ABSTRACT

Title: SURVIVABILITY OF EMERGENCY ESCAPE FROM A SIMULATED SPACE SHUTTLE ENTRY TRAJECTORY

Elisa Gail Shapiro Master of Science 2005

Directed By: Associate Professor David L. Akin, Aerosospace Engineering
Space Systems Laboratory
University of Maryland

As a component of the NASA SVTI on reusable launch vehicles, the University of Maryland Space Systems Laboratory in 2002 began an investigation of the feasibility of providing crew escape options from the Space Shuttle or a future reusable launch vehicle throughout the entry profile. MATLAB™ code was written to simulate a typical winged vehicle entry trajectory. The code was extended to simulate ejection trajectories for a ballistic capsule on a minute-by-minute basis throughout the original entry profile. The ejection trajectories, representing instantaneous change in vehicle aerodynamic properties, were studied for theoretical survivability and were analyzed to determine the optimum design characteristics for an escape vehicle. The results indicate that escaping from a reentering vehicle is survivable for most of the

trajectory, with the exception of a small period right around maximum aerodynamic pressure.

SURVIVABILITY OF EMERGENCY ESCAPE FROM A SIMULATED SPACE SHUTTLE ENTRY TRAJECTORY.

By

Elisa Gail Shapiro

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Master of Science
2005

Advisory Committee:
Professor David L. Akin, Chair
Professor Mary Bowden
Professor Norman Wereley

Preface

There is an art, it says, or rather, a knack to flying.

The knack lies in learning how to throw yourself at the ground and miss.

Pick a nice day, it suggests, and try it.

The first part is easy.

All it requires is simply the ability to throw yourself forward with all your weight, and willingness not to mind that it's going to hurt.

That is, it's going to hurt if you fail to miss the ground.

Most people fail to miss the ground, and if they are really trying properly, the likelihood is that they will fail to miss it fairly hard.

Clearly, it is this second part, the missing, which presents the difficulties.

- Douglas Adams

  from *The Hitchhiker's Guide to the Galaxy*

Dedication

This thesis is dedicated to the men and women who have died in the course of space exploration. The crews of Soyuz 1, Soyuz 11, STS 51-L Challenger and STS 107 Columbia.

Vladmir Komarov

Georgi Dobrovolsky

Vladislav Volkov

Victor Patsayev

Francis R. Scobee

Michael J. Smith

Judith A. Resnik

Ellison S. Onizuka

Ronald E. McNair

Gregory B. Jarvis

Sharon Christa McAuliffe

Rick Husband

William McCool

Mike Anderson

Kalpana Chawla

Dave Brown

Laurel Clark

Ilan Ramon

Acknowledgements

I would like to thank Dr. David Akin for finding the funding to hire me, and assisting me with this thesis, even if it did take four tries to find a topic that would work. I apologize for my inability to tell a 'than' from a 'then' in my draft of this paper. I would like to thank my parents for lending me money for car repairs when I was broke, letting me mooch dinner when I was hungry, and letting me call up and scream when I needed to.

This work has been sponsored by the Space Vehicle Technology Institute, under grant NCC3-989, one of the NASA University Institutes, with joint sponsorship from the Department of Defense. Appreciation is expressed to Claudia Meyer, Mark Klemm and Harry Cikanek of the NASA Glenn Research Center, and to Dr. John Schmisseur and Dr. Walter Jones of the Air Force Office of Scientific Research.

Table of Contents

This Table of Contents is automatically generated by MS Word, linked to the Heading formats used within the Chapter text.

List of Tables

List of Figures

Chapter 1: Introduction

1.1 The Problem of Crew Safety

Human space travel is an inherently risky prospect, one with little room for error. From its first human mission the United States (US) has sought not only to minimize that risk with stringent safety requirements, but also to offer the possibility of astronaut escape in at least some portions of the mission. However, the only way to ensure the complete safety of an astronaut is to never launch a mission.

> The design problem simply stated is to provide the highest crew safety attainable at a minimum expense of weight, volume, and complexity on the space vehicle design. [1]

In the initial years of the space program, when all the technology in use was new and relatively untested, a large amount of thought and effort was put into astronaut escape methods. As the technology, and the confidence level in it, improved less thought was put into escape from the vehicle and more into the reliability of the vehicle itself, until in the current vehicle designs there are few or no options for an escape from the vehicle should something go drastically wrong. However, with a new vehicle in the preliminary design phase, and the memory of the most recent space accident still fresh, the ideas for emergency escape from a vehicle are resurfacing.

Ideas for escaping from a doomed vehicle go back to the beginning of the space program. Wernher Von Braun sketched out his concept of an emergency escape system even before the first space vehicle had been designed. (figure 2.1) [2]. The Mercury capsules were equipped with emergency escape rockets to pull the capsule off the launch vehicle in case of rocket failure on liftoff (figure 2.2). In the early 1960's General Electric developed the MOOSE escape system (figure 2.3) as an emergency escape for a singular astronaut from orbit [3], and on the International Space Station, the number of permanent astronauts is limited by the capacity of the Soyuz vehicle on board, so that in the event of a catastrophic emergency all astronauts aboard will have a means for leaving the station. With the shuttle system the United States shifted its focus from systems of escape, to system reliability. However, the tragedies of two lost shuttles have proved that this method, while necessary, is not sufficient.

While it is only mildly difficult to build escape capacity into some parts of the mission cycle, allowing for escape throughout the mission is a much more difficult problem. The extreme temperatures, accelerations and turbulence seen during the harshest periods of a launch or reentry trajectory make for a much more difficult problem than the relative calm of orbit or the final subsonic stages of reentry. In addition, the characteristics that would make an escape vehicle ideal in one stage of the mission might be detrimental in another. A major part of solving this problem is coming up with the characteristics that would allow for a successful compromise between all the mission stages.

While it is unlikely that the United States' current human rated vehicle, the space shuttle, can be retrofitted to allow for an emergency escape, a new system is in the design process and will most likely be required to allow for that option for at least some portions of the mission. What, to this date, has not been studied is whether the harsh conditions seen on launch, and even more so on entry, would allow such a system to be designed to work at all points in the mission.

1.2 Thesis Objective

This thesis will present the results of an analytical study into the feasibility of a lifeboat style escape vehicle throughout an orbital mission. The study simulated a lifeboat escaping from a shuttle-like trajectory. Escape trajectories were plotted on a near continuous basis, with key conditions noted in each case. Results of this study will be discussed in chapter 5 along with the feasibility of building lifeboat capabilities into a design for lifting space vehicles. Figure 1.1 shows a conceptual lifeboat escape sequence.

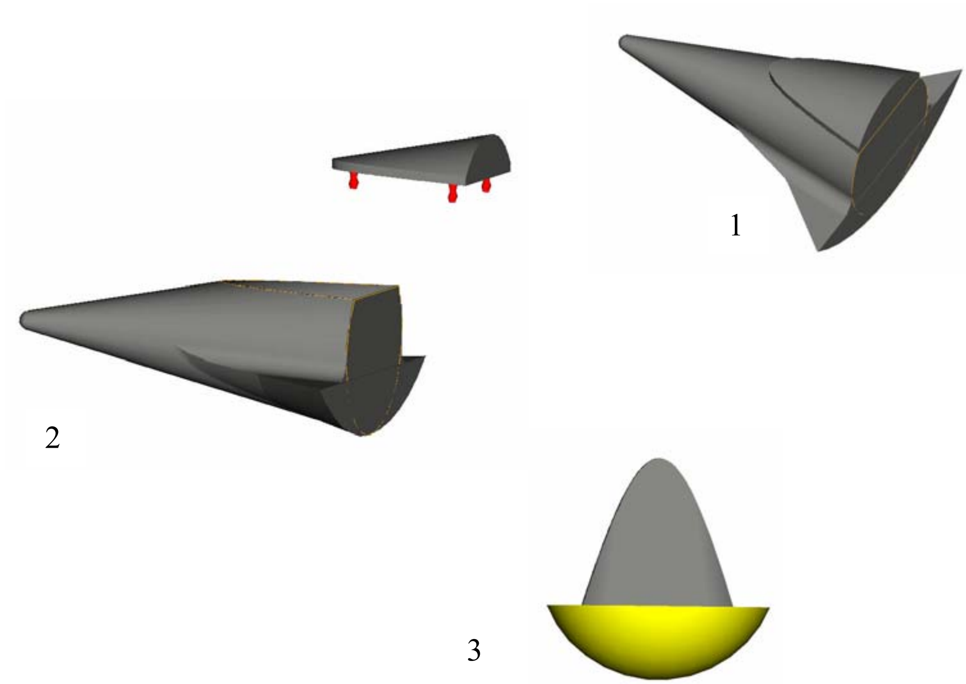

Figure 1.1 Conceptual Escape Sequence. 1. Lifeboat begins attached to nominal vehicle. 2. Lifeboat ejects from nominal Vehicle during emergency. 3. Lifeboat deploys heatshield to continue reentry away from doomed nominal vehicle

Chapter 2: Historical Mission Contingency Designs

2.1 Conceptual Escape Options

The space race began when rockets were still a very new and unstable technology, and that was the best understood part of a space mission. It was well understood that spaceflight would be a dangerous prospect, and that in case of emergencies rescue would be very difficult if not impossible. To alleviate these hazards much thought has gone into which portions of a mission are most dangerous, and how these dangers can be averted.

The least reliable portion of any space mission, especially in the early years of the space program, is the launch itself, due to unreliability in the launch vehicles. Therefore most proposed and enacted escape designs were for the periods before separation from the launch vehicle. In general these designs are referred to as the Launch Abort System (LAS). Within that period, three problem periods were pinpointed: liftoff, transonic through maximum dynamic pressure, and shutdown and staging. Within these periods, the various failures that could occur could give the astronauts warnings ranging from less then half a second up to ten or more seconds [1]. In addition, when emergencies involve the launch vehicle there is a large chance that an explosion may occur. In cases of explosion there are extra hazards that must be dealt with. The first is the shock wave, which can be fatal to both astronaut and craft. Acceptable shock wave levels were set at below 15 psi for .1 second for astronaut survival, and below 5 psi for .1 second for eardrum rupture [1]. Other hazards from explosions include fireballs, shrapnel, and thermal radiation. All of these hazards occur in the first few seconds after the explosion, with their danger levels dropping quickly after an initial time period and distance has been put between the incident and the escape craft [1]. The duration and distance required, however, depends uniquely on the type of launch vehicle and fuel being used. In addition the maximum acceptable acceleration pulse is set at 20 g's, with maximum sustained acceleration of 10 g's [3].

An aspect of the assumption that the least reliable portion of the craft was the launch vehicle, is that all designs involved simply separating the crew capsule from the launch vehicle. While this has and will save lives, it leaves few options if the malfunction is not with the launch vehicle, as has happened several times in both the US and Russian space programs.

The one exception to the stay with the ship mentality was a purely theoretical design. In 1952, years before the advent of NASA's Project Mercury, Wernher von Braun proposed a system to allow astronauts to escape from a doomed vehicle during the launch phase. The system consisted of a series of individual capsules that could eject from the bottom of the vehicle, dropping the inhabitants towards the ground where they would be slowed by a parachute and a small solid rocket before landing. A radar beacon would then activate to allow the survivors to be picked up by rescuers. The capsules were to be pressurized, with the ability to perform basic life support tasks such as CO_2 scrubbing and temperature control (figure 2.1) [2]. Von Braun's concept was for a large lifting vehicle. When small ballistic capsules were decided on as the mechanism for America's first ventures into space, it was decided that it was much safer to leave the astronauts in their spacecraft.

In 1958, while working on Project Mercury, Maxime Faget came up with the concept of the Escape Tower Abort System. The idea was that in case of a launch abort

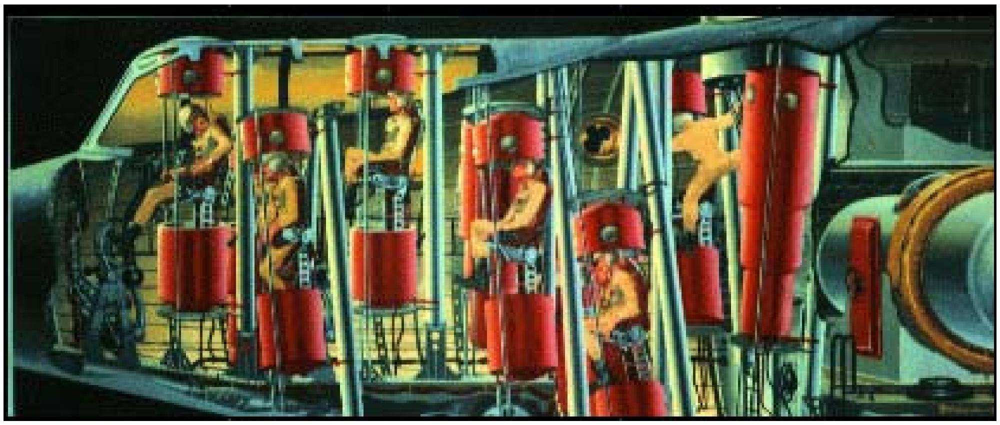

Figure 2.1 Werner Von Braun's Astronaut Escape Concept, circa 1952

the small, powerful solid rocket (figure 2.2) would quickly pull the Mercury capsule away from the volatile booster with a twenty-g acceleration burst. The capsule could then renter normally, parachuting itself and the astronauts aboard it to safety. This concept was used with some modification in the American Mercury and Apollo programs and copied by the Russians for use on their own systems. While the American version was never tested, the Russian version succeeded in saving the lives of two cosmonauts in 1983 [4].

Instead of an escape tower, the Gemini vehicles were equipped with ejection seats that could be used in the first minute of launch before the speed becomes too great. In the case of emergencies after this point the Gemini craft would fire its retrorockets to separate it from the booster and then open the main parachutes for the return to Earth. An additional change in the Gemini configuration was that a human override was introduced into the system. In the Mercury system, if impending vehicle failure was sensed the abort signal was automatically given, firing the escape tower. Because the fuel in the Titan

Figure 2.2 Mercury Capsule with Launch Escape Tower

rocket was less explosive and abort decisions would have to be made on the order of seconds rather then microseconds, it was decided to give the crew the final abort decision. In Gemini, potential failures were displayed to the crew at which point they had to initiate an abort. This allowed for a human override in situations where those on board might have the best idea of what conditions actually existed. The Apollo escape system was between these two designs. In cases of immediate impending doom an automatic abort would launch the capsule away from the launch vehicle. In case of less dire emergencies, the decision to abort was left to the astronauts [5].

During the Apollo era there were also plans for rescuing an astronaut from orbit, though none of them got much past the development stage. Most were based on similar design concepts, and the best developed and most famous of these was the General Electric Manned Orbital Operations Safety Equipment (MOOSE) Program (figure 2.3). The idea was that a pressure-suited astronaut could strap the MOOSE system to his back, jump out of his spacecraft and safely reenter to Earth. MOOSE consisted of a folded heat shield, a canister of polyurethane foam, a chest mounted parachute, and a handheld retro motor. After bailing out the astronaut was to pull the deployment cord which expanded the heat shield and encase him in form fitting foam. The astronaut would then hand aim and fire the retrorocket. After a ballistic reentry he would pull the rip cord and parachute in for a landing. The system was remarkably light, weighing in at about 215 kilograms including the astronaut [3].

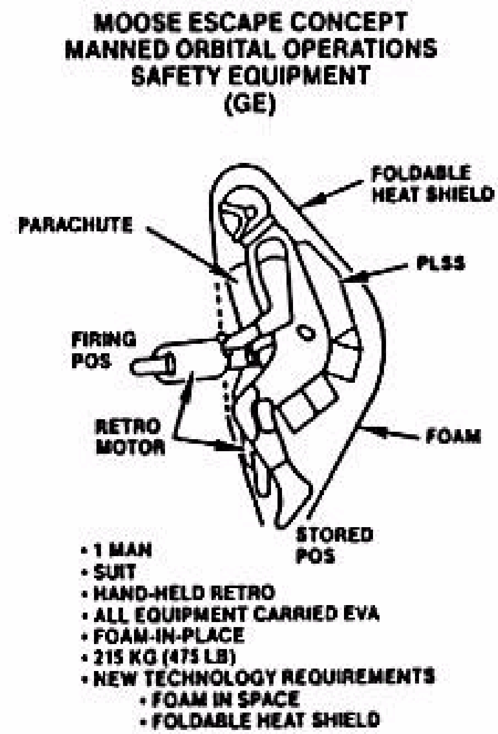

Figure 2.3   MOOSE Escape System

During this same period an orbital escape vehicle was being developed by the airforce based around the concept of an inflatable reentry paraglider that had been used in a suborbital micrometeoroid experiment. The idea was to attach the folded lifeboat to the outside of a spacestation. Upon abandoning the station an astronaut would crawl into a small pod while the wing of his vehicle was inflated with nitrogen stored on the station. Once inflated, retrorockets could be fired bringing the lifeboat into a reentry trajectory. Once into the atmosphere the vehicle could be steered using wing warping. The advantage of this style vehicle was that it allowed for a very large landing footprint. However, after it became apparent that no space station would be built, and such a vehicle would not be able to fit in an Apollo sized capsule funding for the research dried up [6].

With the advent of the shuttle era, and the success of the previous missions, the focus of the safety community shifted from bailout and rescue to reliability. As shuttle plans became more all consuming of the US space community, all talk of bail options dwindled. After the *Challenger* disaster, a special partial pressure suit, the Launch and Entry Suit (LES) was designed to protect astronauts from depressurization at high altitudes, and to provide survival gear and a parachute in case of the necessity of bailout. This was replaced in 1995 with the Advanced Crew Escape Suit (ACES), which was similar except that it is a full pressure suit. It was decided that it was impractical to retrofit the shuttle with any other bailout options [3] [7].

A last design that, while not directly relating to bailout or rescue, is important to this thesis is the Parashield design (figure 2.4) created by a group of MIT graduate students in 1988. Parashield was an ultra light weight, very low ballistic coefficient, reentry vehicle. The design was based on the theory that with a low enough ballistic coefficient a vehicle would slow down high enough in the atmosphere that only a very light weight heat shield would be required to protect the vehicle. Since the heat shield is one of the heaviest parts of any reentry vehicle's design, this allows for large amounts of mass to be cut from the design. A scale model was built and launch was attempted on an experimental rocket by the American Rocket Company (AMROC) in 1988. The launch failed, but all research and

Figure 2.4 Parashield, lightweight, low beta reentry design

simulations seem to indicate that the concepts Parashield was based on are sound. Since in any lifeboat design weight is going to be a major driving factor, the concepts developed in Parashield are beneficial to keep in mind when working with lifeboat design.

2.2 Mission Abort Modes

Both the American and Soviet space programs have kept a long list of abort scenarios for each of their vehicle designs. For all vehicles, these scenarios tend to fall into one of five categories; Abort to Orbit, Abort Once Around, Abort to Transatlantic Landing, Return to Launch Site, and Launch Escape Systems. In general, any given vehicle has at least three of these options available in its design. Below the four shuttle abort modes are described (figure 2.5). The aborts were similar in previous US and soviet vehicles and those aborts are not specifically described. The fifth and final developed abort mode explained below, a Launch Escape System, is not an available option on the US space shuttle.

In any abort scenario there are several main points of damage that must be avoided. The first, in the case of booster failures, is the product of any explosion that occurs, mainly fireballs, shockwaves and shrapnel. The second main concern is the aerodynamic forces caused by the velocity and atmospheric density at which the vehicle is traveling. In this case the higher in altitude that the abort occurs the easier the abort will be due to the decreased atmospheric density.

2.2.1 Return to Launch Site Abort (RTSL)

A Return to Launch Site Abort involves an abort very soon after launch. In this case the orbiter flies downrange to dissipate propellant before returning to fly in under power for a landing at or near the launch site. In a capsule style vehicle the abort mode is somewhat different. In this case the capsule would detach from the booster at some point before normal main engine cutoff (MECO) and come down in its normal landing mode.

