ik,1) = x(ik,nxt)
  105 continue

c*******************************************c
c Outer Loop for Shuttle-Only dV High to low c
c use (nnn_stp),1,-1; Savannah Solution = 13 c
c*******************************************c

      do 150 nnn = 13,1,-1
c      (nnn_stp),1,-1
       shut_dV = (n_stp*dble(nnn))

      if (nnn .eq. 12)then
      stop
      endif

      wdot_flag = wdotref
      mass_flag = massref
      thrst_flag = thrstref
      s = sref
      t = timeref
      aoa = aoaref
      beta = betaref
      psi_flag = 0.d0
      spin_flag = 0.d0

      wdotshut = ((0.637067906233d-19 + wdot_sh*dble(nnn)))

      numstep = (maxT-minT)/dble(num)

c********************************c
c Inner Loop for LFBB/Shuttle dV  c
c Savannah Solution = 5           c
c********************************c
      do 140 nn = 5,num
c 1,num

      if (nn .eq. 6) then
      stop
      endif
```

```
      wdot_flag = wdotref
      mass_flag = massref
      thrst_flag = thrstref
      s = sref
      t = timeref
      aoa = aoaref
      beta = betaref
      psi_flag = 0.d0
      spin_flag = 0.d0

c**************************************c
c Builds up dV, low to highest value.  c
c**************************************c

      dV = ((numstep*dble(nn)))
      orig_dV = dV

      wdotLFBB = ((8.05465584758d-19 + wdot_lf*dble(nn)))

      n = 7

      do 110 kk = 1,7
          x(kk,1) = loop(kk,1)
  110 continue

      nxt = 0

      call haming(nxt)

      if (nxt .eq.0) then
      write (6,*) 'nxt is zero',nxt
      endif

      if(nxt .eq. 0) go to 140
        do 135 l = 1,nstp
         do 119 m = 1,nskp
            call haming(nxt)
  119 continue

      write (4,*) (t*806.8118744d0),((x(1,nxt)*6378145d0)-
     - 6378145d0), (x(5,nxt)/.0174532925199d0)

c**************************************************************c
c Time, , alt, Pres lbf/ft^2(N to lbs then m^2 to ft^2);       c
c 1 atm = 2116.2166 lb/ft^2 = 101325 N/m^2                     c
c Various output files                                         c
c**************************************************************c

      write (1,*) (t*806.8118744d0),((x(1,nxt)*6378145d0)-
     1 6378145d0),(s*30.0618114811D+9)

      gs = ((((x(4,nxt)**2)/1.09912835158d-3)*(1.d0-dcos(x(5,nxt))))/
     1 1.00085427871d0)
      write (2,*) (t*806.8118744d0),((x(1,nxt)*6378145d0)-
     1 6378145d0),gs
```

```
      if (x(1,nxt) .lt. 1.0d0) then
       go to 140
      endif

      if (Thrust .lt. 0.d0) then
      go to 140
      endif

      lat_shut = x(3,nxt)
      long_shut = x(2,nxt)

c**********************************************************c
c Savannah runway Lat = 32.1172 deg, Long = 278.800066667 deg c
c**********************************************************c
      lat_land = 0.56055088652d0
      long_land = 4.8659791181d0

c***********************************************************c
c Charleston S.C. runway Lat = 32.8916 N, Long = 279.967 deg E c
c***********************************************************c
c       lat_land = 0.574066716248d0
c       long_land = 4.88636281844d0

c************************c
c JAX, Jacksonville, FL. c
c************************c
c       lat_land = 0.531979264889d0
c       long_land = 4.85759746526d0

c*****************c
c BGR, Bangor, ME. c
c*****************c
c       lat_land = 0.78205382166d0
c       long_lang = 5.08213116464d0

      dist_go = dacos(cos((pi/2.d0)-lat_land)*cos((pi/2.d0)-lat_shut)+
     1 sin((pi/2.d0)-lat_land)*sin((pi/2.d0)-lat_shut)*cos(long_shut -
     2 long_land))

      psi_cor = (-1.d0*(dasin((sin(pi/2.d0)-lat_land)*(sin(long_shut-
     1 long_land)/sin(dist_go)))))

      if (iidebug .eq. 4) then
      write (6,*) '*****************************************************'
      write (6,*) 'wdotshut ',wdotshut,'for n_time ',n_time
      write (6,*) 'Shuttle dV for this run  = ',shut_dV
      write (6,*) 'Combined dV this run was = ',orig_dV
      write (6,*) 'In 100, prior to LFBB Sep '
      write (6,*) 'Thrust = ',Thrust
      write (6,*) ' '
      write (6,*) 'Distance to Runway in miles = ',((dist_go*6378145d0)/
     1 1609.344d0)
      write (6,*) 'Heading correction (act-this)degs = ',(-1.d0*psi_cor/
     1 .0174532925199d0)
      if (wdot_flag .eq. 2.d0) then
      if ((s*30.0618114811D+9) .gt. 20.d0)then
```

```
      write (6,*) 'ET Sep s>20 begin pull up'
      gs = ((((x(4,nxt)**2)/1.09912835158d-3)*(1.d0-dcos(x(5,nxt))))/
     1 1.00085427871d0)
      write (6,*) 'g s at bottom of pull up = ',gs
      endif
      endif
      write (6,*) ' '
      write (6,*) 'r is     ',((x(1,nxt)*6378145d0)-6378145d0), ' meters'
      write (6,*) 'theta is',(360.d0 -(x(2,nxt)/.0174532925199d0)),
     1 ' Degrees West Longitude'
      write (6,*) 'phi is   ',(x(3,nxt)/.0174532925199d0), ' degrees Lat'
      write (6,*) 'v is     ',(x(4,nxt)*7905.36828d0), ' meters/sec'
      write (6,*) 'gamma is',(x(5,nxt)/.0174532925199d0), ' degrees'
      write (6,*) 'psi is   ',(x(6,nxt)/.0174532925199d0), ' degrees'
      write (6,*) 'aoa is ', aoa/.0174532925199d0
      write (6,*) 'beta is ', beta/.0174532925199d0
      write (6,*) 'm is     ',(x(7,nxt)*5.976D+24), ' kgs mass'
      write (6,*) ' '

c**********************************************************************c
c Inclination based on inertial psi (takes earth rotation into account) c
c**********************************************************************c

c      write (6,*) 'Inclination is ',datan(((7.292116d-05*806.8118744d0)*
c     1 x(1,nxt)*dcos(x(3,nxt))+x(4,nxt)*dcos(x(5,nxt))*dsin(x(6,nxt)))/
c     2 (x(4,nxt)*dcos(x(5,nxt))*dcos(x(6,nxt))))/.0174532925199d0,
c     3 ' degrees'
      write (6,*) 'Seconds into flight ',int((t*806.8118744d0))
      write (6,*) 'Dyn Press psf is    ',int((s*30.0618114811D+9))
      write (6,*) ' '
      endif

      if (s*30.0618114811D+9 .lt. 2.d0) then
      thrst_flag = 1.d0
      endif

c******************************************c
c Helicopter if gamma angle reaches value   c
c gam flag passed to Rhs so gam dot set = 0 c
c******************************************c

      if (x(5,nxt) .lt. 4.36332312998d-3) then
c 13 deg 226.892802759d-3)then
c 15 deg 261.799387799d-3) then
c 14 deg down 244.346095279d-3) then
c 0 deg      4.36332312998d-3) then
      gam_flag = 1.0d0
      endif

c+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++c
c ET ready before LFBB need to drop LFBB first                        c
c*********************************************************************c
c This is loop where thrst flag = 1 and s goes above 3, must          c
c drop LFBB and then ET if mass flag = 0 and these events occur       c
c This is only an issue if trying to steer in to JAX, drop LFBB early c
c*********************************************************************c
```

```
      if (thrst_flag .eq. 1.d0) then
       if (s*30.0618114811D+9 .gt. 2.d0) then
       if (mass_flag .eq. 0.d0) then
      wdot_flag = 2.d0
      write (1,*) 's is climbing drop LFBB then ET'
      write (2,*) 's is climbing drop ET'
      write (6,*) ' '
      write (6,*) 'c*********************************************c'
      write (6,*) 'c*********************************************c'
      write (6,*) 'Dynamic Pressure is climbing; dropping LFBB & ET'
      write (6,*) 'c*********************************************c'
      write (6,*) 'c*********************************************c'
      write (6,*) ' '
      x(7,nxt) = 0.174575376673d-19
      mass_flag = 2.d0
c Go 40 deg aoa if ET drop
      aoa = .698131700796d0
      do 120 kki = 1,7
          x(kki,1) = x(kki,nxt)
  120 continue

      nxt = 0

      call haming(nxt)

      if (nxt .eq. 0) then
      write (6,*) 'nxt is zero Line 760',nxt
      endif

      if(nxt .eq. 0) go to 140

c************c
c Helicopter c
c************c

      if (x(5,nxt) .lt. 4.36332312998d-3) then
c 13 deg 226.892802759d-3)then
c 15 deg 261.799387799d-3) then
c 14 deg 244.346095279d-3) then
c 0 deg     4.36332312998d-3) then
      gam_flag = 1.0d0
      endif

c  Integrating...
      do 125 ll = 1,nstp
         do 122 mm = 1,nskp
            call haming(nxt)
  122 continue

      write (4,*) (t*806.8118744d0),((x(1,nxt)*6378145d0)-
     - 6378145d0), (x(5,nxt)/.0174532925199d0)

      if (s*30.0618114811D+9 .lt. 2.d0) then
```

```
      thrst_flag = 1.d0
      endif

c*************************************************************c
c  Time, , alt, Pres lbf/ft^2(N to lbf then m^2 to ft^2);     c
c 1 atm = 2116.2166 lb/ft^2 = 101325 N/m^2                    c
c  Various output files                                       c
c*************************************************************c

      write (1,*) (t*806.8118744d0),((x(1,nxt)*6378145d0)-
     1 6378145d0),(s*30.0618114811D+9)

      gs = ((((x(4,nxt)**2)/1.09912835158d-3)*(1.d0-dcos(x(5,nxt))))/
     1 1.00085427871d0)
      write (2,*) (t*806.8118744d0),((x(1,nxt)*6378145d0)-
     1 6378145d0),gs

      if (x(1,nxt) .lt. 1.00391963494d0) then
      go to 140
      endif

      if (Thrust .lt. 0.d0) then
      go to 140
      endif

      lat_shut = x(3,nxt)
      long_shut = x(2,nxt)

c*************************************************************c
c Savannah runway Lat = 32.1172 deg, Long = 278.800066667 deg c
c*************************************************************c
      lat_land = 0.56055088652d0
      long_land = 4.8659791181d0

c**************************************************************c
c Charleston S.C. runway Lat = 32.8916 N, Long = 279.967 deg E c
c**************************************************************c
c      lat_land = 0.574066716248d0
c      long_land = 4.88636281844d0

c***********************c
c JAX, Jacksonville, FL. c
c***********************c
c      lat_land = 0.531979264889d0
c      long_land = 4.85759746526d0

c*****************c
c BGR, Bangor, ME. c
c*****************c
c      lat_land = 0.78205382166d0
c      long_lang = 5.08213116464d0

      dist_go = dacos(cos((pi/2.d0)-lat_land)*cos((pi/2.d0)-lat_shut)+
     1 sin((pi/2.d0)-lat_land)*sin((pi/2.d0)-lat_shut)*cos(long_shut -
     2 long_land))
```

```
      psi_cor = (-1.d0*(dasin((sin(pi/2.d0)-lat_land)*(sin(long_shut-
     1 long_land)/sin(dist_go)))))

c*****************************************************************c
c Check to see if after dropping ET Orbiter attempts to execute      c
c modified skip reentry, will the bottom of pull-up be above ground?  c
c*****************************************************************c

      if (thrst_flag .ne. 0.d0) then
      if (aoa .gt. 0.d0) then
      if (Cl .gt. 0.d0) then
      Kl = ((Cl*Sarea*rho)/(2.d0*m))
      B = ((Kl*Ho*dexp(-(x(1,nxt)-1.d0)/Ho)) - dcos(x(5,nxt)))
      Ht = (Ho*dlog((Kl*Ho)/(dcos(0.d0)+B)))

      if (Ht .gt. 0.d0) then
c      write (5,*) 'Ho is ',Ho*6378145d0
      write (5,*) ' '
      write (5,*) 'HT bottom of pull up is =',Ht*6378145d0
      write (5,*) 'Kl 940c below pos Ht is ',Kl
      write (5,*) 'B is ',B
      write (5,*) 'Cl is ',Cl
      write (5,*) 'aoa entering pu is ',(aoa/.0174532925199d0)
      write (5,*) 'x5 gam entering pu is ',(x(5,nxt)/.0174532925199d0)
      write (5,*) 'v entering is ',(x(4,nxt)*7905.36828d0), ' m/sec'
      write (5,*) 'Dyn ent Press psf is    ',int((s*30.0618114811D+9))
      write (5,*) 'theta is',(360.d0 -(x(2,nxt)/.0174532925199d0)),
     1 ' Degrees West Longitude'
      write (5,*) 'phi is  ',(x(3,nxt)/.0174532925199d0), ' degrees Lat'
      write (5,*) 'Shuttle dV after LFBB Sep = ',shut_dv
      write (5,*) 'Combined dV this run was = ',orig_dV
      write (5,*) 'wdot is       = ',wdot
      write (5,*) 'wdotflag is   = ',wdot_flag
      write (5,*) 'dV is         = ',dV
      write (5,*) 'Time is = ',n_time
      write (5,*) 'cl ',Cl
      write (5,*) 'alt=',((x(1,nxt)-1.d0)*6378145d0)
      write (5,*) 'Seconds into flight ',(t*806.8118744d0)
      write (5,*) ' '
      endif
      endif
      endif
      endif

c*********************************************c
c Check to see if ET is empty if yes goto Glide c
c*********************************************c

      if(x(7,nxt)+mdot*hh .lt. 0.224670919545d-19) then
       write (6,*) 'Warning! Out of fuel...wdot flag to 2'
c      wdot_flag = 2.d0
       write (6,*) 'wdot flag = ',wdot_flag
c      x(7,nxt) = 0.174575376673d-19
      endif

      lat_shut = x(3,nxt)
```

```
      long_shut = x(2,nxt)

c**********************************************************c
c Savannah runway Lat = 32.1172 deg, Long = 278.800066667 deg c
c**********************************************************c
      lat_land = 0.56055088652d0
      long_land = 4.8659791181d0

c***********************************************************c
c Charleston S.C. runway Lat = 32.8916 N, Long = 279.967 deg E c
c***********************************************************c
c      lat_land = 0.574066716248d0
c      long_land = 4.88636281844d0

c************************c
c JAX, Jacksonville, FL. c
c************************c
c      lat_land = 0.531979264889d0
c      long_land = 4.85759746526d0

c*****************c
c BGR, Bangor, ME. c
c*****************c
c      lat_land = 0.78205382166d0
c      long_lang = 5.08213116464d0

      dist_go = dacos(cos((pi/2.d0)-lat_land)*cos((pi/2.d0)-lat_shut)+
     1 sin((pi/2.d0)-lat_land)*sin((pi/2.d0)-lat_shut)*cos(long_shut -
     2 long_land))

      psi_cor = (-1.d0*(dasin((sin(pi/2.d0)-lat_land)*(sin(long_shut-
     1 long_land)/sin(dist_go)))))

c************************c
c Data Conditional Output c
c************************c

      if (Thrust .ge. 0.d0) then
      if (x(1,nxt) .gt. 1.00391963494d0) then
       if (x(1,nxt) .lt. 1.00783926988176d0)then
c 50 km 783926988176d0) then
c 454677653142d0)then c 29km
c 30km 470356192906d0) then
c 27km        415481303733d0) then

        if (x(5,nxt) .gt. -1.04719755119d0) then
      write (6,*) ' '
      write (6,*) '**************************************************'
      write (6,*) '**************************************************'
      write (6,*) '**************************************************'
      write (6,*) 'Shuttle dV after LFBB Sep = ',shut_dv
      write (6,*) 'Combined dV this run was = ',orig_dV
      write (6,*) 'wdot is      = ',wdot
      write (6,*) 'dV is        = ',dV
      write (6,*) 'Thrust       = ',Thrust
c        write (6,*) 'State Vector = '
```

```
c         write (6,*) (x(k,nxt),k=1,7)
      write (6,*) ' '
      write (6,*) 'Distance to Runway in miles = ',((dist_go*6378145d0)/
     1 1609.344d0)
      write (6,*) 'Heading correction (act-this)degs = ',(-1.d0*psi_cor/
     1 .0174532925199d0)
      if (wdot_flag .eq. 2.d0) then
      if ((s*30.0618114811D+9) .gt. 20.d0)then
      write (6,*) 'ET Sep s>20 begin pull up'
      gs = ((((x(4,nxt)**2)/1.09912835158d-3)*(1.d0-dcos(x(5,nxt))))/
     1 1.00085427871d0)
      write (6,*) 'g s at bottom of pull up = ',gs
      endif
      endif
      write (6,*) ' '
      write (6,*) 'r is    ',((x(1,nxt)*6378145d0)-6378145d0), ' meters'
      write (6,*) 'theta is',(360.d0 -(x(2,nxt)/.0174532925199d0)),
     1 ' Degrees West Longitude'
      write (6,*) 'phi is  ',(x(3,nxt)/.0174532925199d0), ' degrees Lat'
      write (6,*) 'v is    ',(x(4,nxt)*7905.36828d0), ' meters/sec'
      write (6,*) 'gamma is',(x(5,nxt)/.0174532925199d0), ' degrees'
      write (6,*) 'psi is  ',(x(6,nxt)/.0174532925199d0), ' degrees'
      write (6,*) 'aoa is ', aoa/.0174532925199d0
      write (6,*) 'beta is ', beta/.0174532925199d0
      write (6,*) 'm is    ',(x(7,nxt)*5.976D+24), ' kgs mass'
      write (6,*) 'Seconds into flight ',int((t*806.8118744d0))
      write (6,*) 'Dyn Press psf is    ',int((s*30.0618114811D+9))
      write (6,*) 'Throttle Down Time = ',n_time
      write (6,*) ':) '

            endif
           endif
          endif
         endif

      if (x(1,nxt) .lt. 1.0039664197d0) then
      write (6,*) 'Attained 25 km!'
      go to 140
      endif

  125 continue
      endif
      endif
      endif


c*******************************************************************c
c*******************************************************************c
c STAGING when mass reaches 901,696kg LFBB is empty                 c
c*******************************************************************c

      if (mass_flag .eq. 0.d0) then
      if(x(7,nxt)+mdot*hh .lt. 1.50886257082d-19) then
      wdot_flag = 1.d0
c      write (6,*) 'Dropped LFBB 901.696'
      write (6,*) 'c**************************************c'
```