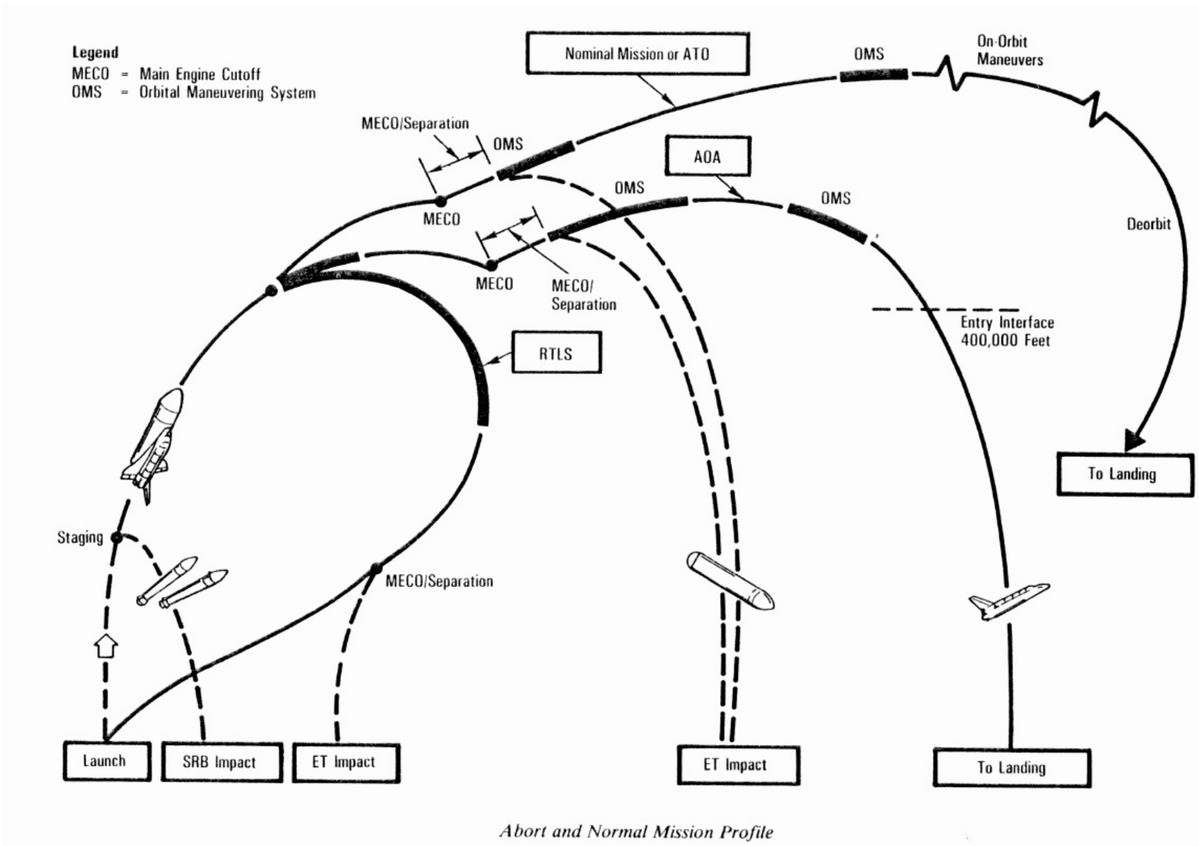

Figure 2.5 Shuttle Intact Abort Modes [8]

2.2.2 Transatlantic Abort Landing

In a Transatlantic Abort Landing, the vehicle does not have the boost to maintain a stable orbit, or even a single orbit, but is too high at the time of the abort to return to the launch site. Instead, the vehicle would take a suborbital trajectory and land some distance away. In the US scenario this would be a landing site in Europe. In the scenario developed by the former Soviet Union, the landing site was near the Chinese border. In the space shuttle transatlantic abort scenario a ballistic entry is utilized, meaning that attitude control is not necessary during the landing so the orbital maneuvering system does not need to be functional for this abort to take place.

2.2.3 Abort Once Around

In an Abort Once Around, the vehicle does not have enough boost to maintain a stable orbit, but instead orbits once before coming in for a normal landing. In order for this option to be viable, the Orbital Maneuvering System (OMS) must still be functioning.

2.2.4 Abort to Orbit

In an Abort to Orbit scenario, either a booster failure occurs late in the launch or no outright failure occurs, but for some reason less thrust is obtained than was expected. In this case, the vehicle obtains a lower orbit than was expected. At this point mission control has time to investigate the failure and can attempt to characterize its nature and severity before either attempting to boost the vehicle to its proper orbit or bringing it down for an early landing.

2.2.5 Launch Escape Systems

The last intact abort option, not available on the space shuttle, but used in almost every capsule style vehicle flown is the Launch Escape System. This option is intended for use in the seconds leading up to and directly after engine ignition. Should a failure occur in the engine immediately around launch this system is designed to pull the occupied capsule away from the booster as quickly as possible so that, should the booster explode, the occupied capsule would not be caught in the explosion. In general this has been accomplished with a very short duration high thrust rocket burn. The capsules are generally separated from the boosters at close to 20 g's of acceleration. The system was designed to give the capsule enough altitude to not only remove it from the shockwave and debris of an exploded booster, but also that the parachutes could be safely deployed to bring it back for a landing. [9]

Chapter 3: Historical Incidents

3.1 Historical Emergency Procedures

Unfortunately a true test of any emergency procedure cannot occur until there is an actual emergency. The individual systems can be tested, as well as how the system functions as a whole, but until the system has been seen in action it is difficult to predict exactly how well it will function as a whole. Also it is a truism that the biggest challenge that will be faced is the one that cannot be planned for. In the almost fifty year history of the manned space travel, about one third of the missions have had some kind of anomaly [3]. One crew was rescued by a launch bailout system, four crews have perished, and several others were in situations that could have proved fatal. Not every tragedy could have been averted with a lifeboat system, such as the one being suggested in this thesis, and not every anomaly results in the loss of ship or crew. Below several situations are described, along with an analysis of what emergency procedures were in place and what additional emergency facilities could have been utilized. Special emphasis is placed on whether some form of escape vehicle would have been useful in these situations.

3.2 Mercury MA-6

On February 20, 1962, after a month of delays, John Glenn launched in Friendship 7 for the United States' first manned orbital mission. During the first orbit a bent shaft on one of the rotary limit switches moved and broke an electrical contact leading to warning signals being transmitted to Mission Control that the heat shield had become detached from the spacecraft and was only being held on by the three titanium straps that attached the retrorocket for reentry [10]. While trying to ascertain if they were getting a true reading, and while trying to come up with a contingency plan it was decided to keep Glenn in the dark about the situation.

After conferring with the spacecraft and heat shield designers it was decided that Glenn should reenter with the retrograde package still attached and hope that if the heat shield was actually loose the straps would hold it in place until air pressure became sufficient to do the job. However, should the retrorockets not fire completely they would have to be jettisoned to avoid the chance of a fireball as propellant in the tanks caught fire.

As the mission continued experts on the ground debated the likelihood of a faulty signal, with instructions to Glenn about reentering with the retrograde package fluctuating as he passed over various control stations. Eventually it was decided to enter without jettisoning the retrograde package. The pack burned through partway through reentry, causing straps to flop over the window, and chunks of the retrograde package to go flying past the window.

This early in the history of manned space it was unlikely that any kind of escape vehicle could have been successfully incorporated, as the technology was barely at a high enough level for the systems that were built. What this incident shows however is how early on the dangers of unexpected problems occurring in orbit became apparent, and how much these could effect the reentry process. Also apparent was that due to the distance and environment, there is room for a lot of uncertainty over what exactly and how serious the malfunction is. The information available on the status of the craft is only as good as the sensors reporting it. Sensor malfunctions can add serious hazard to the crew, as they may cause emergency procedures to be enacted when there is no actual emergency. Glenn was in a perfectly functional craft, but because of a malfunctioning sensor he was forced to deviate from standard reentry procedures.

3.3 Gemini 6

On the morning of December 15th 1965 Walter Schirra and Thomas Stafford were to launch in Gemini 6 on a mission to come within docking distance with the crew of Gemini 7 already in orbit. At launch a plug fell out of the tail of the Titan rocket activating a cockpit clock that was not supposed to start until liftoff. The malfunction detection system on the Titan rocket caused the burn to shut down after 1.2 seconds due to the lack of upward motion. By mission regulations the astronauts aboard should have ejected from the capsule at this point since if the vehicle had moved even a centimeter it would have collapsed back on the pad and exploded. However Schirra, the commander for the mission, in a split second decision decided that there had been no motion and therefore the booster would not settle back on the pad and explode. His instincts told him that at that moment it was safer to remain with the craft. Schirra's decision saved the mission, and also himself and his crewmate some probably injury. It also allowed the launch to be rescheduled in 3 days, and the docking to be successfully completed, even if at the moment his only motivation was that he believed it safer to stay with the ship[11] [12] [13].

What this incident tells us is the value of allowing the human decision into the abort process whenever possible. Had the abort been in the automatic control of the computer, or even under the command of mission control as in the Russian system, the mission would have been a failure. The use of the launch escape system in this situation would have been riskier for the crew, and destroyed the crew capsule. Allowing the experience and senses of the astronauts aboard to make the final decision saved the mission.

This is not the case in all launch emergencies. In at least some cases the danger is beyond human reaction time and a computer must be allowed to control the abort. However when the emergency allows for enough time, there seems to be value in allowing human judgment to make the final abort call.

3.4 Gemini 8

Neal Armstrong and David Scott launched in Gemini 8 on March 16, 1966. Their mission was to dock with the Agena target vehicle already in orbit. The docking was accomplished uneventfully about 6 ½ hours after launch. When the crew began trying to maneuver the combined vehicles they went into a spin. Thinking the error was in the target vehicle they separated, but this only caused the spin to get worse. At its worst the spin was end over end at about one revolution a second, enough to cause dizziness and bring the crew close to blackout.

After disengaging the Orbital Attitude Maneuvering System (OAMS), Armstrong and Scott checked that the Reentry Control System (RCS) was still functioning. They then turned on the OAMS thrusters one at a time, finally determining that thruster number 8 had failed ‘on’ such that it was continuously firing. Armstrong then used the thrusters in the RCS system to stabilize the craft. Emergency use of RCS thrusters meant that the mission had to be terminated early, since if any further thrusters failed the crew could be stranded in orbit. Gemini 8 splashed down in a contingency landing site in the Pacific southeast of Okinawa 10 hours and 41 minutes after it launched [11] [12] [13].

This incident was the first serious orbital emergency encountered by the United States, and in its initial onset, similar to the incident that would cause the loss of Soyuz 1 in the next year. This incident proved the usefulness of having redundant systems. Had the crew not had the fully independent RCS system available to them the outcome could have been much different. The ability to turn off the OMS ring and use the completely separate system allowed them to bring the vehicle under control, and prevent the mission from ending with fatalities.

Had they not been able to bring the craft under control with the RCS or had the RCS failed to respond there may have been no way to bring the craft back safely. Had that been the case, some form of orbital escape system may have been of value, assuming it had the necessary thrusters available to stabilize the spin. As it was the availability of the redundant attitude control system in the RCS saved the crew and the craft even though the remainder of the mission had to be aborted.

3.5 Soyuz 1

Soyuz 1 was launched on April 23, 1967, piloted by Vladmir Komarov. A second Soyuz vehicle was expected to launch later that day, but the fact that it did not suggests that the Soviets knew of difficulties with the mission early on. While the exact failure was never released, it is suspected that one of the solar arrays failed to deploy properly, leaving one of the ship's radiators covered thus preventing the heat generated by the ship's electronics from dissipating as planned. This appears to have led to the failure of the attitude control system, causing the ship to tumble end over end. During the 17th orbit, Komarov put the vehicle into a spin to attempt to keep it stable during the reentry burn. During the 18th orbit the retro rockets were fired putting the capsule on a ballistic trajectory. The resulting entry would have encountered very high g accelerations and left the vehicle short of the normal landing zone.

As the vehicle was reentering the main parachute deployed, but because of the tumbling and rotating of the capsule its straps became tangled. Komarov then tried to deploy the backup parachute, which became entangled in the main chute. While the other failures contributed to the situation it was this final failure that proved fatal, as the capsule hit the ground at an estimated speed of 500 miles per hour [4].

It is not apparent whether a lifeboat would have been appropriate in this case. While the ability to eject may have saved Komarov it appears that a reasonable number of backup procedures were already in place. Based on what is known of the failures a lifeboat ejection could have been attempted at two places. Had there been a lifeboat available it is possible that Komarov could have ejected in orbit before attempting reentry in a vehicle with no attitude control. However at that point since the majority of systems required for landing were intact it is likely that Komarov would still have attempted reentry with the ship. The second point at which an ejection would have been likely was after the backup parachute failed to deploy, however by this point that seems like almost an excessive level of backup procedures. It appears in this case, that rather then a full lifeboat a redundant attitude control system would have been a much more efficient and economic infrastructure investment.

3.6 Soyuz 5

On January 15, 1969 Soyuz 5 launched, a day after Soyuz 4, on a mission for the first Soviet docking of piloted vehicles. The crew on this mission, Boris Volynov, Yevgeniy Khrunov and Aleksey Yeliseyev, flew up to meet Vladimir Shatalov. These missions represented the first Soviet manned space effort since the Soyuz 1 failures almost 2 years earlier. The mission proceeded as planned with the vehicles performing a hard dock, and two of the crew members, Khrunov and Yeliseyev, performed and EVA transfer to Soyuz 4 leaving Volynov alone in the Soyuz 5 capsule. Soyuz 4 then reentered first landing on January 17, on target in Kazakhstan.

At 10:20 the next morning Volynov began his reentry with his retroburn. After this the three sections of the spacecraft were supposed to separate leaving the entry module unencumbered for the return trip. However, when Volynov looked out the window he could see the solar panels of the service module still attached. At this point there was nothing Volynov or the ground control could do, and as the combined modules began tumbling into the atmosphere everyone, including Volynov, was convinced that he would not survive the entry.

Because of the attached service module the heat shield was not taking the brunt of the heat exposure, causing the crew cabin to begin to fill with smoke as insulation began burning. At this point in the entry attitude control jets were supposed to fire to steady the vehicle, however all the fuel had been spent during the initial burn in an unsuccessful attempt to adjust the attitude of the combined modules. Finally the intense heat caused the hydrogen peroxide tanks to explode. The force of the explosion was enough to force the crew hatch inwards, but it also finally blew off the service module.

Volynov then had a second scare when, after the main parachute deployed, its straps began to twist. Once again there was nothing anyone could do but wait and hope. The straps eventually untwisted allowing the chute to slow the capsule enough that the soft landing engines were able to slow the vehicle enough for a survivable landing. Volynov survived the landing with injuries to his upper jaw, but no broken bones or other major injuries[4] [14].

This situation would have warranted a lifeboat ejection had one been available. From the moment Volynov realized the service module had failed to separate there was no question that the entry was going to be hazardous and very likely fatal. Had the Hydrogen Peroxide tanks blown a few seconds later the capsule would likely have burned through and survival would have been impossible. While a combination of luck and good design of the Soyuz craft saved Volynov's life, an emergency lifeboat ejection would likely have been less hazardous.

3.7 Soyuz 11

On June 6, 1971 Soyuz 11 was launched on a mission to the Salyut 1 Space Station. The setup for this capsule allowed for a three man crew, however the volume was to small too allow them to wear pressure suits. After a nominal month long mission the crew mothballed the station and prepared to reenter in their Soyuz capsule. This version of Soyuz was designed with twelve retrorockets meant to fire sequentially upon reentry. In this case they fired simultaneously, causing, a seal on the capsules pressure equalization valve to open. This valve is only supposed to be opened at low altitude when there is a breathable amount of air. The cabin lost all atmosphere in approximately 30 seconds at an altitude of approximately 168 km, killing the crew.

While there was a manual shutoff for this valve the procedure to enact it took over a minute, giving the crew no chance of survival. The crew could have been saved if they were provided with pressure suits for the reentry portion of the flight, a protocol that became standard procedure after this flight. This was the last three man crew launched until a new Soyuz design was produced with enough volume for a three man crew to launch while wearing pressure suits [4].

In the final analysis, this situation would not have warranted a lifeboat ejection, however the ability to retreat to a separately pressurized lifeboat compartment, even if the ejection was not completed, would have been valuable. A pressurized lifeboat may be a solution to time critical life support emergencies, as well as situations of full vehicle failure. In this case the cosmonauts could have lived off a lifeboat life support system for the duration of the entry. The only other mission wide option for life support failures is backup pressure suits for every astronaut. There, however, are heavy, and difficult to don quickly. An easily sealed section of the vehicle or lifeboat with a separate short term life support system might provide an extra level of security to the crew throughout the mission. Should the problem be solved they could reenter the main vehicle. Should the situation continue they could eject without having to come back into contact with the contaminated system.

3.8 Soyuz 18

Launched on April 5, 1975, the mission was intended to dock at the Salyut 4 Space Station. During launch there was a malfunction in the A-2 booster where a sequencer relay prematurely fired two of the four pyrotechnic latches that held the core stage to the upper stage. This also disabled the remaining two latches so that the core stage could not be detached after shutdown causing the upper stage to begin firing while the core was still attached finally resulting in the launch trajectory begining to deviate. When the flight path deviated 10 degrees from the nominal the Launch Abort System automatically activated.

The Launch Abort System functioned nominally, shutting down the upper stage and separating the spacecraft from the booster. After separation, which occurred at 180 km and a velocity of approximately 5.5 km/s, the spacecraft reoriented for reentry and proceeded to retroburn into a ballistic reentry trajectory. The cosmonauts saw as much as 20 g's during their reentry and had absolutely no control over the system. They landed close to the Chinese border and were located within a few hours.

This system was the Soviet equivalent of a U.S. Space Shuttle abort to Transatlantic Landing [8] or a Gemini retro-rocket abort [1]. The error occurred late enough in the launch that the vehicle was already at close to orbital velocities. In addition the abort took place above the thickest part of the atmosphere. Thus the capsule did not have to worry about aerodynamic forces during the crucial time while it was reorienting for landing.

3.9 Soyuz T-10A

On September 26, 1983 the Soyuz T-10 mission was scheduled to launch for the Salyut 7 Space Station. Ninety seconds before liftoff a malfunction allowed fuel to spill out around the base of the booster. With less then a minute left before launch the spilled fuel caught fire causing ground control to trigger an abort. Unfortunately the communication lines to the capsule had already burned through and the command was not transmitted. The command was resent twenty seconds later over the backup radio system causing the Launch Escape System solid rocket motors to ignite. Unlike in the NASA system, the Soviet system did not allow the cosmonauts to independently activate the abort system from the capsule. The boosters exploded on the launch pad six seconds after the system was activated.

The escape rocket fired for five seconds pulling the cosmonauts away from the booster at a 17 g acceleration. After this smaller sustainer motors fired and air brakes deployed to keep the shroud vertical. Pyrotechnics then fired to separate the capsule from the orbital module allowing the capsule to drop out of the launch shroud. This all occurred at approximately 650 meters altitude. Once the capsule was out of the shroud the heat shield jettisoned and the emergency parachute opened bringing the capsule back to Earth close to the launch complex. Both the cosmonauts were uninjured though understandably shaken [4].

This was the only live use of any of the any of the very similar launch escape systems used by both NASA and the Soviets. The system functioned as planned pulling the crew away from the doomed boosters before a fatal explosion could occur.

3.10 STS-51-F Challenger

All of the above examples, both from the U.S. and the Soviets, have been in capsule style vehicles. In 1982 the U.S. began launching manned missions solely on the new space shuttle. Unlike its predecessors this was a winged vehicle, which while launching in a similar fashion to previous vehicles, reentered on a much different and more complex, trajectory then had been used before. The first abort situation with this new vehicle occurred during the ascent portion of its 15th mission that launched on July 29, 1985.