```
      write (6,*) 'c***************************************c'
      write (6,*) '        LFBB Empty; Dropping LFBB         '
      write (6,*) 'c***************************************c'
      write (6,*) 'c***************************************c'
c      write (1,*) 'Dropped LFBB 901 after 100 prior to et'
      write (6,*) ' '
      write (6,*) 'Alt LFBB Drop = ',((x(1,nxt)*6378145d0)-6378145d0)
      write (6,*) 'Thrust for LFBB Drop = ',Thrust
      write (6,*) ' '
      write (2,*) 'Dropped LFBB'

      mass_flag = 1.d0
      do 126 kki = 1,6
          x(kki,1) = x(kki,nxt)
  126 continue

      nxt = 0
c***************************c
c  New Shuttle/ET-only mass  c
c***************************c

      x(7,1) = 1.13322121821d-19

      call haming(nxt)

      if (nxt .eq. 0) then
      write (6,*) 'nxt is zero 1007',nxt
      endif

      if(nxt .eq. 0) go to 140

c  Integrating...
      do 132 ll = 1,nstp
         do 127 mm = 1,nskp
            call haming(nxt)
  127 continue

c*************************************************c
c Modified Helicopter Execute on back side of slope c
c after trajectory peak, when altitude had dropped  c
c to desired level to initiate Helo, gam            c
c change not as drastic as first type of Helicoper  c
c*************************************************c

      if(x(1,1) .lt. 1.013326758799d0)then
cc 310km 1.0486034732669d0)then 5.5 gs max
cc 300km 1.0470356192906d0)then 7.2 gs max
cc 200km 1.031357079527d0) then
cc 160km    1.0250856636216d0) then
      d_gam = -0.23d0
c      d_psi = 0.03d0*(-14.0815236524d0)
      gam_flag = 0.d0
      endif

c*******************************************************c
c Helicopter if previous Mod Helo not used, off otherwise. c
```

```
c*******************************************************c

c      if (x(5,nxt) .lt. 4.36332312998d-3) then
c 13 deg 226.892802759d-3)then
c 15 deg 261.799387799d-3) then
c 14 deg down 244.346095279d-3) then
c 0 deg     4.36332312998d-3) then
c      gam_flag = 1.0d0
c      write (4,*) 'gam at zero, gam flag is= ',gam_flag
c      endif

      write (4,*) (t*806.8118744d0),((x(1,nxt)*6378145d0)-
     - 6378145d0), (x(5,nxt)/.0174532925199d0)

      if (s*30.0618114811D+9 .lt. 2.d0) then
      thrst_flag = 1.d0
      endif

c***************************************************************c
c Time, , alt, Pres lbf/ft^2(N to lbf                            c
c  then m^2 to ft^2); 1 atm = 2116.2166 lb/ft^2 = 101325 N/m^2   c
c  Various output files                                          c
c***************************************************************c

      write (1,*) (t*806.8118744d0),((x(1,nxt)*6378145d0)-
     1 6378145d0),(s*30.0618114811D+9)

      gs = ((((x(4,nxt)**2)/1.09912835158d-3)*(1.d0-dcos(x(5,nxt))))/
     1 1.00085427871d0)
      write (2,*) (t*806.8118744d0),((x(1,nxt)*6378145d0)-
     1 6378145d0),gs

c*******************************************************c
c Drop External Tank when reentering atmosphere, s>2 psf  c
c*******************************************************c

      if (thrst_flag .eq. 1.d0) then
      if (s*30.0618114811D+9 .gt. 2.d0) then
      wdot_flag = 2.d0
      write (1,*) 's is climbing drop inner loop ET'
      write (2,*) 's is climbing drop inner ddd loop ET'
      write (6,*) ' '
      write (6,*) 'c***************************************c'
      write (6,*) 'c***************************************c'
      write (6,*) 'Dynamic Pressure is climbing; dropping ET'
      write (6,*) 'c***************************************c'
      write (6,*) 'c***************************************c'
      write (6,*) ' '
      write (5,*) 's is climbing drop ET; inner loop'
      write (6,*) 'alt & Thrust is ',((x(1,nxt)*6378145d0)-
     1 6378145d0),Thrust
      aoa = .698131700796d0
      mass_flag = 2.d0
      x(7,nxt) =  0.174575376673d-19

      do 128 kki = 1,7
```

```
          x(kki,1) = x(kki,nxt)
  128 continue

      nxt = 0
      call haming(nxt)

      if (nxt .eq. 0) then
      write (6,*) 'nxt is zero 1229',nxt
      endif

      if(nxt .eq. 0) go to 140

c************c
c Helicopter c
c************c

      if (x(5,nxt) .lt. 4.36332312998d-3) then
c 13 deg 226.892802759d-3)then
c 15 deg 261.799387799d-3) then
c 14 deg 244.346095279d-3) then
c 0 deg      4.36332312998d-3) then
      gam_flag = 1.0d0
      endif

c  Integrating...
      do 130 lml = 1,nstp
         do 129 mlm = 1,nskp
            call haming(nxt)
  129 continue

      write (4,*) (t*806.8118744d0),((x(1,nxt)*6378145d0)-
     - 6378145d0), (x(5,nxt)/.0174532925199d0)

c*******************************************************************c
c Time, , alt, Pres lbf/ft^2(N to lbf                               c
c  then m^2 to ft^2); 1 atm = 2116.2166 lb/ft^2 = 101325 N/m^2      c
c  Various output files                                             c
c*******************************************************************c

      write (1,*) (t*806.8118744d0),((x(1,nxt)*6378145d0)-
     1 6378145d0),(s*30.0618114811D+9)

      gs = ((((x(4,nxt)**2)/1.09912835158d-3)*(1.d0-dcos(x(5,nxt))))/
     1 1.00085427871d0)
      write (2,*) (t*806.8118744d0),((x(1,nxt)*6378145d0)-
     1 6378145d0),gs

      if (x(1,nxt) .lt. 1.00391963494d0) then
      go to 140
      endif

      if (Thrust .lt. 0.d0) then
      go to 140
      endif

c***************************************************************c
```

```
c Savannah runway Lat = 32.1172 deg, Long = 278.800066667 deg c
c*********************************************************c
      lat_land = 0.56055088652d0
      long_land = 4.8659791181d0

c**********************************************************c
c Charleston S.C. runway Lat = 32.8916 N, Long = 279.967 deg E c
c**********************************************************c
c       lat_land = 0.574066716248d0
c       long_land = 4.88636281844d0

c************************c
c JAX, Jacksonville, FL. c
c************************c
c       lat_land = 0.531979264889d0
c       long_land = 4.85759746526d0

c******************c
c BGR, Bangor, ME. c
c******************c
c       lat_land = 0.78205382166d0
c       long_lang = 5.08213116464d0

      lat_shut = x(3,nxt)
      long_shut = x(2,nxt)

      dist_go = dacos(cos((pi/2.d0)-lat_land)*cos((pi/2.d0)-lat_shut)+
     1 sin((pi/2.d0)-lat_land)*sin((pi/2.d0)-lat_shut)*cos(long_shut -
     2 long_land))

      psi_cor = (-1.d0*(dasin((sin(pi/2.d0)-lat_land)*(sin(long_shut-
     1 long_land)/sin(dist_go)))))



      if (iidebug .eq. 4) then
      write (6,*) '*****************F********************************'
      write (6,*) 'wdotshut ',wdotshut,'for n_time ',n_time
      write (6,*) 'Shuttle dV for this run  = ',shut_dV
      write (6,*) 'Combined dV this run was = ',orig_dV
      write (6,*) ' '
      write (6,*) 'Distance to Runway in miles = ',((dist_go*6378145d0)/
     1 1609.344d0)
      write (6,*) 'Heading correction degs = ',(-1.d0*psi_cor/
     1 .0174532925199d0)
      write (6,*) ' '
      write (6,*) 'Thrust = ',Thrust
c       write (6,*) 'State Vector = '
c       write (6,*) (x(k,nxt),k=1,7)
      write (6,*) ' '
      write (6,*) 'Distance to Runway in miles = ',((dist_go*6378145d0)/
     1 1609.344d0)
      write (6,*) 'Heading correction (act-this)degs = ',(-1.d0*psi_cor/
     1 .0174532925199d0)
      if (wdot_flag .eq. 2.d0) then
      if ((s*30.0618114811D+9) .gt. 20.d0)then
```