The mission, commanded by Gordon Fullerton with Roy Bridges Jr. as pilot, launched normally. However, after the Solid Rocket Boosters (SRBs) were jettisoned the center engine sensors showed the engine beginning to overheat. Flight controllers initially throttled that engine back to 65%, and T+645 seconds after launch, when the engine continued to overheat, the General Purpose Computer (GPC) automatically shut that engine off. By this point of the ascent the shuttle had sufficient altitude to achieve an abort to orbit with only two engines. Fullerton was told to turn the abort switch to "ATO" which throttled the remaining two engines up to 91% for 70 seconds to compensate for the lost engine. At this point however, one of the remaining engines began to report a temperature rise. With two engines out the shuttle would be forced to attempt a riskier TAL abort. However, with this second anomaly, flight controllers began suspecting a faulty sensor, and recommended that the crew override the GPC and keep the second engine from shutting down.

The shuttle initally obtained a 120 km orbit, which was later raised to 275 km using the maneuvering system. While not ideal for the mission that had been planned this orbit was better then a full abort and modifications to the mission objectives were made. The rest of the mission was uneventful and Challenger landed at Edwards Airforce Base in California on August 6. After landing, the engines were examined and it was determined that problem was indeed with the sensors and not the engines.

This abort marks the second time in the US space program when faulty sensors almost caused disaster. Unlike in the Mercury mission, in this case contingency plans were in place and functioned as planned, however in both cases emergency procedures were enacted unnecessarily because of faulty readings. Though nothing was truly wrong with the craft, the crew was put in unnecessary danger because of faulty information. While the abort plans worked perfectly in this case they were detrimental to the mission and the overall safety of the crew. The better option would be to have redundant sensors or a way of testing the validity of the sensor readings [7] [15].

3.11 STS-51-L Challenger

On January 28, 1986 mission STS-51-L of the Challenger Space Shuttle was lost 73 seconds after launch when the shuttle broke up due to an O-ring failure in the solid rocket moter. The report released six months after the accident concluded that the forces to which the crew were subjected were probably not enough to kill them, or even to cause serious injury. It is hypothesized based on photographs and the last milliseconds of data reported by the shuttle systems, that the orbiter breakup itself produced forces no worse then a gravity pulse of 20 g's in the vertical axis lasting less then two seconds. That at least some of the crew survived the initial breakup is supported by the fact that at least three of the Personal Egress Air Packs (PEAPs) were activated. The PEAPs contained the emergency air supply for the crew, and had to be manually activated. However, in the case of loss of pressure in the crew compartment the PEAPs would not have been enough to prevent a rapid loss of consciousness, though they probably would have prevented death. What appears to have caused the deaths of the crew, that as the orbiter fell it collided with the ocean at a speed of 207 miles per hour (333 km/hour) [16].

The Challenger accident is the type of situation for which the Launch Escape Systems had been designed. Had the crew been able to separate from the doomed vehicle they would have likely survived the orbiter breakup and been able to return to Earth in a more controlled fashion. However, because of the design of the orbiter, and the focus on reliability instead of escape, no such system was implemented on the shuttle.

3.12 STS-107 Columbia

On February 1, 2003 Space Shuttle Columbia was returning from a 17 day mission when it broke up due to a breach in the heat shield. It is believed that the breach occurred on liftoff when a 0.76 kg piece of foam insulation separated from the external fuel tank and struck the orbiter wing at approximately 236 m/s. The mission appeared nominal until after the orbiter had passed through the entry interface (EI), the transition point between orbital and atmospheric conditions, which occurs at 400,000 feet (122 km). 270 seconds (4.5 minutes) after EI the first off nominal readings began to appear. Conditions quickly degraded as hot gas entered the structure of the wing. The first sensor reading was lost at EI + 487 seconds (8.11 minutes) as the hot gasses melted the wiring bundles. 160 measurements failed in the two minutes following this. The failing sensors were the first indications the crew would have had that something was wrong.

As the structure of the wing weakened and deformed due to the influx of hot gasses, the orbiter attempted to veer from its planned trajectory. Initially this was compensated for by the flight control system, with the only the vehicle sideslip, the angle between the relative wind direction and the velocity vector, exceeding what had been seen in previous flights. During this same period temperature readings continued to rise and debris began to fall from the shuttle. The first debris events were captured on video at EI + 577 seconds (9.6 minutes).

Still, it is unknown at what point the crew of Columbia were aware that they were dealing with a fatal error. The best indication is that at EI + 927 seconds (15.45 minutes) the orbiter's attitude control was no longer able to counteract the increased lift and drag from the left wing, and the sign of the sideslip angle changed, indicating an imminent loss of vehicle control [17]. The main body of the orbiter did not begin to break up until EI + 972 seconds (16.2 minutes) [18], leaving at least 45 seconds in which action could have been taken had an escape vehicle been present. Even had a lifeboat been available, the breakup occurred during some of the most intense moments of the entry. A lifeboat would have had to eject into an incredibly difficult aerodynamic and thermal environment.

3.13 Results of Historical Analysis

The findings of the Columbia Accident Investigation Board (CAIB) Synopsis report that "By the time Mission Control received the data on the wing heating, the damage was unrecoverable" [19]. Columbia and the other missions discussed above show that as much planning and research as one may put into a mission there is always something unplanned for that can go wrong. In many cases this problem may not even be recognized until it is too late. Focusing almost exclusively on reliability issues as NASA has for the duration of the shuttle program is not necessarily the best approach. While a high level of reliability is definitely necessary, it is not sufficient. It is better to prevent an accident then to plan for dealing with one that has occurred, no amount of planning can account for all accidents.

The board also found that

> Throughout its history, NASA has consistently struggled to achieve viable safety programs and adjust them to the constraints and vagaries of changing budgets. Yet, according to multiple high level independent reviews, NASA's safety system has fallen short of the mark [19].

This rather damning review leads to the conclusion that safety will be an even greater focus on the next generation of manned space vehicles.

The above missions also show two main periods of malfunction onsets. The first, during the first minutes of launch, has been an acknowledged hazard period since the beginning of manned space travel. When implemented the various LES and LAS systems have been very effective at preventing disasters during this period, saving the crew in 3 of the 4 times they have been needed, not counting abort to orbit situations.

The second class of incidents are those that begin in orbit and are exacerbated during reentry. Of these, when the incidents occurred in capsule style vehicles, in most cases, contingency plans were robust enough to secure a successful rescue. The exception to this is Soyuz 1, where the issues overwhelmed the contingency plans, mostly due to the lack of redundancy in the attitude control system as evidenced by the earlier survival of the Gemini VIII crew. In any case, in capsule style vehicles few new issues emerged during reentry that were not directly related to the issues already apparent in orbit.

By contrast, while the damage that led to the loss of the Columbia orbiter occurred upon launch, the danger signs did not appear until well past entry interface. This meant that the anomalies encountered came as a complete surprise. Also whereas reentry in a capsule style vehicle tends to be a fairly short and straightforward situation, in a shuttle style vehicle it is much longer and more complex. In a capsule there is a limited amount of attitude control that needs to be accomplished once the vehicle is into the atmosphere. In a shuttle style vehicle attitude control is constant and complex. This makes them much more vulnerable to late occurring anomalies, and makes those anomalies harder to deal with when they do occur.

From these incidents it appears that a Launch Escape System is valuable in all cases, for all styles of spacecraft. Booster errors occur fairly regularly, though often they are mild enough to either not seriously effect the mission, or at most require an abort to orbit. Various launch abort scenarios for later occurring launch errors are also necessary, and in most cases appear to be well identified and operable. By contrast, an Orbital Escape System could be valuable for all styles of craft, but in many cases can be done without so long as there is a sufficient level of redundancy, as in Gemini 8, and in a case of faulty sensors such as Mercury MA-6, might actually endanger the crew further.

A lifeboat capable of ejecting throughout the full mission profile only appears necessary when dealing with a shuttle style craft. The trajectory and controls in capsule style vehicles are simple enough that problems not already present on orbit do not tend to have time to propagate. The exception to this is sudden life support problems which have other, lighter, solutions.

Mission	**Year**	**Incident**	**Crew Lost**
Mercury MA-6	1962	Sensor failure indicating prematurely detached heat shield	No
Gemini 6	1965	Premature Booster Shutdown	No
Gemini 8	1966	ACS Failure	No
Soyuz 1	1967	ACS Failure	Yes
Soyuz 5	1969	Service Module Failed to Separate	No
Soyuz 11	1971	Sudden Decompression	Yes
Soyuz 18	1975	Faulty Booster	No
Soyuz T-10	1983	Booster Fire	No
STS-51-F	1985	Engine Failure due to sensor error	No
STS-51-L	1986	Booster Explosion	Yes
STS-107	2003	Heat Shield Failure	Yes

Table 3.1 Summary of Historical Emergency Incidents

Chapter 4: Simulation Design

4.1 Simulation Focus

The purpose of this thesis was to be an analysis of the likelihood of survival in an emergency escape from a spacecraft throughout its mission. An analysis of all possible vehicles in all possible mission phases is too broad of a scope for this project, so the decision was made to limit it to reentry situations. As described in chapter 2 there are several successful methods of launch aborts which have been demonstrated on both Soviet and American systems. This experience base confirms that there are methods of rescuing crew from a variety of launch emergencies between liftoff and the attainment of the final mission orbit.

Inside the scope of reentry, the decision was made to further focus on shuttle style (high lift) vehicles. High lift entries are generally more complex and stressful than a low-L/D capsule style entry. If an escape is possible for the high lift case, it should also be possible for the simpler capsule style entry.

4.2 Entry Conditions

During descent in a reentry trajectory there are several regions of conditions experienced. In each of these regions a drastically different environmental conditions are experienced. Initially the vehicle is in the orbital conditions of microgravity and no distinguishable atmosphere. As the descent begins the growing atmospheric density introduces drag, which slows the vehicle down, and introduces temperature, sensed acceleration and dynamic pressure increases. Entry interface (EI) is defined as the point at which the acceleration due to drag is greater then 0.05 g's, usually this occurs somewhere near 122 km above the surface [8].

As the descent continues the intensity of the above mentioned environmental characteristics increases, until each peak separately at altitudes that differ depending on vehicle characteristics. The point of peak dynamic pressure (Max Q) tends to be the benchmark point in this region of a reentry trajectory. Shortly after the Max Q point the environmental intensity begins to dissipate until eventually the vehicle goes subsonic and final landing procedures can begin.

As mentioned above, the altitude at which the peak aerodynamic forces are experienced depends on the characteristics of the vehicle. The higher the ballistic coefficient of the vehicle, the lower in the atmosphere deceleration will begin, leading to greater atmospheric density during deceleration and therefore higher temperatures. However the actual peak acceleration will be the same for any given ballistic coefficient. [21].

The vehicle characteristic that controls peak acceleration is the lift to drag ratio (L/D). The higher the L/D of a vehicle the lower peak acceleration it will see. Few purely ballistic entry vehicles have been built because even a very small amount of lift can greatly decrease the peak accelerations experienced by a vehicle on an entry trajectory. This leads to a much more gentle ride for the crew and the vehicle systems themselves. The addition of lift will also increase the landing footprint of a vehicle. This can be beneficial if there is a specific landing target set, as it allows for more leeway in initiating the entry.

4.3 Nominal Trajectory Design

One of the initial requirements for this analysis was a nominal entry trajectory from which to stage simulated lifeboat ejections. As was mentioned before, the US space shuttle was chosen to be used as the baseline vehicle. Effort was made to find an accurate representation of the US space shuttle's reentry trajectory. When no such trajectory could be obtained, it was decided to simulate one based on known metrics of the trajectory. The design characteristics of the nominal vehicle were set to match those of the shuttle orbiter. Other characteristics of the simulation, mainly roll angle and initial flight path angle, were modified until the simulation was close to the characteristics listed in table 4.1. The simulated trajectory is shown in figure 4.2. Further information on the design of the simulation follows in the rest of chapter 4.

4.4 Lifeboat Subsets

Before any simulation was attempted, several theoretical types of ejection zones were defined along a normal reentry trajectory. These zones were defined by the region along the nominal trajectory in which an ejection would be initiated. Zone 1, for

Maximum	**Shuttle**	**Simulated**
Acceleration (g's)	< 3	2.2
Dynamic Pressure	580 PSF 27,770 Pa	365 PSF 17,498 Pa
Temperature (°C)	< 1650	1537

Table 4.1 Comparison of Shuttle and Simulated Entry Trajectories

instance, was defined as the part of the trajectory around entry interface, when aerodynamic forces are still at a minimum. Ejecting in this zone would allow for time to orient the vehicle into the optimum orientation before any intense conditions would be encountered. In addition, the vehicle would only have to be designed to withstand extremes that its own design would experience. There would be no residual accelerations or temperature from the nominal vehicle to deal with. Designs for ejections in this zone were proposed during the Apollo era.

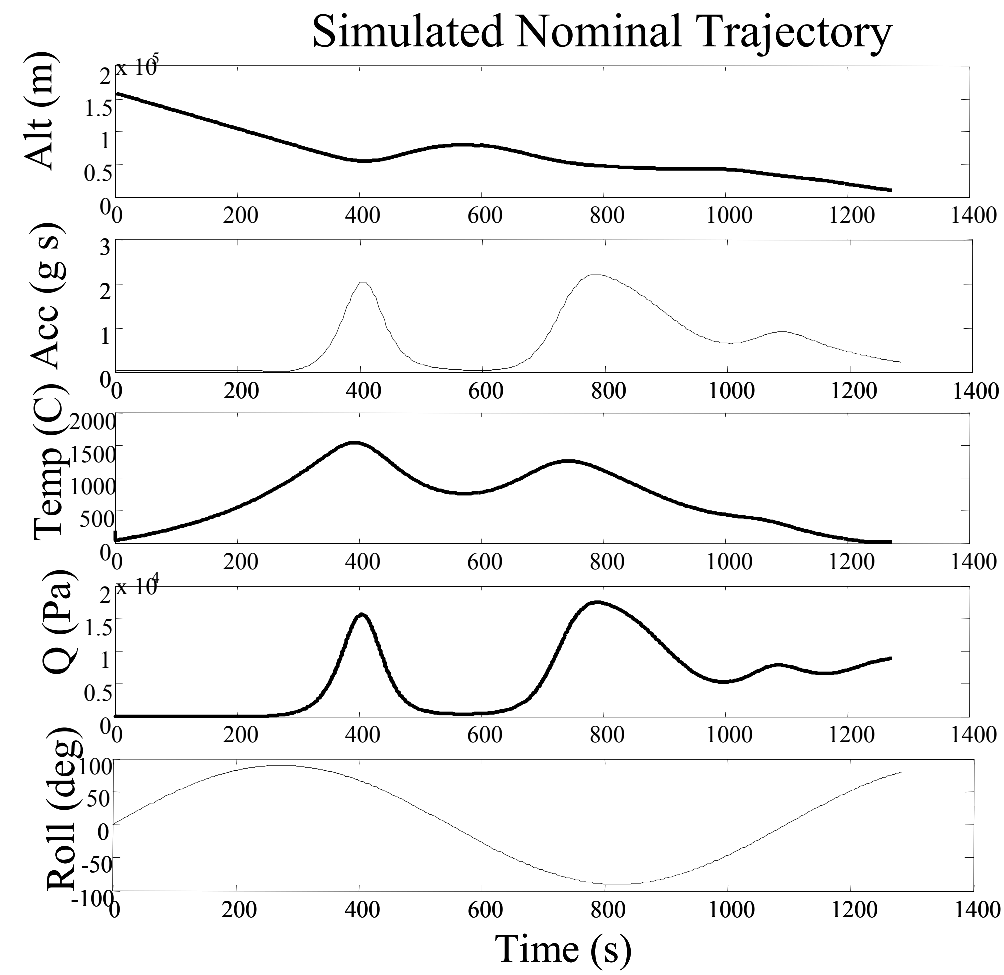

Figure 4.1 Simulated Nominal Trajectory

Zone 2 was defined as the regions on either side of Max Q, before and after the nominal vehicle has experienced the most extreme aerodynamic conditions. Lifeboats in this zone would have to be more rugged then those in zone 1, as ejections in this zone would be ejecting into at least somewhat intense residual conditions. They would have to quickly establish a stable orientation to avoid accelerations and temperatures in undesirable orientations.

Zone 3 would then be the region directly around the nominal vehicle's Max Q. This is the region where the highest temperatures and accelerations are experienced by the nominal vehicle, and therefore would be the least hospitable to an emergency ejection. Vehicles ejecting into this zone would have to be very rugged to withstand the types of temperatures and accelerations that would be residual of the nominal vehicle.

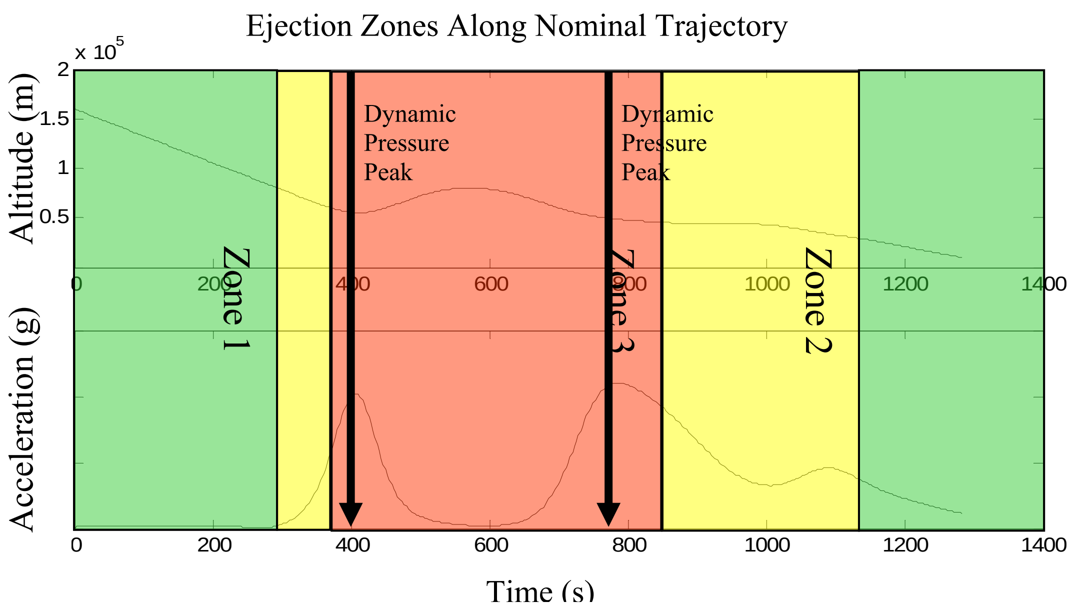

Figure 4.2 Defined Ejection Zones Along Nominal Vehicle's Entry Trajectory

4.5 Simulation Basis

There were two main steps in the design of the simulation. The first step involved using a preexisting spreadsheet simulation [20] to perform tests on selected data points of several different lifeboat designs to look for obvious trends. The second step involved creating a new simulation in MATLAB™ to do the continuous analysis.

Both simulations have the same mathematical basis. The equations of motion are calculated based on differential equations around flight path angle (γ), velocity (v), and altitude (h). Numerically integrating these equations results in the trajectories for both the nominal vehicle and the lifeboats depending on which initial conditions are used.

$$\gamma = D*(L/D) - (1-v/vcirc2) * g * \cos(\gamma) * \cos(\theta_{Roll})/v \qquad \textbf{Equation 0.1}$$

$$v = -D - g* \sin(\gamma)-a_{thrust} \qquad \textbf{Equation 0.2}$$

$$h = v * \sin(\gamma) \qquad \textbf{Equation 0.3}$$

[21]

Temperature estimates on the outer heat shield of the lifeboat were based on the Chapman heating equations for laminar flow. This equation returns stagnation point temperature estimates of the heating rate seen by the craft.

$$q^{\cdot} = 2*(2.568*10^{-9})*11.3538*((q / R_{hs})^{.5})* v^{3.2}$$

$$q = J/cm^2$$

$$R_{hs} = m$$

$$v = m/s \qquad \textbf{Equation 0.4 [23]}$$

All other equations used in the simulation code were based on the results of the above four equations and can be found in the complete code printout in the appendix.

4.6 Weaknesses in Initial Simulation

The preexisting spreadsheet used in the initial simulation used most of the same variables as the final coded simulation, including ballistic coefficient, radius of curvature of the heat shield, lift to drag ratio, and initial flight path angle, to calculate an entry trajectory. While good enough for initial estimates, there were several reasons why this simulation was not capable enough for the final simulation. The main reason was that the spreadsheet implementation made simulating more then a few trajectories at once complex. This made it difficult to accomplish the goal of an exhaustive ejection map with the spreadsheet. Secondly, the coded simulation made it easier to manipulate variables to obtain cleaner trajectories. The spreadsheet simulation incorporated phugoid motion which actual shuttle trajectories are designed to avoid. These phugoids suggested nearly double the peak acceleration actually seen in actual trajectories. In the coded simulation, as with the actual shuttle, roll angles were used to avoid phugoids. While this could have been done with the spreadsheet it was easier in code.

There were two main functions for which the spreadsheet simulation was useful. First, when obtaining the initial data points only hand picked points were used. Therefore especially inaccurate sections of the trajectory could be avoided. Secondly, the spreadsheet was useful in error checking the coded simulation.

4.7 Coded Simulation

Because of the issues discussed above, after initial data points were studied, the next step was to rewrite the simulation in a more flexible platform. MATLAB™ was selected as the language to be used, and the code was written to simulate both the nominal trajectory and the lifeboat escapes. The validated spreadsheet simulation was used as an error check for the code. Some differences were expected between the spreadsheet and coded simulation as different methods of integration were used. 4th order Runge Kutta numerical integration with a three second time step was used in the spreadsheet. The coded simulation used the built in ODE45 numerical integration function. While this function uses the same basic Runge Kutta equations, it makes multiple intermediate calculations inside the three second times step, with intermediate timesteps based on the rate of change of the function.

The code works by first simulating a nominal trajectory based on supplied nominal vehicle characteristics and orbital start conditions. Using start conditions pulled from that trajectory and a second set of supplied characteristics, it then calculates trajectories for lifeboats ejecting at set intervals for the duration of the nominal trajectory (figure 4.3). Final results are then plotted in 3-d mesh plots, with significant results highlighted in 2-d plots.

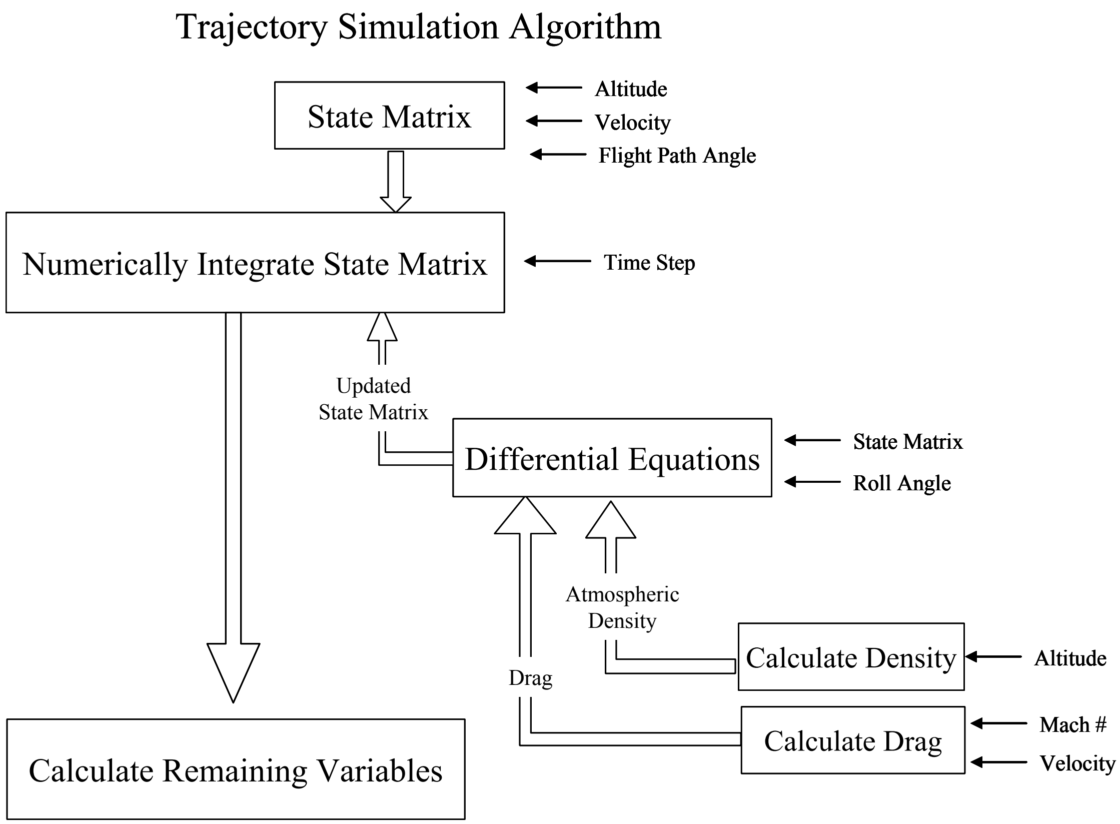

Figure 4.3 Algorithm used for Trajectory Calculations

Computer Simulation Algorithm

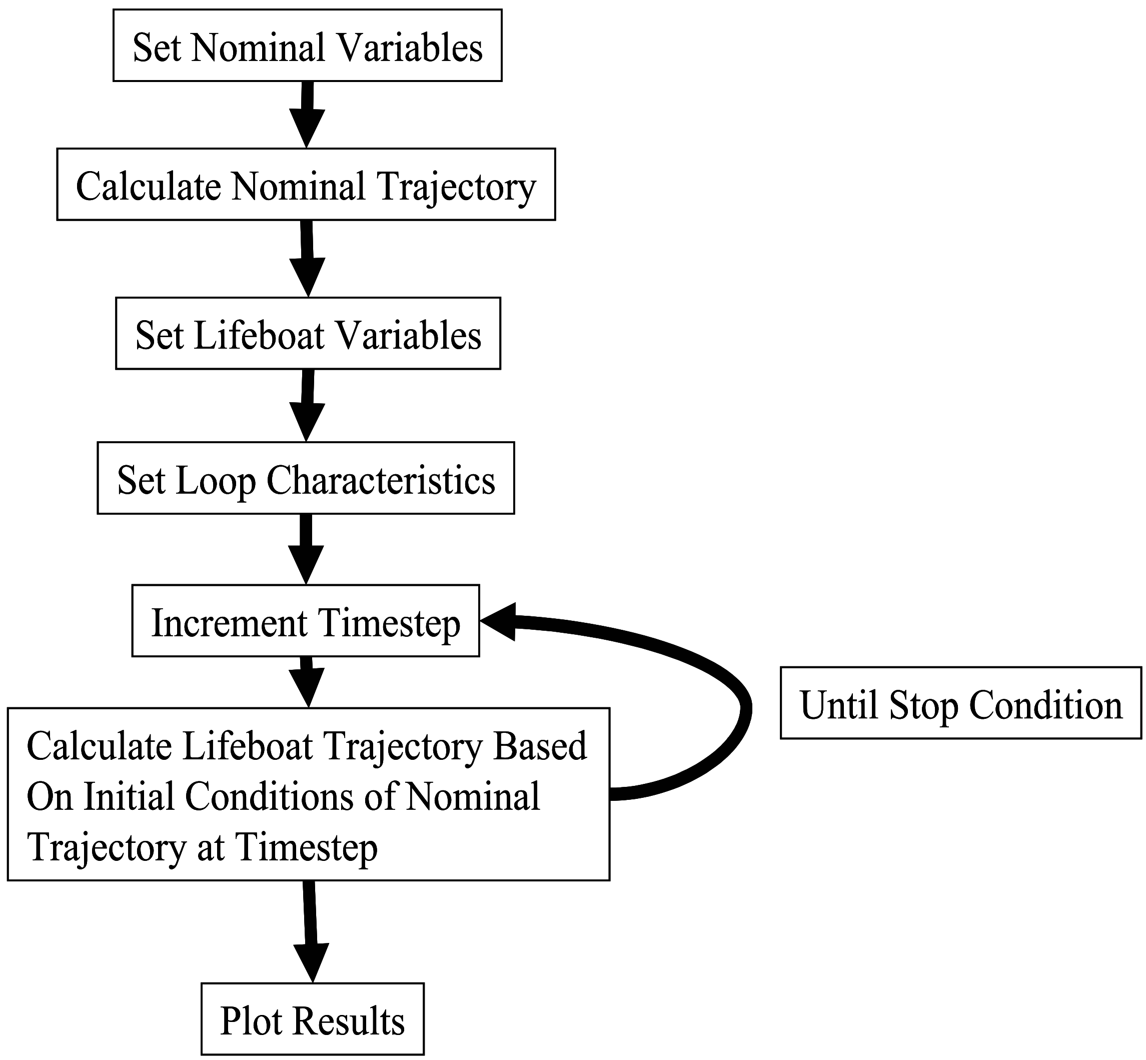

Figure 4.4 Algorithm used in computer simulation

• Roll	• Time	• Flight Path Angle	• Altitude	• Dynamic Pressure	• Downrange Distance
• Crossrange Distance	• Velocity	• Sensed Acceleration	• Mach Number	• Heat Rate	• Outer Shield Temperature
• Inner Shield Temperature	• Total Heat	• Atmospheric Density	• Yaw	• Cp	• Thrust

Table 4.2 Tracked Variables in MATLAB Code

Using the coded simulation, eighteen variables were tracked at each time step, and used to analyze the lifeboat trajectories (Table 4.2). Of these variables the principle metrics for comparing the trajectories were sensed acceleration and outer shield temperature. The calculation of another variable, inner shield temperature, was attempted, by writing a function to perform a finite difference analysis based on the proposed heat shield characteristics. However, due to difficulty in defining boundary conditions, this code was deemed too inaccurate to be relied on. Since inner temperatures are a more important metric with respect to survivability than outer temperatures calculations of inner shield temperature were made using empirical results as reported by the makers of the heat shield materials and other conference papers. Based on these results, first order estimations of inner shield temperature could be made based on the outer shield temperature and the thickness of the heat shield material.

Three of the above variables were calculated as a result of the numerical integration. Flight path angle, altitude, and velocity were all results of the coupled differential equations at the center of the simulation (equations 4.1-4.3). From these three variables the others could all be calculated. The outer shield temperature results were estimated for peak stagnation point heating using the Chapman heating equations (equation 4.4). The time variable represents the cumulative time since the start of the simulation. Each lifeboat trajectory continues counting from the time of its launch so that the point in the trajectory at which the launch occurred is obvious at a glance. The user at the beginning of each simulation run sets an internal variable, deltaT, that represents the time between reported results. As a factor of using ODE45, the integration calculations are made using very small, varying timesteps. The results of these integrations however, are only reported over the set interval of deltaT. In the case of this simulation, that interval was set at three seconds as this supplied a high density of data without overloading either the user or the system. Should greater results density be required, this can easily be changed to any reasonable time period.

The roll variable was used primarily in the nominal trajectory. By creating a roll trajectory that spent the maximum amount of time at high angles of roll, near 90°, the amount and severity of phugoids were significantly decreased. This allowed the final nominal trajectory to mimic that of an actual shuttle trajectory much better than the one that was used for the spot checks.

Atmospheric density was calculated using a simple exponential based on altitude, however should it be required, the function can easily be changed to employ more accurate methods.

Chapter 5: Results

5.1 Data Formats

The data in this project was all obtained by manipulating the spreadsheet and simulation code to examine various escape scenarios. The code was specifically written to maximize flexibility in both the styles of entry vehicle it could model and in how the data returned could be displayed. Nominally several styles of graphs were displayed following each code run. In addition, the complete raw results were saved to a series of text files to be further manipulated if needed.

Three main configurations were used for this project. The first configuration started with a set of initial conditions that produced the nominal trajectory. The second changed the initial conditions, and ran the code through a series of loops to produce the series of lifeboat trajectories. The third configuration used the previous two and added a loop that changed lifeboat design characteristics in search of overall optimum values. As mentioned above, text and graphical output were stored in each run of the coded simulation.

5.2 Initial Simulation Results

In the first steps of this analysis, which used the spreadsheet simulation to look for broad trends which could be refined in the coded simulation, lifeboat ejection points along the nominal trajectory were handpicked to obtain lifeboat trajectories in two ejection scenarios. Since this first simulation was simply looking for broad trends a higher range of conditions was considered than would actually be expected to be encountered. Ejection points and lifeboat variables were all considered at extremes. It was hoped that these extremes would accentuate variations caused either by the conditions at ejection or by the designs of the lifeboats themselves.

The first scenario was a trajectory at an ejection acceleration close to twice what could be expected in a normal shuttle trajectory. This was to be considered the ultimate worst case, and would never be encountered in a shuttle reentry. The second scenario was an ejection at an acceleration more typical of a shuttle style trajectory. The resulting trajectories were compared based on plots of acceleration and temperature (figures 5.1-5.3).

Two different styles of lifeboats (Table 5.1) were checked under these conditions to determine if there were any obvious trends. The lifeboat design was the same for both tests except for the ballistic coefficient, β. Ballistic coefficient is expressed as equation 5.1, and is used to determine the amount of drag on an object. Two different ballistic coefficients were used in the initial tests, a very light lifeboat with β of 100 kg/m^2 and a heavier, more dense lifeboat with β of 1000 kg/m^2. The results of these checks are summarized in Table 5.2.

$$\mathbf{m / AC_d} \qquad \textbf{Equation 5.1}$$

Figure 5.1 shows a plot of how the trajectories unfold over time. The nominal trajectory shows the large phugoids that made the spreadsheet entry trajectory unsuitable for the overall nominal trajectory in this study. The effect of ballistic coefficient in the ejection trajectory is seen in this graph, as the high beta lifeboats

Ballistic Coefficient (β) (kg/m^2)	Heat Shield Radius (m)	Lift to Drag Ratio	Acceleration at Mid-Stress Ejection (g)	Acceleration at High-Stress Ejection (g)	Baseline Trajectory Initial Flight Path Angle (γ) (degrees)
100	2.8	.2	1	3.5	2
1000	2.8	.2	1	3.5	2

Table 5.1 Characteristics of Lifeboat and Check Points in Spreadsheet Simulation

descend in a much steeper trajectory from the moment of ejection than the low beta lifeboats, which maintain a slower, more gradual descent. While this graph is useful in visualizing the trajectories it does not help in deciding on optimal design characteristics. For that the other two graphs must be used.

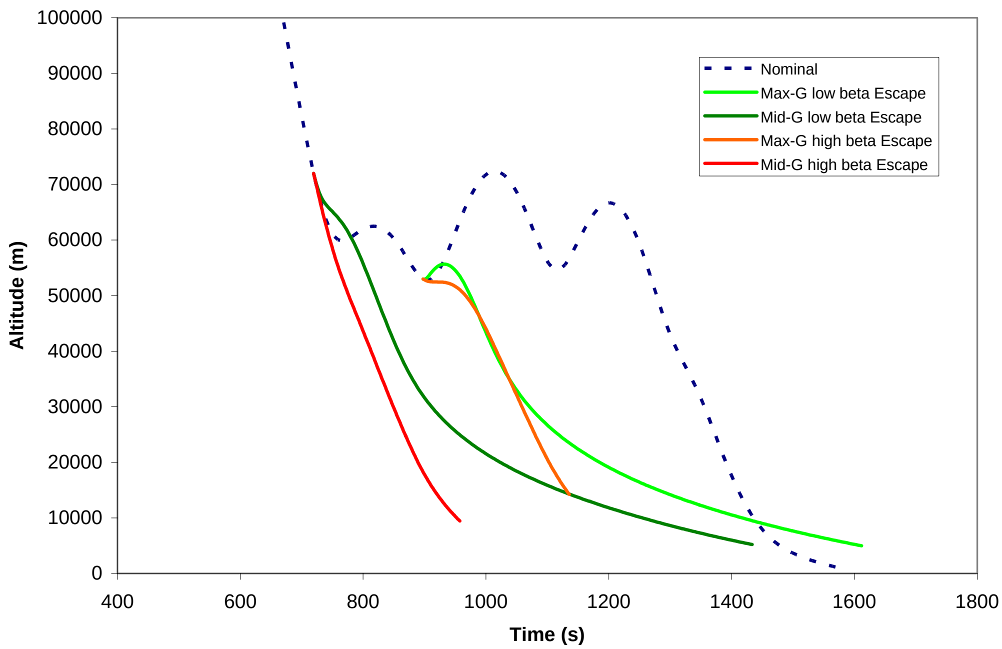

Figure 5.1 Spreadsheet Simulation Results, Altitude vs. Time

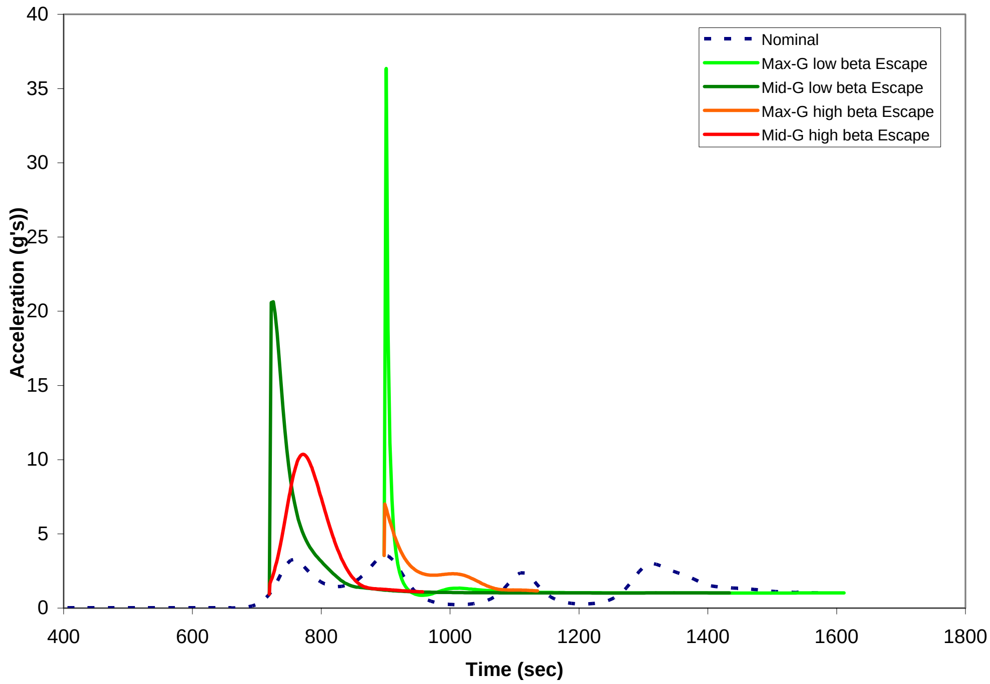

Figure 5.2 Spreadsheet Simulation Results, Acceleration vs. Time

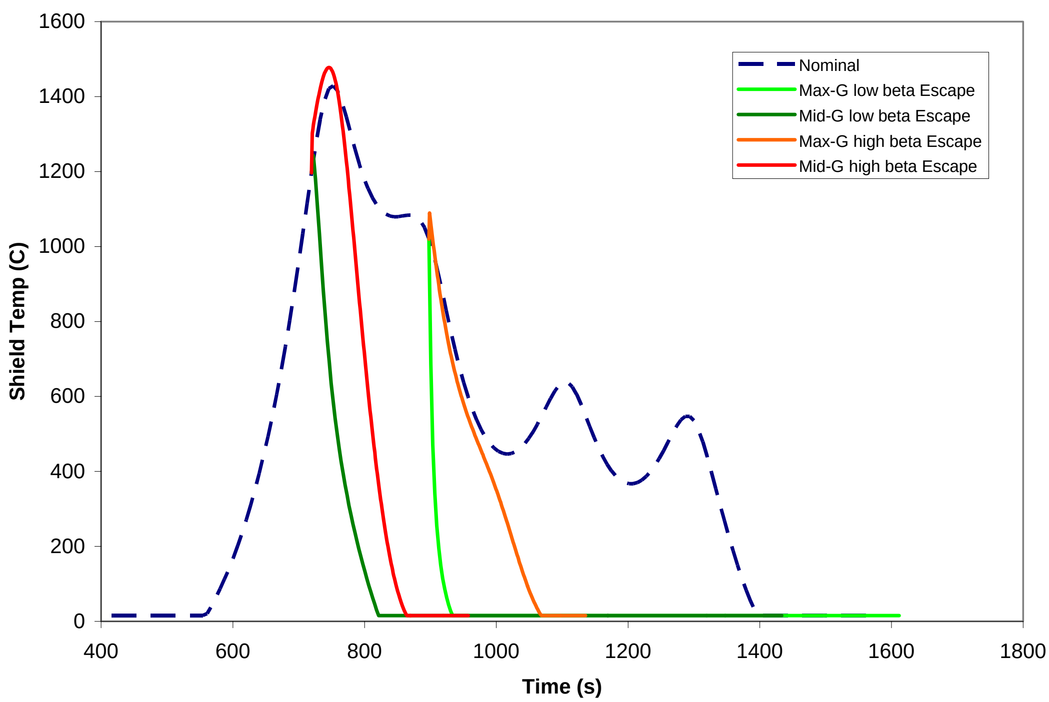

Figure 5.3 Spreadsheet Simulation Results, Outer Shield Temperature vs. Time

Figure 5.2 shows the acceleration spikes caused by ejection. In this case the low beta case is the more extreme as instantaneous accelerations peak above 35 g's. The high beta case peaks at a more manageable 10 g's. To put this in perspective, astronauts in an Apollo launch escape systems would have experienced 20 g acceleration spikes for a period of several seconds, while the survivable acceleration spike as determined by rocket sled tests is around 84 g's for 0.04 seconds[23].