```
      write (6,*) 'ET Sep s>20 begin pull up'
      gs = ((((x(4,nxt)**2)/1.09912835158d-3)*(1.d0-dcos(x(5,nxt))))/
     1 1.00085427871d0)
      write (6,*) 'g s at bottom of pull up = ',gs
      endif
      endif
      write (6,*) ' '
      write (6,*) 'r is    ',((x(1,nxt)*6378145d0)-6378145d0), ' meters'
      write (6,*) 'theta is',(360.d0 -(x(2,nxt)/.0174532925199d0)),
     1 ' Degrees West Longitude'
      write (6,*) 'phi is  ',(x(3,nxt)/.0174532925199d0), ' degrees Lat'
      write (6,*) 'v is    ',(x(4,nxt)*7905.36828d0), ' meters/sec'
      write (6,*) 'gamma is',(x(5,nxt)/.0174532925199d0), ' degrees'
      write (6,*) 'psi is  ',(x(6,nxt)/.0174532925199d0), ' degrees'
      write (6,*) 'aoa is ', aoa/.0174532925199d0
      write (6,*) 'beta is ', beta/.0174532925199d0
      write (6,*) 'm is    ',(x(7,nxt)*5.976D+24), ' kgs mass'
      write (6,*) ' '
      write (6,*) 'Seconds into flight ',int((t*806.8118744d0))
      write (6,*) 'Dyn Press psf is    ',int((s*30.0618114811D+9))
      write (6,*) ' '
      endif

c***********************c
c Modified Skip Reentry c
c***********************c

      if (thrst_flag .ne. 0.d0) then
      if (aoa .gt. 0.d0) then
      if (Cl .gt. 0.d0) then
      Kl = ((Cl*Sarea*rho)/(2.d0*m))
      B = ((Kl*Ho*dexp(-(x(1,nxt)-1.d0)/Ho)) - dcos(x(5,nxt)))
      Ht = (Ho*dlog((Kl*Ho)/(dcos(0.d0)+B)))
      if (Ht .gt. 0.d0) then
c      write (5,*) 'Ho is ',Ho*6378145d0
      write (5,*) ' '
      write (5,*) 'Inner loop info follows'
      write (5,*) '----------------------'
      write (5,*) 'Current alt        = ',((x(1,nxt)-1.d0)*6378145d0)
      write (5,*) 'HT bottom of pu is = ',Ht*6378145d0
      write (5,*) 'Kl 1317 is         = ',Kl
      write (5,*) 'B is               = ',B
      write (5,*) 'Cl is              = ',Cl
      write (5,*) 'aoa is    ',(aoa/.0174532925199d0)
      write (5,*) 'x5 gam is ',(x(5,nxt)/.0174532925199d0)
      write (5,*) 'v is      ',(x(4,nxt)*7905.36828d0), ' meters/sec'
      write (5,*) 'Dyn Press psf is    ',int((s*30.0618114811D+9))
      write (5,*) 'theta is',(360.d0 -(x(2,nxt)/.0174532925199d0)),
     1 ' Degrees West Longitude'
      write (5,*) 'phi is  ',(x(3,nxt)/.0174532925199d0), ' degrees Lat'
      write (5,*) 'Shuttle dV after LFBB Sep = ',shut_dv
      write (5,*) 'Combined dV this run was  = ',orig_dV
      write (5,*) 'wdot is      = ',wdot
      write (5,*) 'wdot flag    = ',wdot_flag
      write (5,*) 'dV is        = ',dV
      write (5,*) 'Timez is = ',n_time
```

```
      write (5,*) 'cl ',Cl
      write (5,*) 'Seconds into flight ',(t*806.8118744d0)
      write (5,*) ' '
      endif
      endif
      endif
      endif

c*******************c
c Conditional Output c
c*******************c

      if (Thrust .ge. 0.d0) then
      if (x(1,nxt) .gt. 1.00156785397635d0) then
       if (x(1,nxt) .lt. 1.00783926988176d0)then
c 25 km .00391963494d0) then
c 50 km 783926988176d0) then
c 29 km 454677653142d0) then
c 30 km 470356192906d0) then
c 27 km 415481303733d0) then
        if (x(5,nxt) .gt. -1.04719755119d0) then
c -50 deg 872664625995d0) then
      write (6,*) ' '
      write (6,*) '****************************************************'
      write (6,*) '*************** Sep LFBB *************************'
      write (6,*) '****************************************************'
      write (6,*) 'Shuttle dV after LFBB Sep = ',shut_dv
      write (6,*) 'Combined dV this run was = ',orig_dV
      write (6,*) 'Thrust        = ',Thrust
c        write (6,*) 'State Vector = '
c        write (6,*) (x(k,nxt),k=1,7)
      write (6,*) ' '
      write (6,*) 'Distance to Runway in miles = ',((dist_go*6378145d0)/
     1 1609.344d0)
      write (6,*) 'Heading correction (act-this)degs = ',(-1.d0*psi_cor/
     1 .0174532925199d0)
      if (wdot_flag .eq. 2.d0) then
      if ((s*30.0618114811D+9) .gt. 20.d0)then
      write (6,*) 'ET Sep s>20 begin pull up'
      gs = ((((x(4,nxt)**2)/1.09912835158d-3)*(1.d0-dcos(x(5,nxt))))/
     1 1.00085427871d0)
      write (6,*) 'g s at bottom of pull up = ',gs
      endif
      endif
      write (6,*) ' '
      write (6,*) 'r is    ',((x(1,nxt)*6378145d0)-6378145d0), ' meters'
      write (6,*) 'theta is',(360.d0 -(x(2,nxt)/.0174532925199d0)),
     1 ' Degrees West Longitude'
      write (6,*) 'phi is  ',(x(3,nxt)/.0174532925199d0), ' degrees Lat'
      write (6,*) 'v is    ',(x(4,nxt)*7905.36828d0), ' meters/sec'
      write (6,*) 'gamma is',(x(5,nxt)/.0174532925199d0), ' degrees'
      write (6,*) 'psi is  ',(x(6,nxt)/.0174532925199d0), ' degrees'
      write (6,*) 'aoa is ', aoa/.0174532925199d0
      write (6,*) 'beta is ', beta/.0174532925199d0
      write (6,*) 'm is    ',(x(7,nxt)*5.976D+24), ' kgs mass'
      write (6,*) 'Seconds into flight ',int((t*806.8118744d0))
```

```
      write (6,*) 'Dyn Press psf is    ',int((s*30.0618114811D+9))
      write (6,*) 'Throttle Down Time = ',n_time
      write (6,*) ':)> '
            endif
           endif
          endif
         endif

c 10km
      if (x(1,nxt) .lt. 1.00156785397635d0) then
      write (6,*) 'Attained 10 km!'
c 25 km
c      if (x(1,nxt) .lt. 1.0039664197d0) then
c      write (6,*) 'Attained 25 km!'
      go to 140
      endif

  130 continue
      endif
      endif

c***************************************************************c
c If LFBB sep occurs alone, then ET; the above loop re-        c
c initializes Haming and continues.  BELOW is continuation     c
c of LFBB separation loop no skip reentry till ET is Gone      c
c***************************************************************c

      if (x(1,nxt) .lt. 1.00391963494d0) then
      go to 140
      endif

      if (Thrust .lt. 0.d0) then
      go to 140
      endif

      lat_shut = x(3,nxt)
      long_shut = x(2,nxt)

c**************************************************************c
c Savannah runway Lat = 32.1172 deg, Long = 278.800066667 deg c
c**************************************************************c
      lat_land = 0.56055088652d0
      long_land = 4.8659791181d0

c***************************************************************c
c Charleston S.C. runway Lat = 32.8916 N, Long = 279.967 deg E c
c***************************************************************c
c      lat_land = 0.574066716248d0
c      long_land = 4.88636281844d0

c************************c
c JAX, Jacksonville, FL. c
c************************c
c      lat_land = 0.531979264889d0
c      long_land = 4.85759746526d0
```

```
c******************c
c BGR, Bangor, ME. c
c******************c
c      lat_land = 0.78205382166d0
c      long_lang = 5.08213116464d0

      dist_go = dacos(cos((pi/2.d0)-lat_land)*cos((pi/2.d0)-lat_shut)+
     1 sin((pi/2.d0)-lat_land)*sin((pi/2.d0)-lat_shut)*cos(long_shut -
     2 long_land))

      psi_cor = (-1.d0*(dasin((sin(pi/2.d0)-lat_land)*(sin(long_shut-
     1 long_land)/sin(dist_go)))))


      if (iidebug .eq. 4) then
      write (6,*) '*****************F********************************'
      write (6,*) 'wdotshut ',wdotshut,'for n_time ',n_time
      write (6,*) 'Shuttle dV for this run  = ',shut_dV
      write (6,*) 'Combined dV this run was = ',orig_dV
      write (6,*) 'LFBB Gone;  then to come ET1635 '
      write (6,*) ' '
      write (5,*) 'Kl2 is ',Kl,'B is ',B,'alt=',((x(1,nxt)-1.d0)*
     1 6378145d0)
      write (6,*) 'Distance to Runway in miles = ',((dist_go*6378145d0)/
     1 1609.344d0)
      write (6,*) 'Heading correction (act-this)degs = ',(-1.d0*psi_cor/
     1 .0174532925199d0)
      write (6,*) ' '
      write (6,*) 'Thrust = ',Thrust
c      write (6,*) 'State Vector = '
c      write (6,*) (x(k,nxt),k=1,7)
      write (6,*) ' '
      write (6,*) 'Distance to Runway in miles = ',((dist_go*6378145d0)/
     1 1609.344d0)
      write (6,*) 'Heading correction (act-this)degs = ',(-1.d0*psi_cor/
     1 .0174532925199d0)
      if (wdot_flag .eq. 2.d0) then
      if ((s*30.0618114811D+9) .gt. 20.d0)then
      write (6,*) 'ET Sep s>20 begin pull up'
      gs = ((((x(4,nxt)**2)/1.09912835158d-3)*(1.d0-dcos(x(5,nxt))))/
     1 1.00085427871d0)
      write (6,*) 'g s at bottom of pull up = ',gs
      endif
      endif
      write (6,*) ' '
      write (6,*) 'r is    ',((x(1,nxt)*6378145d0)-6378145d0), ' meters'
      write (6,*) 'theta is',(360.d0 -(x(2,nxt)/.0174532925199d0)),
     1 ' Degrees West Longitude'
      write (6,*) 'phi is  ',(x(3,nxt)/.0174532925199d0), ' degrees Lat'
      write (6,*) 'v is    ',(x(4,nxt)*7905.36828d0), ' meters/sec'
      write (6,*) 'gamma is',(x(5,nxt)/.0174532925199d0), ' degrees'
      write (6,*) 'psi is  ',(x(6,nxt)/.0174532925199d0), ' degrees'
      write (6,*) 'aoa is ', aoa/.0174532925199d0
      write (6,*) 'beta is ', beta/.0174532925199d0
      write (6,*) 'm is    ',(x(7,nxt)*5.976D+24), ' kgs mass'
      write (6,*) ' '
```