Ejection Point	Low Beta Peak Acceleration (g)	Low Beta Peak Temperature (°F)	High Beta Peak Acceleration (g)	High Beta Peak Temperature (°F)
Mid-Stress Point (1 g acceleration at ejection)	20.6	2156.5	6.6	2678
High-Stress Point (3.5 g acceleration at ejection)	36.4	1867.5	10.3	1993

Table 5.2 Summary of Results of Lifeboat Check Point Tests

Several trends were apparent from the initial comparisons. First, the harshest conditions in the lifeboat trajectory were seen immediately after its ejection from the primary vehicle. The sudden change in vehicle characteristics, from large high β vehicles to smaller lower β ones, caused a sharp spike in both the acceleration and the temperatures. These spikes subside very quickly, not lasting much more then 10 seconds in the worst case before subsiding to the nominal value for the new ship's characteristics. Despite their short duration, at their peaks these spikes would most likely create conditions beyond the acceptable limits for either vessel or crew. However, the peak point in this comparison was for conditions especially chosen to be worse than could be expected in an actual mission, so only trend based conclusions should be drawn from it.

A second trend apparent in the initial study was that the closer the ballistic coefficient of the lifeboat is to that of the nominal vehicle, the gentler the transition upon ejection. Since the spike is caused by the transition between the high ballistic coefficient, lifting trajectory of the nominal vehicle to the lower ballistic coefficient, more ballistic trajectory of the lifeboat, it makes sense that the more similar those two vehicles are, the easier the transition will be.

The obvious solution to this would be to try to match the ballistic coefficients of the nominal and lifeboat vehicles. There are, however, several problems with this idea. First, while the shock accelerations may be smaller, vehicles with high ballistic coefficients and see higher temperatures than otherwise similar low ballistic coefficient vehicles. Second, it would be nearly impossible to build a functional lifeboat with even close to the ballistic coefficient of the shuttle. The shuttle has a ballistic coefficient on the order of 3000 kg/m^2. A lifeboat with a similar ballistic coefficient and a presumed mass of 1000 kg would require a heat shield with a cross sectional radius of about 0.2 meters. Assuming a crew couch width of 0.5 meters and height of 1.5 m based on the NASA standard dimensions for a 95th percentile American male [24], this is too small a diameter to allow for even a one person lifeboat, let alone a multiple crew lifeboat (figure 5.4).

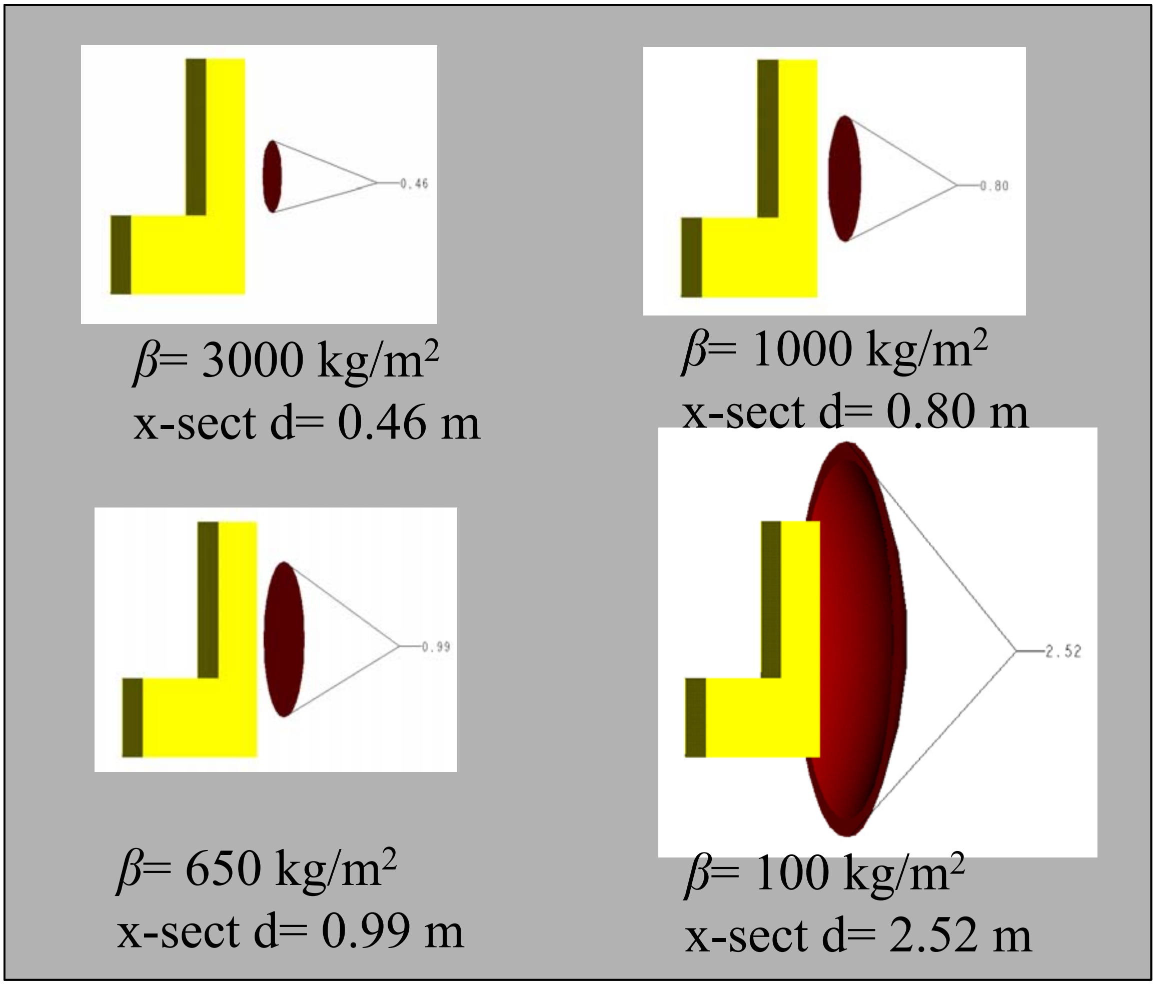

Figure 5.4 Heat Shield Sizes for 1000 kg Lifeboat with varying ballistic coefficients. Couch dimensions are 1.5 m tall by .5 m wide.

From this apparent trade off between acceleration spikes, maximum shield temperatures and dimensional feasibility, it was decided to search for a favorable escape vehicle ballistic coefficient that would find a balance between these extremes. Unfortunately the spreadsheet was not set up in a way to easily allow this to be done. Neither was it designed to do the continuous trajectory analysis that was the final goal of this project.

5.3 Coded Simulation Results

The next stage of the analysis began with the writing and testing of the coded simulation. Once checked against the previously verified spreadsheet results, it was set to step through the various lifeboat design parameters of lift to drag ratio, ballistic coefficient and heat shield radius of curvature to determine the overall optimum values for a full trajectory lifeboat. Ranges for these searches were set based on a combination of known values and assumptions. It was assumed that the lifeboat would have a ballistic coefficient smaller than that of the nominal vehicle so the approximate shuttle ballistic coefficient of 3000 was set as the maximum value in the search. It was also assumed that the lifeboat would not be a glider style vehicle, so the maximum lift to drag ratio was set as 1. Finally, the values for the radius of curvature of the heat shield was arbitrarily set between 1 meter and 5 meters with the knowledge that, if trends showed larger shields were preferred a greater range of values could be explored. For each characteristic set the simulation ran through ejections spaced throughout the trajectory. For each characteristic the worst observed metrics were recorded in a graph. The results of this search are shown in figure 5.5 and 5.6.

The graphs show that the strength of the correlation between vehicle characteristic and their effect on the trajectory varies, with ballistic coefficient having the overall strongest effect and lift to drag ratio having the overall weakest. A summary of the trends is in table 5.3. Using these results nominal characteristics for the lifeboat were set to be a ballistic coefficient of 650 kg/m^2, a lift to drag ratio of 0.2 and a heat shield radius of 2.8 m. While a larger lift to drag ratio would have resulted

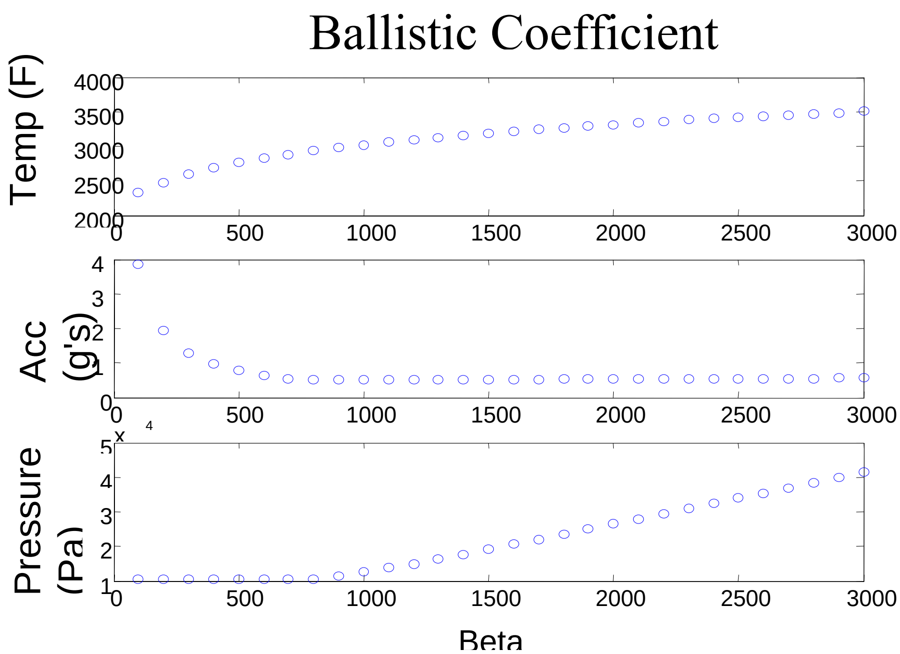

Figure 5.5 Results of Optimum Ballistic Coefficient Search

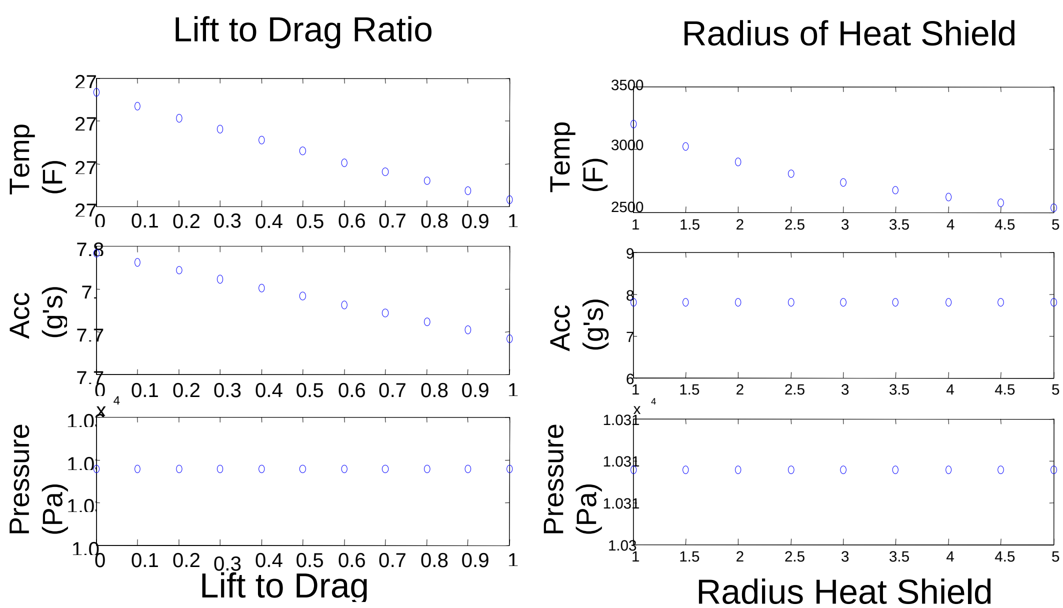

Figure 5.6 Results of Optimum Lift to Drag and Heat Shield Radius Searches

Correlation Strength	**Characteristic**	**Effect on Trajectory**
Strong	Larger β	Higher worst case temperature
	Smaller β	Higher worst case
Weak	Larger L/D	Lower worst case temperature
		Lower worst case acceleration
	Larger Heat Shield	Lower worst case temperature

Table 5.3 Correlation Strengths of Vehicle Characteristics to Experienced Conditions

in slightly lower temperatures and accelerations, the added complications to the trajectory would likely have outweighed any mitigating effects.

Once nominal values were defined, the initial conditions in the code were reset and the code was run with closely spaced ejections to obtain a high definition data set (figure 5.7). In addition the maximum allowable acceleration was set at 8 g's, assuming the use of thrust to counteract excess acceleration above this value. Since the acceleration spike is caused by the sudden change in ballistic coefficient, it can be partly counteracted by a short duration thrust in the direction of the velocity vector, slowing the rate of deceleration at the ejection point. Full results were graphed in 3-d mesh plots, while the worst point from each run, along with important details about where in the ejection trajectory it occurred, were pulled out and graphed in a second plot (figures 5.8-5.12).

While the 3-d mesh plots may be difficult to read for data from specific ejections, they provide valuable information about the trend of the reentry. The high temperature peaks in figure 5.8 are sharp, showing that the conditions tend to be of short duration with cooler temperatures throughout most of the trajectory. In addition, while many of the ejections will see maximum temperatures on the order of 1600° C, only a small portion of ejections occurring in a limited altitude and time range will see the absolute peak temperatures of 1800° C. The 3-d acceleration plots, figure 5.8, shows incidents of peak acceleration to be even more transient than the temperature peaks. Near maximum accelerations are seen only immediately after ejection and then only for short periods. Through the rest of the trajectory mesh accelerations are in the more moderate 3 to 4 g range. Coupled with figure 5.9, figure 5.10, the counter thrust mesh plot, shows how little thrust would be needed to keep accelerations below the 8 g maximum set in this project. Short bursts of thrust immediately after ejection would theoretically slow deceleration enough to limit the

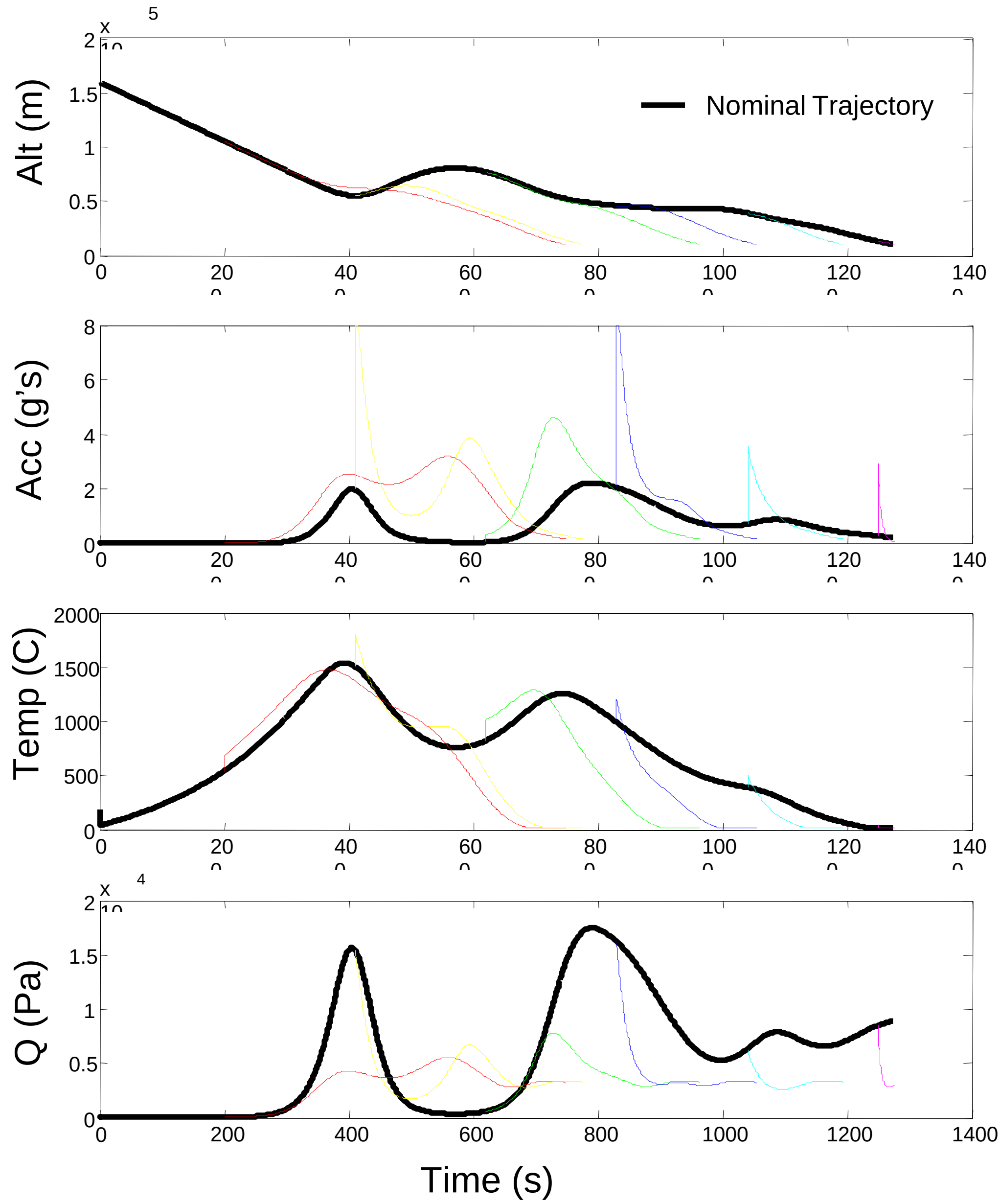

Figure 5.7 Sampling of Simulation Trajectory Results Evenly Space Over Nominal Trajectory for Optimal Lifeboat with Design Characteristics of β=650 kg/m^2, L/D = .2 and Heat Shield Radius=2.8 m.