```
      write (6,*) 'Seconds into flight ',int((t*806.8118744d0))
      write (6,*) 'Dyn Press psf is    ',int((s*30.0618114811D+9))
      write (6,*) ' '
      endif
      if(x(7,nxt)+mdot*hh .lt. 0.224670919545d-19) then
       write (6,*) 'Warning! Out of fuel...wdot flag to 2'
       wdot_flag = 2.d0
       write (6,*) 'wdot flag = ',wdot_flag
       x(7,nxt) = 0.174575376673d-19
      endif

      if (thrst_flag .ne. 20.d0) then
      if (aoa .gt. 0.d0) then
      if (Cl .gt. 0.d0) then
      Kl = ((Cl*Sarea*rho)/(2.d0*m))
      B = ((Kl*Ho*dexp(-(x(1,nxt)-1.d0)/Ho)) - dcos(x(5,nxt)))
      Ht = (Ho*dlog((Kl*Ho)/(dcos(0.d0)+B)))
      if (Ht .gt. 0.d0) then
c      write (5,*) 'Ho is ',Ho*6378145d0
      write (5,*) ' '
      write (5,*) 'HT is =',Ht*6378145d0
      write (5,*) 'Kl 1317 is ',Kl
      write (5,*) 'B is ',B
      write (5,*) 'Cl is ',Cl
      write (5,*) 'aoa is ',(aoa/.0174532925199d0)
      write (5,*) 'x5 gam is ',(x(5,nxt)/.0174532925199d0)
      write (5,*) 'v is    ',(x(4,nxt)*7905.36828d0), ' meters/sec'
      write (5,*) 'Dyn Press psf is    ',int((s*30.0618114811D+9))
      write (6,*) 'theta is',(360.d0 -(x(2,nxt)/.0174532925199d0)),
     1 ' Degrees West Longitude'
      write (6,*) 'phi is  ',(x(3,nxt)/.0174532925199d0), ' degrees Lat'
      write (5,*) 'Shuttle dV after LFBB Sep = ',shut_dv
      write (5,*) 'Combined dV this run was = ',orig_dV
      write (5,*) 'wdot is      = ',wdot
      write (5,*) 'wdot flag    = ',wdot_flag
      write (5,*) 'dV is        = ',dV
      write (5,*) 'Timez is = ',n_time
      write (5,*) 'cl ',Cl
      write (5,*) 'alt=',((x(1,nxt)-1.d0)*6378145d0)
      write (5,*) 'Seconds into flight ',(t*806.8118744d0)
      write (5,*) ' '
      endif
      endif
      endif
      endif

      dist_go = dacos(cos((pi/2.d0)-lat_land)*cos((pi/2.d0)-lat_shut)+
     1 sin((pi/2.d0)-lat_land)*sin((pi/2.d0)-lat_shut)*cos(long_shut -
     2 long_land))

      psi_cor = (-1.d0*(dasin((sin(pi/2.d0)-lat_land)*(sin(long_shut-
     1 long_land)/sin(dist_go)))))

c********************c
c Conditional Output c
c********************c
```

```
      if (Thrust .ge. 0.d0) then
      if (x(1,nxt) .gt. 1.00391963494d0) then
       if (x(1,nxt) .lt. 1.00783926988176d0)then
c 50 km 783926988176d0) then
c 29 km 454677653142d0) then
c 30 km 470356192906d0) then
c 27 km 415481303733d0) then
        if (x(5,nxt) .gt. -1.04719755119d0) then
c -50 deg 872664625995d0) then
      write (6,*) ' '
      write (6,*) '*****************************************************'
      write (6,*) '*****************************************************'
      write (6,*) '*****************************************************'
      write (6,*) 'Shuttle dV after LFBB Sep = ',shut_dv
      write (6,*) 'Combined dV this run was = ',orig_dV
      write (6,*) 'wdot is       = ',wdot
      write (6,*) 'dV is         = ',dV
      write (6,*) 'Thrust        = ',Thrust
c        write (6,*) 'State Vector = '
c        write (6,*) (x(k,nxt),k=1,7)
      write (6,*) ' '
      write (6,*) 'Distance to Runway in miles = ',((dist_go*6378145d0)/
     1 1609.344d0)
      write (6,*) 'Heading correction (act-this)degs = ',(-1.d0*psi_cor/
     1 .0174532925199d0)
      if (wdot_flag .eq. 2.d0) then
      if ((s*30.0618114811D+9) .gt. 20.d0)then
      write (6,*) 'ET Sep s>20 begin pull up'
      gs = ((((x(4,nxt)**2)/1.09912835158d-3)*(1.d0-dcos(x(5,nxt)))))/
     1 1.00085427871d0)
      write (6,*) 'g s at bottom of pull up = ',gs
      endif
      endif

      write (6,*) ' '
      write (6,*) 'r is    ',((x(1,nxt)*6378145d0)-6378145d0), ' meters'
      write (6,*) 'theta is',(360.d0 -(x(2,nxt)/.0174532925199d0)),
     1 ' Degrees West Longitude'
      write (6,*) 'phi is  ',(x(3,nxt)/.0174532925199d0), ' degrees Lat'
      write (6,*) 'v is    ',(x(4,nxt)*7905.36828d0), ' meters/sec'
      write (6,*) 'gamma is',(x(5,nxt)/.0174532925199d0), ' degrees'
      write (6,*) 'psi is  ',(x(6,nxt)/.0174532925199d0), ' degrees'
      write (6,*) 'aoa is ', aoa/.0174532925199d0
      write (6,*) 'beta is ', beta/.0174532925199d0
      write (6,*) 'm is    ',(x(7,nxt)*5.976D+24), ' kgs mass'
      write (6,*) 'Seconds into flight ',int((t*806.8118744d0))
      write (6,*) 'Dyn Press psf is    ',int((s*30.0618114811D+9))
      write (6,*) 'Throttle Down Time = ',n_time
      write (6,*) ':) '
            endif
           endif
          endif
         endif

      if (x(1,nxt) .lt. 1.0039664197d0) then
```

```
      write (6,*) 'Attained 25 km!'

      go to 140
      endif

  132 continue
      endif
      endif
  135 continue
  140 continue
  150 continue
  160 continue
      write (6,*) 'Out at 160'
      stop
      end
c**********************************c

$INCLUDE: 'haming.for'
$INCLUDE: 'abo_rhs.for'
$INCLUDE: 'atm.for'
$INCLUDE: 'aero.for'
```

ABO_RHS.FOR

```
c Capt. Thomas Miller, 18 Feb 1999

      subroutine rhs(nxt)

      implicit double precision (a - h)
      implicit double precision (o - z)

      common /debug/ idebug,ig,iidebug

      common /flags/ mass_flag,thrst_flag,d_psi,d_gam,massref,wdotref
      double precision mass_flag,thrst_flag,d_psi,d_gam,massref,wdotref

      common /flags2/ thrstref,sref,timeref,aoaref,betaref,dv_max2
      double precision thrstref,sref,timeref,aoaref,betaref,dv_max2

      common /flags3/ psi_flag,spin_flag,spin,betapsi,dif,pi
      double precision psi_flag,spin_flag,spin,betapsi,dif,pi

      common /Thrust/ Thrust,wdot_flag,wdot,minT,maxT,num,orig_dV,dgam
      double precision Thrust,Isp,msq,mcu,num,orig_dV,dgam
      double precision wdot_flag,wdot,minT,maxT,numstep

      common /ham/ t,x(7,4),f(7,4),err(7),n,h,hh,mode,loop(7,1)
      double precision t,x,f,err,hh,h,loop

      common /ctrl/ aoa,beta,mdot,dV,wdotLFBB,wdotshut,numstep,n_stp
      double precision aoa,beta,mdot,dV,wdotLFBB,wdotshut,dpsi

      common /amat/ a(15,15),hamil,ithrot,igt,omega,gam_flag
      double precision a,hamil,omega,n_stp,gam_flag

      common /maxq/ s,wdot_lf,wdot_sh,shut_dV,dv_min,dv_max,nnn_stp
      double precision s,shut_dV,dv_min,dv_max,dv_stp

      double precision P0,ALT,DALT,TALT,dDdr,d2Ddr,sonic,dmfpdr
      double precision mfp,PALT

      double precision Kn, Cd, Cdp, Clp

      common /pullup/ Ht,Kl,Ho,B,rho,Sarea,Cl,m
      double precision Ht,Kl,Ho,B,rho,Sarea,Cl,m

      data istart /0/

c***************************c
c Extract state vector      c
c***************************c

      r = x(1,nxt)
      theta = x(2,nxt)
      phi = x(3,nxt)
      V = x(4,nxt)
      gamma = x(5,nxt)
```

```
      psi = x(6,nxt)
      m = x(7,nxt)
      Ho = (7010.4d0/6378145.d0)


c*******************************************c
c Calculate common auxillary quantities      c
c*******************************************c

      cosgam = dcos(gamma)
      singam = dsin(gamma)
      secgam = 1.d0/ dcos(gamma)
      tangam = dtan(gamma)
      cosphi = dcos(phi)
      sinphi = dsin(phi)
      secphi = 1.d0/ dcos(phi)
      tanphi = dtan(phi)
      cospsi = dcos(psi)
      sinpsi = dsin(psi)
      Vsq = V*V
      msq = m*m
      rsq = r*r
      sphisq = secphi*secphi
      tphisq = tanphi*tanphi
      sgamsq = secgam*secgam
      tgamsq = tangam*tangam
      sphicu = secphi*secphi*secphi
      sgamcu = secgam*secgam*secgam
      Vcu = V*V*V
      rcu = r*r*r
      mcu = m*m*m

c**********************************************c
c Calculate aerodynamic vals P0 in N/m^2 is not  c
c  P0 = 101325.d0 this is sea level.  For this   c
c  model abort occurs L+9 secs above this        c
c**********************************************c

      P0 = 99621.5573252d0

      ALT = (r - 1.d0)* 6378145d0

      call ATM(ALT,P0,PALT,TALT,DALT,dDdr,d2Ddr,sonic,mfp,dmfpdr)

c***************************c
c Convert units on rho etc  c
c***************************c

      rho = DALT*((6378145d0**3.d0)/5.976d24)
      drhodr = dDdr*((6378145d0**4.d0)/5.976d24)
      d2rhodr2 = d2Ddr*((6378145d0**5.d0)/5.976d24)
      Kn = mfp/(12.058d0/1.89051832469d-6)

      PALT = PALT*(6378145d0*806.8118744d0**2.d0)/(5.976d24)

      g = (9.80665d0*806.8118744d0**2.d0)/(r*6378145d0)
```

```
c************************************c
c Calculate Lift force per unit mass  c
c************************************c

      call AERO(aoa,Kn,Cd,Cl,Cdp,Clp)

c*****************************************************c
c Calculate Drag acceleration (/m)                  c
c  drag=(.5*CdArhoV^2)/m, s=.5rhoV^2; A=surface area, c
c  Sarea area 2690 ft^2 or 249.9091776 m^2           c
c  lift=(.5Cl*Surface Area*rho*V^2)/m                c
c*****************************************************c

      s = .5d0*rho*Vsq
      Sarea = (249.9091776d0/6378145d0**2.d0)

      omega = 7.292116d-05*806.8118744d0

      if (wdot_flag .eq. 1.d0) then
       dV = shut_dV
      endif

c
c 1 deg / tu = 1deg->rad*806.8118744
c
      dgam = (d_gam * (-14.0815236524d0))
      dpsi = d_psi
c      write (6,*) 'dgam prior to check for neg angle',dgam

c*********************************************c
c 45,20,10 and 30 degrees for dgam in that order,      c
c  with .75 vs .5 gam 31 Jan, this reverses gam c
c  as the hill is crested, a pull up....  c
c*********************************************c

c Helicopter gam = 0

      if (gam_flag .eq. 1.0d0) then
      dgam = 0.d0
      endif


c beta
      if (thrst_flag .eq. 1.d0) then
      cosbeta = ((m*g*singam)+(Cd*Sarea*s)+(m*dV))

      sinbeta = (((m*V*2.d0*omega*r*cosphi*cosphi*cospsi*singam)-
     1 (m*V*2.d0*omega*r*cosphi*sinphi*cosgam)-(m*Vsq*cosgam*cosgam*
     2 sinpsi*sinphi)+(m*V*cosgam*dpsi*r*cosphi))/(r*cosphi))

      beta = datan2(sinbeta,cosbeta)
      sinbeta = dsin(beta)
      cosbeta = dcos(beta)
      endif
```

```
c aoa
      if (thrst_flag .eq. 1.d0) then
      cosaoa = (((m*g*singam)+(Cd*Sarea*s)+(m*dV))/cosbeta)

      sinaoa = (((m*g*cosgam*r)-(m*Vsq*cosgam)+(m*2.d0*omega*V*cosphi*
     1 sinpsi*r)+(m*dgam*V*r)+Cl*Sarea*s*r)/r)

      aoa = datan2(sinaoa,cosaoa)

      sinaoa = dsin(aoa)
      cosaoa = dcos(aoa)
      endif

c Thrust

      if (thrst_flag .eq. 1.d0) then
      Thrust = dsqrt((((g*m*singam)+(Cd*Sarea*s)+(m*dV))**2.d0)+
     1 (((g*m*cosgam)+(m*2.d0*omega*V*cosphi*sinpsi)-((m*Vsq*cosgam)/r)+
     2 (m*dgam*V)+(Cl*Sarea*s))**2.d0)+(((2.d0*omega*V*m*cosphi*cospsi*
     3 singam)-(2.d0*omega*V*m*sinphi*cosgam)-((Vsq*m*cosgam*cosgam*
     4 sinpsi*tanphi)/r)+(m*cosgam*dpsi*V))**2.d0))
      endif

c*****************************************************************c
c This Thrust equation is for climbing out of                     c
c  atmosphere alpha and beta are both zero, no aoa or yaw.        c
c*****************************************************************c

      if (thrst_flag .eq. 0.d0) then
      Thrust = ((m*g*singam)+(Cd*Sarea*s)+(m*dV))-
     1 (PALT*(43.45410664d0/(6378145d0**2.d0)))
      endif

      if (thrst_flag .eq. 0.d0) then
       if (Thrust .gt. 6.82531786504d-19) then
      Thrust = 6.82531786504d-19
       endif
      endif

      wdot = wdotLFBB

      if(wdot_flag .eq. 1d0) then
       wdot = wdotshut
      endif

c**************************************************************c
c  Check staging thrust of SSME's < 0.747943510197d-19    c
c**************************************************************c

      if(wdot_flag .eq. 1d0) then
      if(Thrust .gt. 0.747943510197d-19) then
       Thrust = 0.747943510197d-19
      endif
      endif

      Isp = (Thrust/wdot)*806.8118744d0
```

```
      Isp = Isp/806.8118744d0

      gsea = 1.d0

      mdot = -(Thrust/(gsea*Isp))

c****************************************c
c  Out of fuel time to Glide...hopefully  c
c****************************************c

      if(wdot_flag .eq. 2.d0) then
       Thrust = 0.d0
       wdot = 0.d0
       Isp = 0.d0
       mdot = 0.d0

c aoa 41 deg
c      aoa = 0.715584993316d0
c aoa 40 deg
c      aoa = 0.698131700796d0
c aoa 14 deg
c      aoa = 0.244346095279d0
c aoa 35 deg
c      aoa = 0.610865238197d0
c aoa 30 deg
c      aoa = 0.523598775597d0
c aoa 36 deg
c     aoa = 0.628318530716d0
c aoa 42 deg
      aoa = 0.733038285836d0
c aoa 33 deg
c     aoa = 0.575958653157d0
c aoa 31 deg
c     aoa = 0.541052068117d0
c aoa 32 deg
c     aoa = 0.558505360637d0
c aoa 31.5 deg
c     aoa = 0.549778714377d0
c aoa 31.25 deg
c     aoa = 0.545415391247d0
c aoa 31.05 = 0.541924732743d0
c aoa 31.1 = 0.542797397369d0
c aoa 31.15 = 0.543670061995d0
c aoa 31.2 = 0.544542726621d0
c aoa 31.17 = 0.544019127845d0
c aoa 31.18 = 0.54419366077d0

c 34km
       if (x(1,nxt) .lt. 1.0053307035196)then
       aoa = 0.575958653157d0
       endif

       beta = 0.d0
      endif

      sinbeta = dsin(beta)
```

```
      cosbeta = dcos(beta)
      sinaoa = dsin(aoa)
      cosaoa = dcos(aoa)

c****************************************c
c Calculate the equations of motion      c
c****************************************c

      f(1,nxt) = V*singam
c      write (*,*) 'r dot in m/s is ', ((f(1,nxt)*6378145)/806.8118744d0)

      f(2,nxt) = V*cosgam*secphi*sinpsi/r
c      write (*,*) 'Theta dot in deg/s is ', (f(2,nxt)/14.0815236524d0)

      f(3,nxt) = V*cosgam*cospsi/r
c      write (*,*) 'Phi dot in deg/s is ', (f(3,nxt)/14.0815236524d0)

      f(4,nxt) = Thrust*cosaoa*cosbeta/m - g*singam - (Cd*Sarea*s/m)
c      write (*,*) 'V dot in m/ss is ', (f(4,nxt)*9.79827953836d0)

      if (thrst_flag .eq. 0.d0) then
      f(5,nxt) = (-(g*cosgam) + Vsq*cosgam/r +
     1 Thrust*sinaoa/m - 2.d0*omega*V*cosphi*sinpsi)/V
c     write (6,*) 'cd'
c      write (*,*) 'gam dot in deg/s is ', (f(5,nxt)/14.0815236524d0)
      else
      f(5,nxt) = (-(g*cosgam) + Vsq*cosgam/r +
     1 Thrust*sinaoa/m - 2.d0*omega*V*cosphi*sinpsi + (Cl*Sarea*s/m))/V
c     write (6,*) 'cl'
      endif

      f(6,nxt) = (Thrust*cosaoa*sinbeta/(m*cosgam) -
     1 2.d0*omega*V*(-sinphi + cosphi*cospsi*tangam) +
     2 Vsq*cosgam*sinpsi*tanphi/r)/V
c      write (*,*) 'psi dot in deg/s is ', (f(6,nxt)/14.0815236524d0)

      f(7,nxt) = mdot
c      write (*,*) 'Delta mdot in kg/s is ', (f(7,nxt)*9.18049346969d+18)