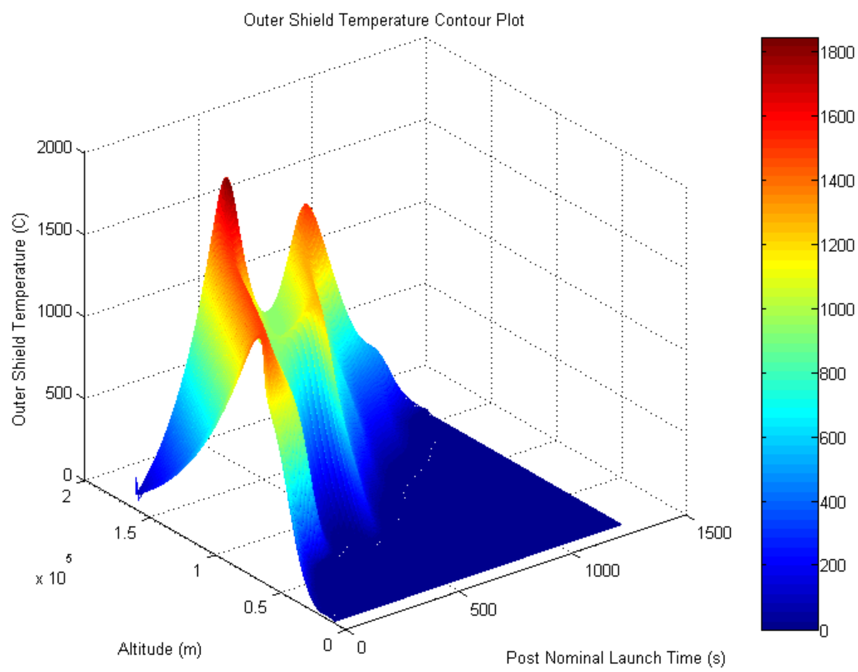

Figure 5.8 Three Dimensional Mesh Plot of Shield Temperature vs Altitude and Time

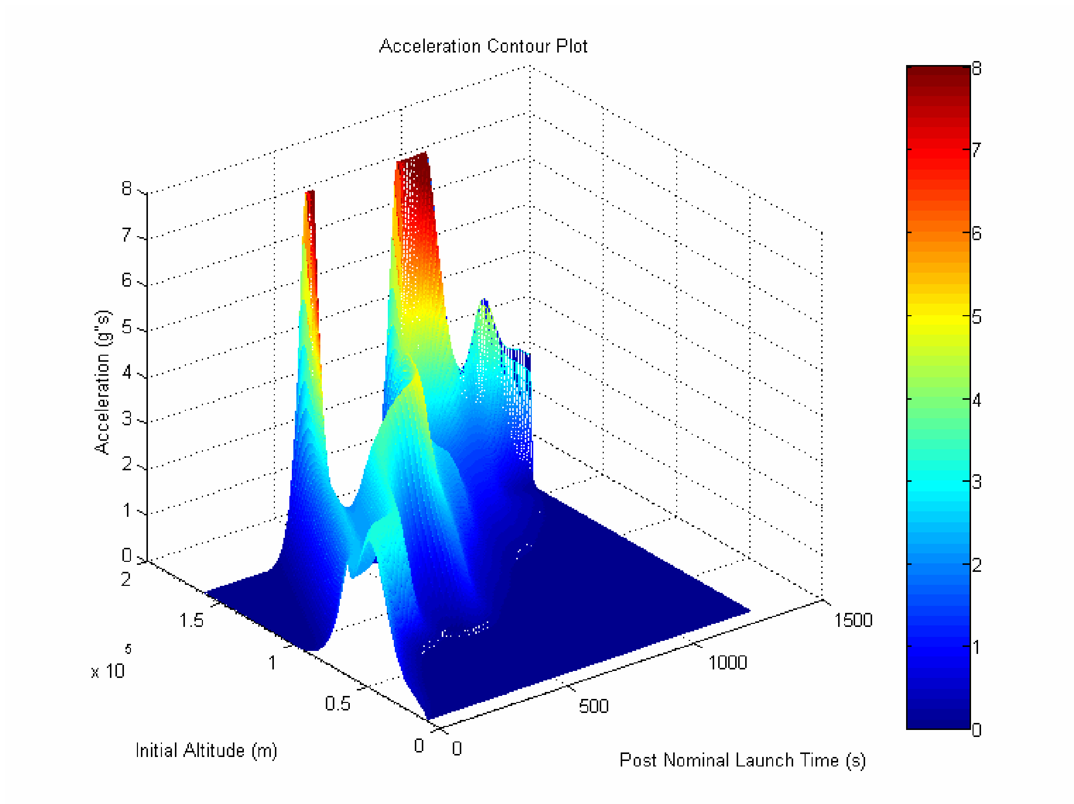

Figure 5.9 Three Dimensional Mesh Plot of Acceleration vs Altitude and Time with Maximum Accleration set to 8 g's

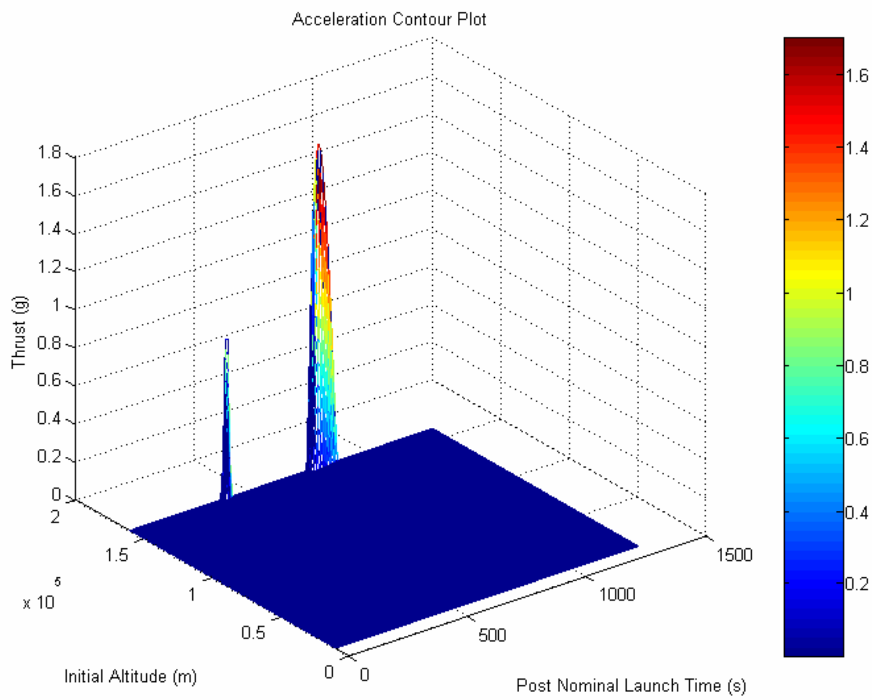

Figure 5.10 Three Dimensional Mesh Plot of Required Thrust vs Altitude and Time with Maximum Acceleration set to 8 g's

effects of the g spike on the lifeboat vehicle. Since these thrusts are only required immediately after ejection, they could be built into the attitude control system that will be required to stabilize the escape vehicle in an optimum orientation.

The worst point plots (figures 5.11-5.12) show the highest acceleration and temperature on any given ejection, along with the altitude at ejection and how long after the ejection this peak occurred. From these worst point plots the least favorable regions for ejection can be determined. These regions are seen in both the plots of temperatures and accelerations, and can be hypothesized to be the most dangerous times to attempt an ejection. In both cases, the worst points are those surrounding the nominal trajectory's period of maximum dynamic pressure (max q). In the case of acceleration the worst points tend to immediately precede max q, while the temperature plots show worst case immediately following max q. These worst point plots can be used to define the ejection zones referred to earlier in this paper.

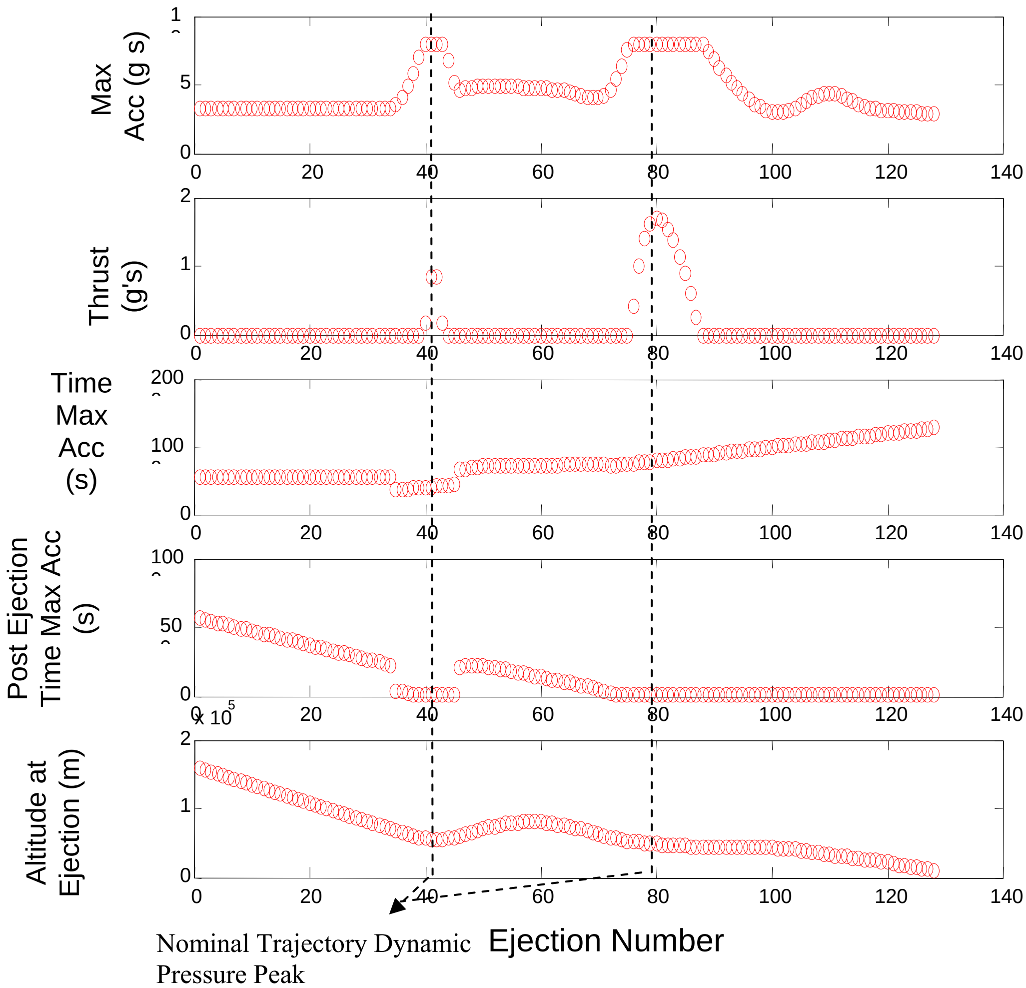

Figure 5.11 Worst Acceleration Occurrence Per Ejection Trajectory

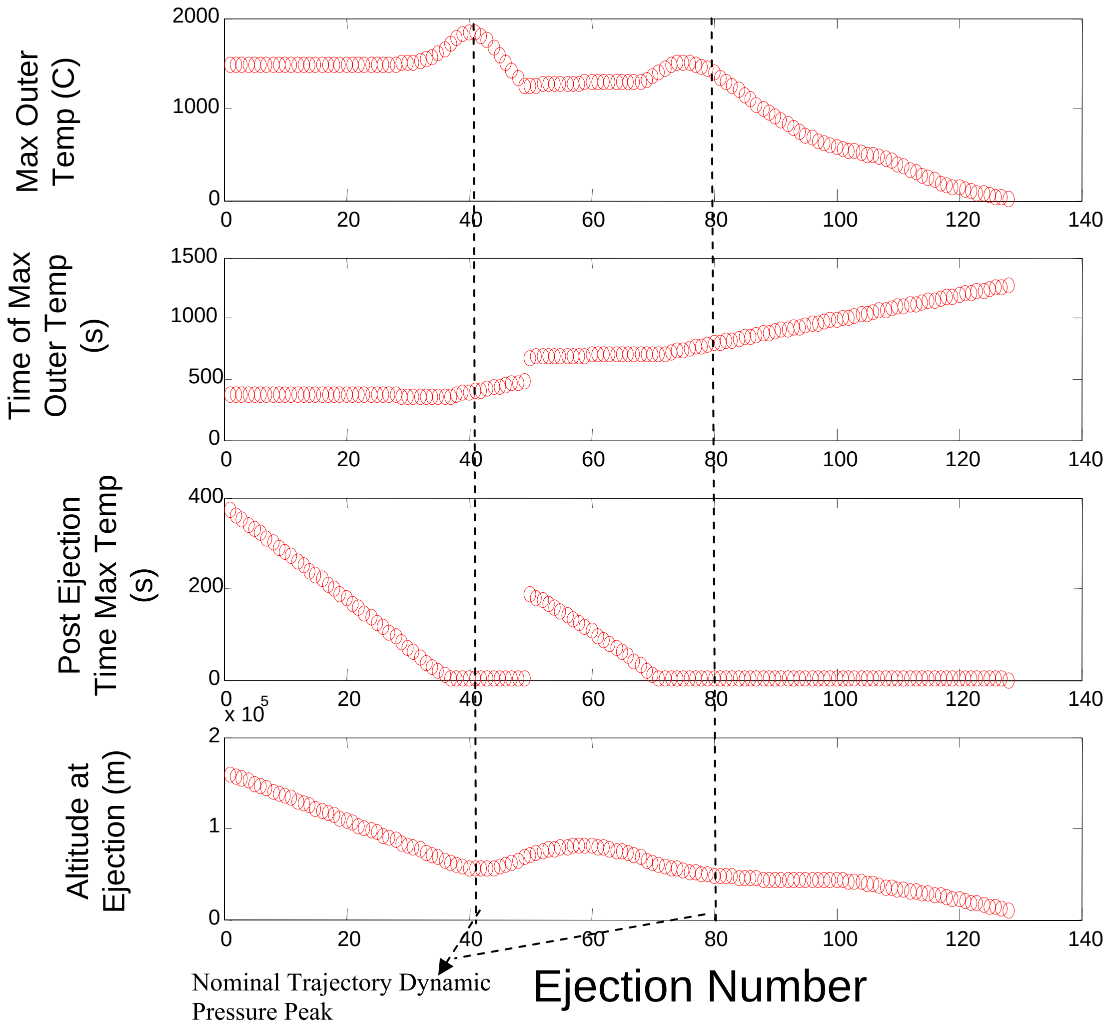

Figure 5.12 Worst Temperature Occurrence Per Ejection Trajectory

5.4 Ejection Zones

Earlier, three conceptual atmospheric zones were defined. With the results of this analysis further details can be derived for these zones. As each of these zones has very different environmental conditions lifeboats designed with these zones in mind could be split into three categories.

These zones can be mapped onto the graphs of the worst acceleration and temperature occurrences and the times at which they occurred (figures 5.13-5.14). When examining these charts, it is apparent that these three zones occur in slightly different places around the maximum dynamic pressure for the different metrics, though for planning purposes they can probably be combined into one zone set.

The worst case accelerations are found in ejections within 4 minutes following the maximum dynamic pressure. Four of the five highest accelerations are seen in this period, with the fifth being an ejection during a phugoid, which is a consequence of the estimated trajectory, and should be of a smaller magnitude or nonexistent in a true trajectory. However, the ejections where the worst case acceleration occurs sooner after ejection, which may be the harder case to plan for, occur earlier in the nominal trajectory then those where the peak temperature is experienced.

The worst case temperature scenario is more complicated. Through the first ejection zone, the maximum temperature stays flat and occurs at the same absolute time in the trajectory regardless of the actual time of ejection. These ejections occur high enough in the atmosphere that the atmospheric density is still too low to cause high heating rates. The temperature peaks, as expected, in the four minutes immediately preceding ejection at maximum dynamic pressure, however, this is also the period during which the aforementioned phugoid occurs. While in general the higher maximum temperatures occur in earlier ejections because of the longer period in flight to gather heat, these peak temperatures may be partially a factor of the phugoid. Even so, the peaks only spike about 200 °C above the temperatures seen in earlier ejections, well within the capabilities of ablative heat shields. The reality of the temperature extremes will just be used to determine the thickness of the heat shield that is required, and from there, the expected mass. The bigger problem with temperature is that, if near peak temperatures are seen immediately after ejection,

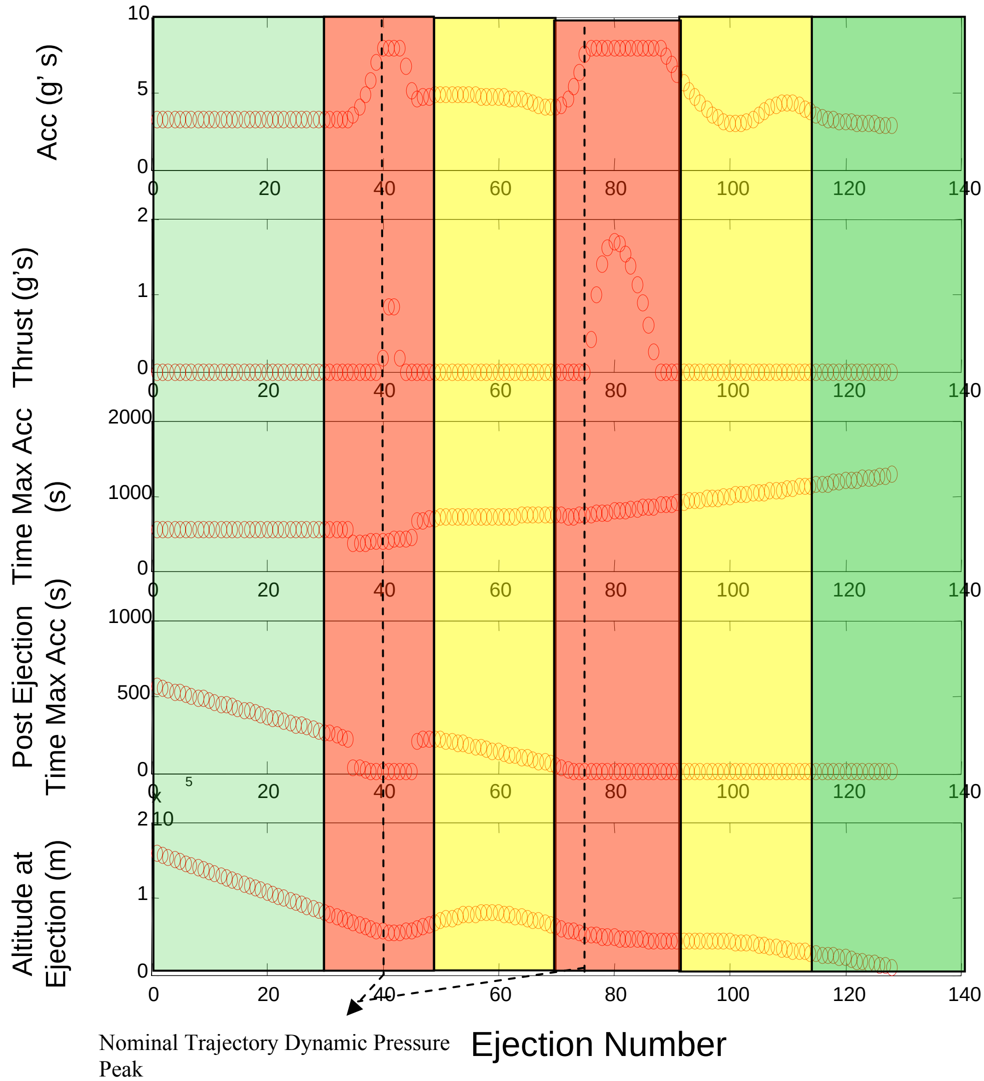

Figure 5.13 Acceleration Based Ejection Zones. Zone 1 is Green, Zone 2 is Yellow and Zone 3 is Red