      return
      end
```

SUBROUTINE ATM.FOR

```
c
      SUBROUTINE ATM(ALT,P0,PALT,TALT,DALT,dDdr,d2Ddr,sonic,mfp,dmfpdr)
c
c     Earth atmosphere program, Regan and Anandarskarian, AIAA
c     "Dynamics of Atmospheric Re-entry", appendix A
c
c     input:      alt       altitude in meters
c                 po        ground level pressure, n/sq m
c     output:     palt      pressure at altitude, n/sq m
c                 talt      temperature at altutude, deg C
c                 dalt      density at altitude, kg/cu m
c                 dDdr      density gradient, kg/m^4
c                 d2Ddr     density second gradient, kg/m^5
c                 sonic     speed of sound, m/s
c                 mfp       mean free path, m
c                 dmfpdr    mean free path derivative, dimensionless
c
      DOUBLE PRECISION Z(21),TM(21),LR(21),B,G0,R,D(21),P(21),P0,D0,RR
      DOUBLE PRECISION  TALT,PALT,DALT,alt,m(21),m0,dd(21),ti
      DOUBLE PRECISION  e1,e2,e3,e4,e5,RE,GALT,sonic,sigma,mfp,s,nu,N
      double precision dDdr,dE1dr,dE2dr,dE3dr,d2Ddr,dmfpdr
      double precision dmoldr,dnudr
C
c     data stmts for break altitudes, temperatures, and molecular wts
c
c     altitudes
      data (z(i),i=1,21)/ 0.d3,  11.0191d3,  20.0631d3,  32.1619d3,
     1               47.3501d3,  51.4125d3,  71.8020d3,  86.00d3,
     2              100.d3,      110.d3,      120.d3,      150.d3,
     3              160.d3,      170.d3,      190.d3,      230.d3,
     4              300.d3,      400.d3,      500.d3,      600.d3,
     5              700.d3 /
c     molecular temperature
      data (TM(i),i=1,21)/ 300.d0,  216.65d0,  216.65d0,  228.65d0,
     1                   270.65d0,  270.65d0,  214.65d0,  186.946d0,
     2                   210.65d0,  260.65d0,  360.65d0,  960.65d0,
     3                  1110.60d0, 1210.65d0, 1350.65d0, 1550.65d0,
     4                  1830.65d0, 2160.65d0, 2420.65d0, 2590.65d0,
     5                  2700.0d0 /
c     molecular wts
      data (m(i),i=1,21)/ 28.9664d0,  28.964d0,  28.964d0,  28.964d0,
     1                    28.964d0,   28.964d0,  28.962d0,  28.962d0,
     2                    28.880d0,   28.560d0,  28.070d0,  26.920d0,
     3                    26.660d0,   26.500d0,  25.850d0,  24.690d0,
     4                    22.660d0,   19.940d0,  17.940d0,  16.840d0,
     5                    16.170d0 /
c     first pass flag
      data ifirst / 0 /
c
C     define constants on first pass
C
```

```
      if( ifirst .ne. 0 ) go to 1000
c
      B=3.139D-7
c     acceleration of gravity
      G0=9.7803D0
c     universal gas const, J/kg
      RR=8313.432D0
      r=RR/m(1)
c     planetary radius, meters
      RE = 6378145d0
c     avagadro's number
      N = 6.0221d+26
C
C     initialize lapse rate for altitude regions
C
      DO 10, L=1,21
           lr(l) = ( TM(L+1) - TM(L) )/( z(L+1) - z(L) )
10    continue
69    close(7)
      D0=P0/(R*TM(1))
      P(1)=P0
      D(1)=D0
      do 20,l=1,20
            r=rr/m(l)
            IF (LR(L).EQ.0.D0) THEN
                     E1=1.D0-(B/2.D0)*(Z(L+1)-Z(L))
                     E2=G0*(Z(L+1)-Z(L))/(R*TM(L))
                     P(L+1)=P(L)*DEXP(-E1*E2)
                     D(L+1)=D(L)*DEXP(-E1*E2)
                     dd(l)=d(l+1)-d(l)/(z(l+1)-z(l))

            ELSE
                     E1=1.d0+(LR(L)/TM(L))*(Z(L+1)-Z(L))
                     E2=G0*B/(r*LR(L))
                     E3=E2*(Z(L+1)-Z(L))
                     E4=E2/B*(B/E2+1.d0+B*((TM(L)/LR(L))-Z(L)))
                     E5=E2/B*(1.d0+ B*((TM(L)/LR(L))-Z(L)))
                     P(L+1)=P(L)*(E1**(-E5))*DEXP(E3)
                     D(L+1)=D(L)*(E1**(-E4))*DEXP(E3)
                     dd(l)=(d(l+1)-d(l))/(z(l+1)-z(l))
            ENDIF
20    continue
c
      ifirst = 1
c
 1000 continue
c     write (*,*) 'atm: arrays stored'
C
C     determine which region altitude falls into
C
      do 500 j = 1,20
           if( alt .lt. z(j+1) ) then
                I = j
                go to 501
           endif
  500 continue
```

```
      I = 20
  501 continue
c     write (*,*) 'atm: altitude band',I
C
C     determine parameters at altitude
c
c     write(*,*)'I',i
      TALT=TM(I)+LR(I)*(ALT-Z(I))
      GALT=G0*(RE**2/((RE+ALT)**2))
      r=rr/m(i)
c
c     If lapse rate is zero
c
      IF (LR(I).EQ.0.D0) THEN
               E1=1.D0-(B/2.D0)*(ALT-Z(I))
               E2=G0*(ALT-Z(I))/(R*TM(I))
               PALT=P(I)*DEXP(-1.d0*E1*E2)
               DALT=D(I)*DEXP(-1.d0*E1*E2)
               dE1dr = -B/2.d0
               dE2dr = G0/(R*TM(I))
               dDdr = -D(I)*dexp(-E1*E2)*( E1*dE2dr + dE1dr*E2)
               d2Ddr = -dDdr*(E1*dE2dr + dE1dr*E2) - 2.d0*D(I)*
     1            dexp(-E1*E2)*dE1dr*dE2dr
c
c     If Lapse Rate not equal to zero
c
      ELSE
               E1=1.D0+(LR(I)/(TM(I)))*(ALT-Z(I))
               E2=G0*B/(r*LR(I))
               E3=E2*(ALT-Z(I))
               E4=E2/B*(B/E2+1.d0+B*((TM(I)/LR(I))-Z(I)))
               E5=E2/B*(1.d0+ B*((TM(I)/LR(I))-Z(I)))
               PALT=P(I)*(E1**(-E5))*DEXP(E3)
               DALT=D(I)*(E1**(-E4))*DEXP(E3)
               dE1dr = LR(I)/TM(I)
               dE3dr = E2
               dDdr = D(I)*( -E4*(E1**(-E4-1.d0))*dE1dr
     1            + (E1**(-E4))*dE3dr )*dexp( E3 )
               d2Ddr = D(I)*( E4*(E4+1.d0)*(E1**(-E4-2.d0))*
     1            dE1dr*dE1dr - E4*(E1**(-E4-1.d0))*dE1dr*dE3dr )*
     2            dexp(E3) + dDdr*dE3dr
      ENDIF
c     write (*,*) 'atm: density',DALT
c
c     speed of sound
c
      sonic = dsqrt( 1.4d0 * r * TM(I) )
c
c     molecular size, meters
      sigma = 3.65d-10
c
c     molecular wt at altitude, kg/mole
      mol = m(I) + (m(I+1)-m(I))*(ALT-Z(I))/(Z(I+1)-Z(I))
c     write (*,*) 'atm: mol',mol
c
c     number density at altitude, number / meter cubed
```

```
      nu = DALT * N / mol
c     write (*,*) 'atm: nu',nu
c
c     mean free path
c
      mfp = 1.d0/( dsqrt(2.d0) * 3.1415926d0 * sigma * sigma * nu )
c     write (*,*) 'atm: mfp',mfp
c
c     mean free path derivative
c
      dmoldr = ( m(I+1)-m(I))/(Z(I+1)-Z(I))
      dnudr = dDdr*N/mol - DALT*N*dmoldr/( mol*mol )
      dmfpdr = - dnudr / ( dsqrt(2.d0)*3.1415926d0 *sigma*sigma*nu*nu )
c     write (*,*) 'atm: dmfpdr',dmfpdr
c
      RETURN
      END
```

INPUT FILE: ABORT.IN

```
1.000027603394376         4.876375247312332   4.993081693153024E-001
4.521228749522317E-003    1.566322763194498   8.325507229168753E-001
   3.413484164565641E-019
   1.240429996817460E-003   7.43667785562000E-001
      299.000000        50.000000

      0.d0      1.0244d0      16
      0.1d0     1.5d0         16
      4
c Above is abort 1 SSME failure at L+10 seconds

        1.005668904744045         4.879500322815867   5.024450999202795E-001
   1.501080746434611E-001   6.442927216051213E-001   7.180002122034157E-001
   1.683184621520560E-019
   1.364472996499195E-001   7.43667785562000E-001
      245.000000        50.000000

      0.d0      1.0244d0      16
      0.1d0     1.5d0         16
      3

c Above is abort 1 SSME failure at L+110 seconds

c psi = 48.9763440371 = .854798459035c
c gam = 89.8133270522 = 1.5675382692300

c  r, theta, phi, V, gamma, psi, mass, t0 & tf, nstp, nskp.
c  Dimensionless, see Bate, Mueller, and White Appendix A
c  Desired r=48825.912, v=1727.667

c************************************************************************
***
    1.0000223180900    4.8763752299700    0.4993081557320
4.00171110043D-03
    1.5675374480100    .854798459035      3.429358472300D-19
0.0D0 1.5629417959900D-1
126 30