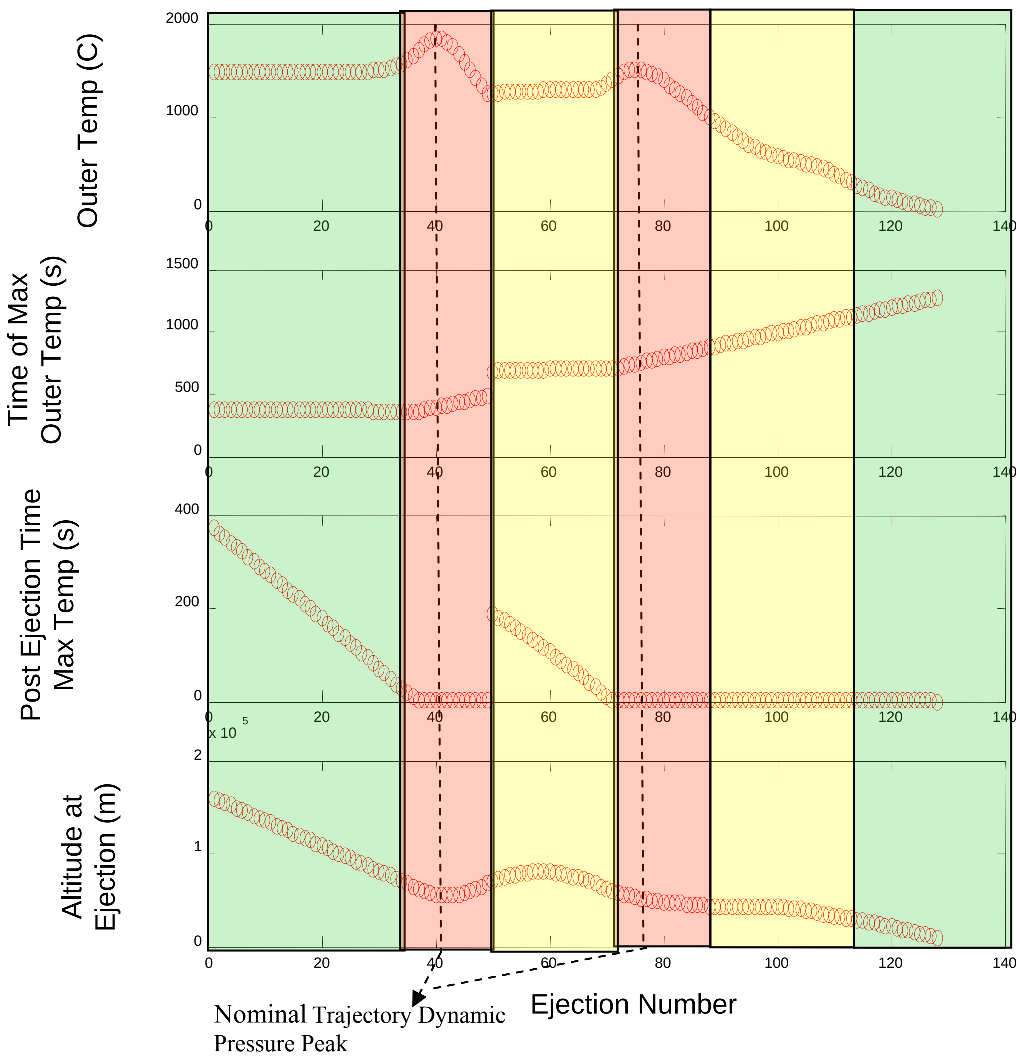

Figure 5.14 Temperature Based Ejection Zones. Zone 1 is Green, Zone 2 is Yellow and Zone 3 is Red

when the attitude of the lifeboat is at its least controlled, more shielding will be needed over the whole of the vehicle. If the vehicle attitude can be controlled before the temperature extremes are reached, only shielding in the velocity direction will be required.

Metric	Maximum	Ejection Time (s)	Time (s)
Acceleration (no counter thrust)	9.69 g's 8.83 g's	790 410	791 411
Temperature	1839 °C 1513 °C	390 740	391 741
Dynamic Pressure in Nominal Trajectory	17498 Pa 15670 Pa	NA NA	791 405

Table 5.5 Harshest Conditions and Time of Occurrence in Optimum Lifeboat

As was mentioned previously, the overall harshest conditions in a given set of trajectories are experienced immediately after the ejection from the nominal vehicle around the point of maximum dynamic pressure. This is true for any ballistic coefficient lifeboat, though the intensity of this peak depends on the ballistic coefficients of both the nominal vehicle and the lifeboat. The sudden change in vehicle characteristics at an already stressful moment causes sudden spikes in both sensed acceleration and shield temperature. However these spikes subside relatively quickly, within a matter of seconds.

Zone 1 was defined as having orbital or near orbital conditions. This means the atmospheric density is still very low, and acceleration and temperature effects are still minimal. A type 1 lifeboat would be designed with these specific conditions in mind. This would be a limited escape lifeboat, and by far the easiest type to design.

Since the only extreme conditions that would be seen are defined by the characteristics of its design rather than the environment it ejects into, this vehicle could be made with minimal structure and shielding. It would also need minimal attitude control as ejection would not occur immediately into extreme conditions and time could be taken to orient it correctly. This type of vehicle would be comparable to the MOOSE escape concept of the 1960's.

Zone 2 was defined as the region on either side of the maximum dynamic pressure of the nominal trajectory. A lifeboat designed for this zone would have to be sturdier with more structure and shielding, as some ejection extremes would be experienced. Attitude control in this vehicle would also be more critical as there would be little or no time to orient before the environmental extremes are experienced, whether these come from the ejection environment or from the lifeboat's own trajectory. However any lifeboat that could survive a zone 2 ejection would also be fully capable of surviving a zone 1 ejection.

Zone 3 lifeboats would be the most challenging problem. Zone 3 was defined as the region directly around the nominal trajectory's max q, where all environmental extremes are experienced. Due to the shock of ejection, a lifeboat ejecting in this zone would be immediately dealing with a more extreme environment then either the nominal or lifeboat vehicles would otherwise see. In this type of vehicle it will be critical to be able to immediately point the vehicle in the optimum reentry orientation. Not only will it be difficult and prohibitively mass intensive to provide sufficient structural support and heat shielding for the entire lifeboat, but the g shock will be sufficient that if the crew is not correctly oriented, they would most likely sustain serious injuries.

In this chapter, the overall optimum characteristics of the lifeboat were defined. While the characteristics arrived at here, a ballistic coefficient of 650 kg/m^2, a lift to drag ratio of 0.2, and a heat shield radius of 2.8 m, are optimum for the mission, a different set of characteristics would be optimum in each zone, with the defined characteristics a compromise between them.

Chapter 6: Discussion and Conclusions

6.1 General Findings

The purpose of this thesis was to determine whether emergency escape was feasible throughout the duration of a space mission. The data reported in the chapter above seems to support that a mission wide ejection scenario could be immediately fatal in a sufficiently designed lifeboat. Assuming the lifeboat can be quickly pointed in the correct orientation, the acceleration and temperature regimes it sees are in no case so intense to be beyond the limits of survivability. Thus the limits on the ability to build such a craft will be due to mass, cost, or the limits of current technology, not mechanical limits.

What is not necessarily clear is whether or not this scenario is required. Of the 11 missions studied in Chapter 3 (table 6.1), six had anomalies that caused or could have caused problems to the entry process. Of those five, three of the missions, Mercury MA-6, Gemini 8 and Soyuz 1, first experienced problems while the vehicle was still in orbit. In two other missions, Soyuz 5 and Soyuz 11, the problems began with the retro burn to initiate the reentry process. Only the Columbia incident occurred in the atmosphere with no prior warning. While this is a small data pool from which to draw conclusions, it could be argued that lifting vehicles are generally more vulnerable to suddenly occurring fatal errors during the later stages of reentry. From historical data it appears that, in the case of capsule-style vehicles, anomalies which lead to reentry problems are usually apparent before reentry actually begins. While it may be possible to design a vehicle that can eject at any point, it may not be worth the mass and cost to do so for this class of vehicle. If a sufficient percentage of anomalies can be spotted from orbit it may only be necessary or desirable to build a zone 1 lifeboat. This prospect is much less expensive in both mass and cost and could probably be fairly easily incorporated into new vehicle designs.

6.2 Future Work

6.2.1 Trade Studies on Lifeboat Styles

6.2.1.1 Limited Ejection Periods

If, as is conjectured above, it is not efficient to design a lifeboat for the full spectrum of entry ejections, then the optimums presented earlier in this paper are not applicable. The optimum values found in the search performed in this analysis are the optimums as represented by ejections through the whole nominal entry trajectory, including the worst case points which will drive the escape system design. Should the full trajectory approach not be used a different set of optimum values will be required, based on the portions of the nominal trajectory considered suitable for ejection.

Mission	Year	Incident	Lifeboat Method	Enacted
Mercury MA-6	1962	Signal indicating prematurely detached heat shield	Type 1	No
Gemini 6	1965	Premature Booster Shutdown	LES	Yes
Gemini 8	1966	ACS Failure	Type 1	No
Soyuz 1	1967	ACS Failure	Type 1 or Type 3	No
Soyuz 5	1969	Service Module Failed to Separate	Type 1	No
Soyuz 11	1971	Sudden Decompression	Type 1	No
Soyuz 18	1975	Faulty Booster	ATAL	Yes
Soyuz T-10A	1983	Booster Fire	LES	Yes
STS-51-F	1985	Engine Failure	ABO	Yes
STS-51-L	1986	Booster Explosion	LES	No
STS-107	2003	Heat Shield Failure	Type 3	No

Table 6.1 Summary of Historical Space Safety Incidents and Possible Mitigation Methods

Based on trends observed in this analysis, if the period around the nominal maximum dynamic pressure is excluded, there will be several beneficial effects on the lifeboat vehicle design. First, the optimum ballistic coefficient will most likely drop significantly. This will allow for designs with larger heat shields on lighter vehicles, a problem that, as demonstrated by figure 5.4, will currently make vehicle design fairly difficult. Secondly, excluding the period of the trajectory around max q will minimize both the g-shock and temperature shock that is seen in the first seconds after ejection, allowing for less required structure and shielding.

Further, limiting the valid ejection period to the segment of the trajectory described above as zone 1 would allow for still easier lifeboat design. In this case there is no g-spike upon ejection, and the characteristics of the entry trajectory are much more under the control of the astronaut or mission control, allowing for better control of the landing zone for the lifeboat. The controllable entry conditions would also allow for even less structure and shielding, leading to a still lighter vehicle. While the historical findings presented above seem to indicate that a zone 1 lifeboat is the most useful, a full trade study should be performed, first to find the optimum design characteristics for lifeboats designed for the various ejection zones, and second to analyze the mass and design savings versus the possible loss of life by limiting the ejection options.

6.2.1.2 Size and Occupancy of Ejection Capsule

A second issue that should be looked at in a trade study is the size of the lifeboat. Should each member of the crew be strapped into separately ejecting lifeboats or is some form of ejecting cabin for the entire crew preferable. There are two main unknowns in this question. First, is whether or not separate ejection capsules actually present any safety benefit over a full crew capsule. Second, the size of the mass penalty for separate capsules needs to be determined as well as advantages and disadvantages of both designs. Inherent in this study should be mass and cost estimates for the various styles of lifeboats, as this will be the main driving force behind the design of actual vehicles.

6.2.2 Methods for Minimizing G-Shock

Should a full mission lifeboat be designed after all, the most pressing issue in the design will be the sudden g-shock that occurs when the ship characteristics suddenly change. While the temperature issues can be dealt with given sufficient shielding, the crew and vehicle structure have a limited acceptable acceleration range. Analysis has shown that one way to reduce this shock is to match the ballistic coefficients of the lifeboat with that of the nominal vehicle as closely as possible. However, when dealing with high ballistic coefficient nominal vehicles, there is a limit to how closely that can be done before the lifeboat vehicle would become unworkable. As was shown in figure 5.4, in order to maintain a high enough ballistic coefficient to avoid a tremendous g-shock the size of the cross sectional area of the vehicle, and hence the size of the heat shield, has to be kept unworkably small. If a method could be created to alleviate the g-shock while lowering the ballistic coefficient, this would provide several benefits.

The main benefit, as was mentioned above, is that it would allow the heat shield diameter to be raised to a workable size. Though this analysis has shown that it is theoretically possible to design a lifeboat that could be used mission wide, the limitations placed on it will also make it very hard to build. The ballistic coefficient requirement will require it to either be too heavy or too small to be very feasible. Finding a method to alleviate the g-shock until the lifeboat is set into the new trajectory would solve this problem.

Below several concepts for reducing the g-shock are discussed. While none have been analytically proven, all are theoretically possible with available technology. Future work may determine which of these is truly feasible and worthy of more study.

6.2.2.1 Deployable Heat Shield

One way to limit both acceleration and temperature might be to use a deployable heat shield. In this design the lifeboat would have a minimal heat shield for the initial ejection. After ejection, additional heat shield area would deploy, decreasing the ballistic coefficient in a controlled manner. This would allow for a high ballistic coefficient at ejection when that is preferable, with a decreasing ballistic coefficient after the initial ejection which will limit the maximum temperature in the later stages of entry.

Deployable heat shields have at least some flight history. Various designs have been used since the 1960's, mostly in smaller satellites and flight tests. Such a heat shield is both space and mass efficient as only the stowed space needs to be accounted for. In some cases this form of heat shield can also act as a parachute decreasing even further the required lifeboat mass [25].

One main question with this style design are the details of shield deployment under ejection conditions. Deployable heat shields that have already flown deploy above the atmosphere, in the region referred to here as zone 1. It is unclear whether such a shield could be designed to deploy in zone 2 or zone 3 conditions. This style heat shield is also most common on small payloads. Another issue to look into is whether the designs for deployable heat shields can support the mass requirements of a lifeboat. Most of the current designs have only flown very small payloads. More testing and development work will need to be done before this concept can be considered realistic.

6.2.2.2 Counter Thrust

Another conceptual way to alleviate the g-shock might be the previously mentioned idea of using a short counter thrust to decrease the rate of deceleration in the first few seconds after ejection. A variation of this has been used in the Soviet Soyuz program where hydrogen peroxide jets fire during the reentry reducing thermal and gravity loads [14]. Thrusting against the deceleration vector, thus preventing it from decelerating too quickly, might be the most efficient way to prevent a large g-shock. Since any vehicle design is going to require some amount of attitude control, the mechanisms for this method will most likely already be included in the design. The only additional mass requirement in this case would then be a larger fuel supply and possibly slightly larger engines. If this method proves viable it might be the most efficient way to limit the conditions experienced by the lifeboat.

6.2.2.3 Added Nominal Mass

A passive method to increase the size of the heat shield while holding the ballistic coefficient steady might be to store more of the primary vehicle systems inside the lifeboat frame. The main constraint on increasing the mass of the lifeboat is that a minimal amount of excess launch mass is required. By placing nominal systems inside the lifeboat the nominal launch mass is not increased, but the lifeboat mass is. A problem with this concept is that it will require a larger lifeboat vehicle in order to provide room to stow these systems.

6.3 Conclusions

Space travel, like all explorations of frontiers throughout history, is dangerous. No lifeboat or abort method is going to be able to avoid all accidents or injuries. If human space exploration is going to continue, the best that can be hoped for is to learn from past mistakes and to make plans for as many imagined contingencies as possible. This thesis has hopefully open up one more contingency option in future spacecraft designs. In the Crew Exploration Vehicle currently being designed requirements will likely call for at least a launch escape system. This thesis suggests that, at the very least, an orbital escape system or type 1 lifeboat would also be a valuable addition. In addition, if such an orbital escape system is designed, the same principles could perhaps be used to create emergency escape vehicles for the International Space Station. Such a system would greatly enhance the functionality of the station, which is currently limited by the escape capabilities of the attached Soyuz capsule.