INITIAL CONDITIONS NOMINAL MODEL PAD A
c gam = 89.81328 = 1.5675374480100
c psi = 48.9763440371 = .854798459035c

Thrust =   4.886402159338922E-019
Initial State Vector Shuttle/LFBB
r is         142.348014143586600 meters
theta is     279.395722163719000 degrees
phi is        28.608250221514440 degrees
v is          31.634999535114840 meters/sec
gamma is      89.813280695407310 degrees
psi is        48.976344416380610 degrees
m is     2049384.623046480000000 kgs mass
```

```
FINAL CONDITIONS
Thrust =    5.679176775602426E-019
State Vector =
      1.007655076737892        4.881694186245593   5.046423886511117E-001
  2.099236787992863E-001   5.538951019324256E-001   7.191380183123132E-001
  1.429195697770823E-019

r is        48825.189420403640000 meters
theta is      279.700473746117200 degrees
phi is         28.913879033181010 degrees
v is         1659.523991600786000 meters/sec
gamma is       31.735851633775240 degrees
psi is         41.203573336799470 degrees
m is       854087.348987843800000 kgs mass
Inclination is       51.604145012591810 degrees
Current time=      126.100000000078300
Current Dyn Press in psf=       35.045760082614790
c***************************************************************************
***
```

Bibliography

1. Bate, Roger R., Mueller, Donald D., and White, Jerry E. Fundamentals of Astrodynamics. New York: Dover Publications, 1971.
2. Bentley, Dennis, Shuttle Ascent Specialist, NASA JSC, Houston TX. Telephone interview 22 July 1998.
3. -----. NASA JSC, Houston TX. Telephone interview 16 February 1999.
4. Beyer, William H. CRC Standard Mathematical Tables and Formulae 29th Ed. Boca Raton: CRC Press, 1991.
5. Blanchard, Robert C., Larman, Kevin T., and Moats, Christina D. "Rarefied-Flow Shuttle Aerodynamics Flight Model," Journal of Spacecraft and Rockets, 31: 550-556 (August 1994)
6. Boas, Mary L. Mathematical Methods in the Physical Sciences 2nd Ed. New York: John Wiley & Sons, 1966.
7. Cox, Richard D., Private Pilot, Single Engine Land, Centerville OH. Telephone interview. 8 January 1999.
8. Carman, Gilber, Shuttle Descent Specialist, NASA, Johnson Spaceflight Center, Houston TX. Telephone interview. 22 July 1998, 12 February 1999.
9. Fraley, Charles, Shuttle RTLS/ECAL Specialist, NASA, Downey CA. Telephone interview. 12 February 1999.
10. Greene, Betty J., Aerospace Engineer, Ballistic and Space Design Branch, National Air Intelligence Center, Wright-Patterson OH. Personal interview. 23 July 1998.
11. Hage, R.T., Simulation Model for Probabilistic Analysis of Space Shuttle Abort Modes NASA Technical Memorandum TM-108432. G3/16 0198109. Huntsville AL: George C. Marshall Space Flight Center, November 1993 (N94-21859).
12. Healy, Thomas J., Chief Engineer, Liquid Fly Back Booster, Boeing Defense Space Group, Downey CA. Telephone interview. 9 July 1998.
13. Hines, Mark R., "Baseline LFBB w/RTLS Eliminated Performance Baseline and Trajectory Run." Electronic Message. 153900Z, 5 November 1998.
14. Hines, Mark R., Trajectory Specialist, RSS Ascent Performance, Boeing Defense Space Group, Downey CA. Telephone interview. 9 November 1998.
15. Hines, Mark R., "Thrust Calculations." Electronic Message with Excel Spreadsheet. 234600Z, 18 November 1998.
16. Humble, Ronald W., Henry, Gary N. and Larson, Wiley J. Space Propulsion Analysis and Design. New York: The McGraw-Hill Companies, 1995.
17. Isakowitz, Steven J. International Reference Guide to Space Launch Systems. Washington DC: American Institute of Aeronautics and Astronautics, 1995.
18. Koffman, Elliot B. and Friedman, Frank L. FORTRAN with Engineering Applications. Massachusetts: Addison-Wesley Publishing Company, 1993.
19. Kramer, Stuart C., Vector Dynamics Air Force Institute of Technology: Private Printing, 1997.
20. Larson, Wiley J. and Wertz, James R. Space Mission Analysis and Design 2nd Ed. Torrance CA: Microcosm, Inc., 1992.

21. Lipschutz, Seymour and Poe, Arthur. Schaum's Outline Series-Programming with FORTRAN. New York: McGraw-Hill, 1978.
22. Little, J.K. Class handout, MENG 531, Nonchemical Propulsion. School of Engineering, AFIT, Wright-Patterson AFB OH, March 1998.
23. Marshall Space Flight Center. MASTRE Trajectory Code Update to Automate Flight Trajectory Design, Performance, Preditctions, and Vehicle Sizing for Support of Shuttle and Derived Vehicles – Programmers Manual: Final Report, April 1993. Contract NASA CR-192558. Library of Michigan, Lansing MI, 1993 (N9328321).
24. NASA Facts Online. "International Space Station Handbook" Public Web Site, n. pag. http://station.nasa.gov/station/reference/factbook/presbudget.html. Feb 14 1999.
25. NASA Facts Online. "Space Shuttle TAL Sites" KSC Release No. 64-88, n. pag. http://www-pao.ksc.nasa.gov/kscpao/nasafact/tal.htm#tal. Jan 12 1999.
26. NASA Facts Online. "Launch Complex 39, Pads A and B" KSC Release No. 252-85, n. pag. http://www-pao.ksc.nasa.gov/kscpao/nasafact/pads.htm. Oct 10 1998.
27. NASA Space Link. "Flight.063.STS-65 Keplerian.Elements" Education Div., n. pag. http://spacelink.msfc.nasa.gov/NASA.Projects/Human.Exploration.and.Development.of.Space/Human.Space.Flight/Shuttle/Shuttle.Missions/Flight.063.STS-65/Keplerian.Elements. Dec 09 1998.
28. NASA News Ref. "Space Transportation System Handbook" Public Relations Doc., n. pag. http://www.ksc.nasa.gov/shutlle/technology/sts-newsref/stws_overview.html. Dec 10 1998.
29. Regan, Frank J. and Anandakrishnan, Satya M. Dynamics of Atmospheric Reentry. Washington DC: American Institute of Aeronautics and Astronautics, 1993.
30. Spenny, Curtis H. Class lecture, SENG 531, "Loss of Space Corporate Knowledge." School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB OH, Summer Quarter 1998.
31. Smolarski, Dennis C. Essentials of FORTRAN. Piscataway, New Jersey: Research & Education Association , 1989.
32. Victer, Ira F., Program Director, Liquid Fly Back Booster, Boeing Defense Space Group, Downey CA. Telephone interview. 11 December 1997.
33. Victer, Ira F., "Liquid Fly Back Booster (LFBB) Program Overview." Company Report, Boeing Defense Group, Downey CA. February 1998.
34. Vinh, Nguyen X., Busemann, Adolf, and Culp, Robert D. Hypersonic and Planetary Entry Flight Mechanics. Ann Arbor: The University of Michigan Press, 1980.
35. Welch, Michael, Aerospace Engineer, Ballistic and Space Design Branch, National Air Intelligence Center, Wright-Patterson OH. Personal interview. 23 July 1998.
36. Weinman, David G. FORTRAN for Scientist and Engineers. Boston: PWS-KENT Publishing Company, 1989.
37. Whitehead, Douglas S., Chief Engineer, Space Shuttle Systems Integration, NASA, JSC Houston TX. Telephone interview. 17 July 1998.
38. -----. NASA JSC Houston TX. Telephone interview. 12 February 1999.
39. Wiesel, William E., Jr. Spaceflight Dynamics, 2nd Ed. New York: The McGraw-Hill Companies, 1997.
40. Wiesel, William E., Jr. Class handout distributed in MECH 636, Advanced Astrodynamics. School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB OH, September 1998.

Vita

Captain Thomas L. Miller Jr. was born on 28 February 1967 in Cincinnati, Ohio. He graduated from La Salle H.S. in May of 1985. After a day in the Navy and 4 months in the Coast Guard, he joined the Army and attended Army Basic Training at Fort Knox Kentucky in February 1986. He spent two years in the 2/2 Armored Calvary Regiment guarding the border between East and West Germany. He was then stationed for two years at Ft. Stewart Georgia as part of 3/69 Armor's Rapid Deployment for the Middle East. In 1990 he left the Army to attend college. He joined the Ohio National Guard and mustered with the 512th Engineer Battalion for two years. While receiving his Bachelors in Electrical Engineering Technology from the University of Cincinnati, he received a full Air Force ROTC scholarship for his 4.0 GPA. He was commissioned on 11 June 1994. After marrying his sweetheart from survival on the 4th of July, he was assigned to the 45th Space Wing, Patrick AFB Florida on 30 September 1994. His duty AFSC was 13S3B, Space-Lift Operations. At Cape Canaveral he was qualified as a Range Control Officer launching Shuttle and Titan rockets. After winning Guardian Challenge and pinning on 1st Lieutenant in 1996, he was selected to be the 45th Space Wing's Chief of Range Operational Training. In July 1997, he was selected to attend the Air Force Institute of Technology's School of Engineering and receive a Masters of Science Degree in Space Operations. His follow-on assignment is to the Space and Missile Center at LA AFB, California. Captain Miller is married to the former Pattie R. Lashley of Tallahassee Florida.

Permanent Address: 28 Brightwood Drive
Cold Springs KY 41076
millers7@hotmail.com