Appendix A: MATLAB Simulation Code

EntryMap Readme

This code is set to calculate the trajectories of a lifeboat escaping at set time periods from a nominal reentry trajectory. The code can be run from the EntryMap.m file. The nominal vehicle characteristics and general program constants such as the timestep, earth radius, etc, are in Get_Start_Constants.m. The lifeboat vehicle characteristics are in Get_Constants.m. All program constants are set in one of these two files. Aerocalcs.m is the function responsible for calculating the trajectory characteristics for each timestep. Equations inside this function are documented with their sources.

```
function EntryMap
%March 19, 2004*********************************
%function EntryMap.m
%Uses Functions********************************
%   Get_Constants
%   Get_Start_Constants
%   doitloop
%   statsloop

   clear;
   close all;

%Array Columns *****************************
   global Roll Time Flt_angle Alt DynP Downrange Crossrange Vel Sensed_Acc
Mach
   global Heat_Rate Shield_Tempo Shield_Tempi Tot_Heat Density Yaw Cp Lgth

%Constants**************************
   global mu Re deltaT LoD Rhs gravity d2r
```

```
   global Gamma Velocity Height Rt
   global sh_iters oldtemps
   global increment addthrust depthinc runstats
   [data]=Get_Start_Constants;
   width=Lgth; %Lght is always the last column
%***************************************************
   addthrust=0;
   [starttraj counter]=doitloop(data, oldtemps);
   Get_Constants;
   depth=0;
   conditions(1,1,1)=0;
   for loop=1:depthinc:counter
      depth=depth+1
      [temp1 counter1]=doitloop(starttraj(loop,:), oldtemps);
      temp1(1,Lgth)=counter1;
      [conditions(1:counter1,1:width, depth)]=temp1;
      %[conditions(depth,1:length,1:width)]=temp1;
   end %loop
   if runstats==1
      temp=statsloop(starttraj,conditions);
   end %if statement
```

```
function Get_Constants
%Last Update 3/23/04
%Function Assigns Global Constants for escape vehicle
%Calls:
%   Get_Density
%Called By:
%  EntryMap

%Globals (Constants)****************************
global LoD beta Rhs rolleqtn addthrust

LoD=.2;
beta=650;
%beta=1000;
Rhs=2.808;          %meters
rolleqtn='0';       %character string Equation to be evaluated for Roll values
%thrust=500;          %thrust to be added to first time step of each escape
%end %Get_Constants
%************************************************
%Finite Difference Heating
global alpha matthick oldtemps layers densv densc ablate
ablate=0;
%kappa=.18;
matthick=.05;
```

```
delx=.001;
layers=matthick/delx;
%cv=1087.84; %J/Kg-K
[kappa temp cv temp2]=Get_Kappas(282.4);
oldtemps=0;
densv=13/(((.3048)^3)*2.205);  %Hyperlight virgin density converted to Kg/m^3
densc=6.27/(((.3048)^3)*2.205);  %Hyperlight charred density converted to Kg/m^3
%dens=224.25;        %Flexible Min-K numbers converted to Kg/m^3
dens=densv;
alpha=kappa/(dens*cv);
%end Get_Constants
```

```
function [data]=Get_Start_Constants
%
%function data=Get_Start_Constants
%Last Update 3/23/04
%Function Assigns Global Constants
%Calls:
%   Get_Density
%Called By:
%   EntryMap
%Assigns Globals and initial conditions for original entry vehicle.

%Array Columns (Globals) ****************************
global Roll Time Flt_angle Alt DynP Downrange Crossrange Vel Sensed_Acc Mach
global Heat_Rate Shield_Tempo Shield_Tempi Tot_Heat Density Yaw Cp Thrst Lgth
global Gamma Velocity Height Rt

   Roll=1;
   Time=Roll+1;
   Flt_angle=Time+1;
   Alt=Flt_angle+1;;
   DynP=Alt+1;
   Downrange=DynP+1;
   Crossrange=Downrange+1;
   Vel=Crossrange+1;
   Sensed_Acc=Vel+1;
   Mach=Sensed_Acc+1;
   Heat_Rate=Mach+1;
   Shield_Tempo=Heat_Rate+1;
   Shield_Tempi=Shield_Tempo+1;
   Tot_Heat=Shield_Tempi+1;
   Density=Tot_Heat+1;
   Yaw=Density+1;
   Cp=Yaw+1;
   Thrst=Cp+1;
```

```
   Lgth=Thrst+1;

   Gamma=1;
   Velocity=2;
   Height=3;
   Rt=4;
%Globals (Constants)****************************
global mu Re deltaT LoD beta Rhs gravity d2r increment rolleqtn Rollcount depthinc
max_acc

gravity=9.80665;    %m/s^2
mu=3.986004e14;     %m^3/s^2
Re=6378000;         %radius of Earth (m)
deltaT=.5;          %seconds
LoD=1;
beta=3000;
max_acc=8;
Rhs=10;          %meters
d2r=pi/180;     %degree to radian conversion
increment=10;   %time increment between results printed to screen in stats2
depthinc=20;%******************                %Fineness of Grid: < depthinc -> finer
grid
%rolleqtn='90*(sin((time)/tou+(pi)))^1';     %Equation for Roll Calculation called in
GetRoll
rolleqtn='90*(sin(time/tou))^1';     %Equation for Roll Calculation called in GetRoll
%rolleqtn='0';     %Equation for Roll Calculation called in Get Roll
Rollcount=1;
%***Initial Conditions**************
counter=1;
data(counter,Roll)=0;
data(counter,Time)=0;
%data(counter,Flt_angle)=0;
data(counter,Flt_angle)=-2;
%data(counter,Alt)=79000;
data(counter,Alt)=160000;
data(counter,DynP)=0;
%data(counter,Vel)=6859.3;
data(counter,Vel)=(mu/(data(Alt)+Re))^.5;
data(counter,Downrange)=0;
data(counter,Crossrange)=0;
data(counter, Sensed_Acc)=0;
data(counter,Mach)=22.46;
data(counter, Heat_Rate)=1.58;
data(counter,Shield_Tempo)=192.6
data(counter,Shield_Tempi)=20
data(counter,Tot_Heat)=0;
```

```
data(counter,Density)=Get_Density(data(counter, Alt));
data(counter,Yaw)=0;
data(counter,Cp)=1.84;
data(counter,Thrst)=0;

%data(counter,Drag)=0;

%************************************************
%Finite Difference Heating
global alpha matthick oldtemps layers ablate
ablate=0;
kappa=.18;
matthick=.5;
cv=1087.84; %J/Kg-K
oldtemps=0;
dens=224.25;      %Flexible Min-K numbers converted to Kg/m^3
alpha=kappa/(dens*cv);
layers=4;
%************************************************
global runstats plotit
global foldpath filename filename1 filename2 statefile savefiles
global ndens ndrag
runstats=1;
savefiles=1;
plotit=1;
parent=eval('cd');
newfold='testfiles';
foldpath=sprintf('%s\\%s',parent, newfold);
dirstring=sprintf('mkdir %s',newfold);
eval(dirstring);
%Trouble causing first letters \n,\r,\t,\b,\f
%In case of one of above use \\ not \
filename=sprintf('%s\\workingtraj.txt',foldpath);
filename1=sprintf('%s\\starttraj',foldpath);
filename2=sprintf('%s\\entrytraj',foldpath);


%end Get_Start_Constants
```

```
function [data, counter]=doitloop(aero, oldtemps)
%March 19, 2004*********************************
%doitloop.m
%Called by
%   EntryMap
%Uses Functions********************************
```

```
%   Get_Constants
%   aerocalcs
%   stats

%Array Columns ****************************
   global Roll Time Flt_angle Alt DynP Downrange Crossrange Vel Sensed_Acc
Mach
   global Heat_Rate Shield_Tempo Shield_Tempi Tot_Heat Density Yaw Cp Thrst
Lgth

%Constants**************************
   global mu Re deltaT LoD Rhs gravity d2r max_acc
   global Gamma Velocity Height Rt
   global increment
   data=aero;
   counter=1;
%Find Initial State**********************************
   height=Re+data(Alt);
   state(Gamma)=data(Flt_angle)*d2r;
   state(Velocity)=data(Vel);
   state(Height)=data(Alt);
   state(Rt)=0;
   chardepth=0;
%*************************************
   while data(counter,Alt) > 10000
      last=counter;
      counter=counter+1; %*******************
      [data(counter,:) state(counter,:) oldtemps]=aerocalcs(data(last,:), state(last,:),
oldtemps, chardepth);
      toscreen=0;
      if toscreen==1
         disp(sprintf('Time= %d seconds',data(counter,Time)));
         disp(sprintf('Altitude= %d meters',data(counter,Alt)));
         disp(sprintf('Velocity=%d m/s^2',data(counter,Vel)));
         disp(' ');
      end %toscreen
   end %while statement
   counter
%end %doitloop
```

```
function [aero, laststate, oldtemps, chardepth]=aerocalcs(data, state, oldtemps,
chardepth)
%Last Update 4/07/04
%Functions Called By:
%   doit
%Calls Functions
```

```
%   Get_Density(alt)
%   accelfunct(t, state)
%   getMach
%   Get_Cp0
%Global Constants ****************************
global Roll Time Flt_angle Alt DynP Downrange Crossrange Vel Sensed_Acc Mach
global Heat_Rate Shield_Tempo Shield_Tempi Tot_Heat Density Yaw Thrst Cp
global mu Re deltaT LoD beta Rhs gravity d2r max_acc
global Gamma Velocity Height Rt
global ablate
%Get_Constants;

%Temp vars (for debugging)***********
debug=0;
%**********************************
%End Temp vars***********************

%************************
   %betavar=3.7*beta/data(Cp);
   global Rollvar
   fltvar=data(Flt_angle)*d2r;
   Rollvar=data(Roll)*d2r;
   Yawvar=data(Yaw)*d2r;
   height=Re+data(Alt);
   %Since Altitude was defined in the matrix as height above earth, height
   %is being used here as distance from center of earth.

%************************
   tspan=deltaT;
%Integrate to Find Velocity********
   orb_opt = odeset('RelTol',1e-8,'AbsTol',[1e-8 1e-8 1e-8 1e-8]);
   [T,newstate]=ODE45('accelfunct', tspan, state, orb_opt);
   last=size(newstate(:,1));
   last=last(1);
   laststate=newstate(last,:);
%Increment travelled Distance*******
   aero(Time)=data(Time)+deltaT;
   aero(Vel)=laststate(Velocity);
   aero(Alt)=laststate(Height);                              %altitude calculation
(Spreadsheet D)
   aero(Roll)=GetRoll(data(Roll),aero(Time));
   Rollvar=aero(Roll)*d2r;
   aero(Density)=Get_Density(aero(Alt));                          %Air Density
(Spreadsheet N)
   aero(Flt_angle)=laststate(Gamma)/d2r;
%********************
```

```
   if debug==1
      disp(sprintf('Flt Path Angle (IJK) = %d degrees',aero(Flt_angle)));
   end
%*********************
   %aero(Drag)=norm(Get_Drag(laststate(Velocity), aero(Density)));

aero(Downrange)=data(Downrange)+deltaT*data(Vel)*cos(Yawvar)*cos(fltvar)/100
0;  %Downrange Distance Calculation (Spreadsheet F)
   aero(Crossrange)=data(Crossrange)+data(Vel)*sin(Yawvar)*deltaT/1000;
%Crossrange Distance Calculation (Spreadsheet G)
   aero(Mach)=getMach(aero(Vel),aero(Alt));                              %Mach
Number (Spreadsheet J)
   aero(Cp)= Get_CP0(aero(Mach));                                        %Specific Heat
(Spreadsheet P)
   aero(DynP)=aero(Density)*aero(Vel)^2/2;                               %Dynamic
Pressure Calculation (Spreadsheet E)
   %Heating Rate Laminar Flow on a Flat Plate, notes L&E 3/4 pg 5
   %cos(phi)=1 sin(phi)=0
   aero(Heat_Rate)=2*(2.568e-9)*11.3538*((aero(Density)/Rhs)^.5)*aero(Vel)^3.2;
%kW/m^2 Heat Rate (Spreadsheet K)
   %Ho=aero(Cp)*((-69-32)*5/9)+(aero(Vel)^2)/2
   %Haw=aero(Cp)*((-69-32)*5/9)+.845*(aero(Vel)^2)/2
   %m=3;    n=.5;   C=1.83e-8*Rhs^(-.5); %Conservative C, assuming Hw/Ho ~=0
   %aero(Heat_Rate)=(aero(Density)^n*(aero(Vel))^m*C)*100^2/1000;      %Heat
Rate in kW/m^2
   if ablate==0
      [aero(Shield_Tempo) aero(Shield_Tempi) oldtemps]=
getShield5(aero(Heat_Rate), oldtemps);      %Shield Temperature (Spreadsheet L)
   else
      [aero(Shield_Tempo) aero(Shield_Tempi) oldtemps chardepth]=
getShield_ablate(aero(Heat_Rate), oldtemps, chardepth);       %Shield Temperature
(Spreadsheet L)
   end
   %pause %**********************
   aero(Tot_Heat)= data(Tot_Heat)+aero(Heat_Rate)*deltaT;                  %Total
Heat (Spreadsheet M)
   aero(Yaw)=
data(Yaw)+data(Density)*gravity*data(Vel)*LoD*sin(Rollvar)*deltaT/(2*beta*d2r);
%Yaw Angle (Spreadsheet O)
   temp1=beta/aero(Cp);
   denstemp=gravity*aero(Density);
   drag=aero(Density)*state(Velocity)^2/(beta/aero(Cp));
   gtemp=GetGravity(laststate(Height));

%aero(Sensed_Acc)=sqrt((denstemp*aero(Vel)^2/(temp1))^2+(denstemp*LoD*cos(
Rollvar)*aero(Vel)^2/(temp1))^2)/gravity;
```

```
   %aero(Sensed_Acc)=sqrt((gravity*drag)^2+(gravity*drag*LoD)^2)/gravity;
   aero(Sensed_Acc)=(abs(-drag-gtemp*sin(laststate(Gamma)))/gtemp);
   if aero(Sensed_Acc)>max_acc
      Cd=2;
      mass=(pi*Rhs^2*Cd/beta);
      aero(Thrst)=(aero(Sensed_Acc)-max_acc)*mass*gravity;    %thrust in Newtons
      aero(Sensed_Acc)=max_acc;
   else
      aero(Thrst)=0;
   end %if > max_acc
%Display for Debug*****************************************
if debug==1
   disp(sprintf('Time = %d seconds', aero(Time)));
   disp(sprintf('Alt = %d meters', aero(Alt)));
   disp(sprintf('Vel = %d m/s', aero(Vel)));
   disp(sprintf('Acceleration = %d m/s^2', aero(Sensed_Acc)));
   disp(sprintf('Flt Path Angle = %d degrees',aero(Flt_angle)));
end %debug
   %end %aerocalcs
```

```
function stats=statsloop(starttraj,data)
%March 19, 2003************************************
%Uses Functions*********************************
%
%Called By*********************************
%   EntryMap
%file variables******************************
global plotit
global foldpath filename1 filename2 statefile savefiles
global mu Re deltaT LoD beta Rhs gravity d2r increment

   start=1;
   finish=2;
   %delimiter=',';
   delimiter='\t';

%Array Columns ****************************
global Roll Time Flt_angle Alt DynP Downrange Crossrange Vel Sensed_Acc Mach
global Heat_Rate Shield_Tempo Shield_Tempi Tot_Heat Density Yaw Cp Thrst Lgth
global Gamma Velocity Height Rt

   maxtemp=1500;
   loop=1;
   temp=0;
```

```
   tottime=0;


   number=1;
   [length width depth]=size(data);


   timecount=0;
   oldtime=-increment;
   startcount=0;
   changelast=0;
%Write to Files**********************************
   tempfile=sprintf('%s.txt',filename1);
   if savefiles ~=0
      dlmwrite(tempfile,starttraj,delimiter);
      %dlmwrite(filename2,data,delimiter);
   end

   for loop2 =(1:depth)
       length=data(1,Lgth,loop2);
       tempfile=sprintf('%s%d.txt',filename2,loop2);
       if savefiles ~=0
         %dlmwrite(tempfile,data(loop2,:,:),delimiter);
         dlmwrite(tempfile,data(1:length,:,loop2),delimiter);
      end
   %**********Find Nominal Max Q*************************
      [nommaxq nommaxqindex]=max(starttraj(:,DynP));
      tnommaxq=starttraj(nommaxqindex,Time);
   %**********Reset Counters********************
       overtemp=[0 0];
       launch_max(1,loop2)=data(1, Time, loop2);
       launch_max(2,loop2)=data(1,Alt,loop2);
       [high_tempo(loop2)
high_tempo_count(loop2)]=max(data(1:length,Shield_Tempo, loop2));
       time_max_tempo(loop2)=data(high_tempo_count(loop2), Time, loop2);
       pst_lnch_max_tempo(loop2)=time_max_tempo(loop2)-launch_max(1,loop2);
       [high_tempi(loop2)
high_tempi_count(loop2)]=max(data(1:length,Shield_Tempi, loop2));
       time_max_tempi(loop2)=data(high_tempi_count(loop2), Time, loop2);
       subsonic=0;
       [max_acc(loop2) max_acc_count(loop2)]=max(data(1:length,Sensed_Acc,
loop2));
       time_max_acc(loop2)=data(max_acc_count(loop2), Time, loop2);
       thust_max_acc(loop2)=data(max_acc_count(loop2), Thrst, loop2);
```

```
        pst_lnch_max_acc(loop2)=time_max_acc(loop2)-launch_max(1,loop2);
        [max_pressure(loop2) max_pressure_count(loop2)]=max(data(1:length, DynP,
loop2));
        time_max_pressure(loop2)=data(max_pressure_count(loop2), Time, loop2);
        [max_HeatRate(loop2)
max_HeatRate_count(loop2)]=max(data(1:length,Heat_Rate, loop2));
        time_max_HeatRate(loop2)=data(max_HeatRate_count(loop2), Time, loop2);
        disp(sprintf('Initial altitude %d meters in file %s', data(1,Alt,loop2), tempfile));
    %******************************************
        for loop=(1:length)
          if (loop > 1) && (loop2 > 1) &&(data(loop, Time, loop2)==0) &&
changelast==0
              last=loop;
              changelast=1;
          end    %if statement
%           if ((data(loop, Shield_Tempo, loop2) > maxtemp) &&
(overtemp(number,finish) ~= 1))
%               overtemp(number, start)=data(loop, Time, loop2);
%               overtemp(number, finish)=1;
%           end;
%           if ((data(loop, Shield_Tempo, loop2) < maxtemp) && (overtemp(number,
finish) == 1))
%               overtemp(number, finish)=data(loop,Time,loop2);
%               timemax=overtemp(number,finish)-overtemp(number,start);
%               tottime=tottime+timemax;
%               disp(sprintf('Time spent over maximum temperature %d', timemax));
%               time1=overtemp(number,start);
%               time2=overtemp(number,finish);
%               disp(sprintf('From %d seconds', overtemp(number, start)));
%               disp(sprintf(' until %d seconds', overtemp(number,finish)));
%               disp('   ');
%               number=number+1;
%               overtemp(number,start)=0;
%           end;
%           if ((subsonic==0) && (data(loop,Mach,loop2) < 1))
```

```
%               subsonic(loop2)=loop;
%           end;
       end; %loop (length)
    end; %loop2 (depth)

%Plots **************************************************
save test.mat starttraj data %max_pressure max_acc high_temp
fignum=1;
```

```
if plotit==1

   n=3;

   figure(fignum)
   subplot(n,1,1),plot(starttraj(:,Time), starttraj(:,Alt),'k');
   title('Initial Trajectory: Altitude vs Time');
   %xlabel('Time (s)');
   ylabel('Altitude (m)');
   subplot(n,1,2),plot(starttraj(:,Time), starttraj(:,Sensed_Acc),'k');
   ylabel('Acc (g''s)');
   subplot(n,1,3),plot(starttraj(:,Time), starttraj(:,Roll),'k');
   ylabel('Roll (deg)');
   xlabel('Time (s)');
%    tempfile=sprintf('%sfig%d',foldpath,fignum);
%    print -djpeg tempfile
   fignum=fignum+1;
%************************
%     plot(max_pressure);
%     xlabel('trial number');
%     ylabel('Q (Pa)');
    n=5;
%    subplot(n,1,1),plot(temp1));
%    xlabel('trial number');
```

```
%    ylabel('Initial Alt (m)');
```

```
   figure(fignum)
   subplot(n,1,1),plot(max_pressure,'ro');
   %xlabel('trial number');
   ylabel('Q (Pa)');
   title('Hashest Conditions per Run');
   subplot(n,1,2),plot(max_acc,'ro');
   %xlabel('trial number');
   ylabel('Acceleration (g s)');
```

```
   %figure(3)


   subplot(n,1,3),plot(high_tempo,'ro');


   %xlabel('trial number');
   ylabel('Outer Temp (C)');

   subplot(n,1,4),plot(high_tempi,'ro');
   %xlabel('trial number');
   ylabel('Inner Temp (C)');

   subplot(n,1,5),plot(max_HeatRate,'ro');
   xlabel('trial number');
   ylabel('Heat Rate (kW/m^2)');

%     tempfile=sprintf('%sfig%d.fig',foldpath,fignum);
%     print -djpeg tempfile
   fignum=fignum+1;
%*****************************
   n=5;
   temp=data(1,Alt,:);


   last=size(max_acc);
   figure(fignum)
   subplot(n,1,1),plot(max_acc,'ro');
   %xlabel('trial number');
   ylabel('Acceleration (g s)');
   title('Worst Accelerations');

   %figure(2)
   subplot(n,1,2),plot(thust_max_acc,'ro');
   %xlabel('trial number');
   ylabel('Thrust (g s)');

   subplot(n,1,3),plot(time_max_acc,'ro'); %,1:last,tnommaxq,'*');
   %xlabel('trial number');
   ylabel('Time of Max Acc (s)');

%     subplot(n,1,4),plot(launch_max,'ro');
```

```
%     %xlabel('trial number');
%     ylabel('Time of Lifeboat Launch (s)');

    subplot(n,1,4),plot(pst_lnch_max_acc,'ro');
    %xlabel('trial number');
    ylabel('Post Ejection Time Max Acc (s)');

    subplot(n,1,5),plot(launch_max(2,:),'ro');
    xlabel('trial number');
    ylabel('Altitdue at Ejection (m)');

    fignum=fignum+1;
%******************************
    n=4;
    figure(fignum)
    subplot(n,1,1),plot(high_tempo,'ro');
    %xlabel('trial number');
    ylabel('Outer Temp (C)');


    title('Worst Temperature');

    subplot(n,1,2),plot(time_max_tempo,'ro'); %,1:last,tnommaxq,'*');
    %xlabel('trial number');
    ylabel('Time of Max Outer Temp (s)');

%     subplot(n,1,3),plot(launch_max,'ro');
%     xlabel('trial number');
%     ylabel('Time of Lifeboat Launch (s)');

    subplot(n,1,3),plot(pst_lnch_max_tempo,'ro');
    %xlabel('trial number');
    ylabel('Post Ejection Time Max Temp (s)');

    subplot(n,1,4),plot(launch_max(2,:),'ro');
    xlabel('trial number');
    ylabel('Altitdue at Ejection (m)');

    fignum=fignum+1;

%******************************
%xtext='Post Escape Time (s)';
xtext='Post Nominal Launch Time (s)';
figure(fignum)
  length=max(data(1,Lgth,:));
```

```
  temp1=[data(1:length,Alt,1)];

  for loop=1:depth

%       for loop2=1:data(1,Lgth,loop)
%           data(loop2,Time,loop)=data(loop2,Time,loop)-data(1,Time,loop);
%       end
      temp2(loop)=data(1,Time,loop);

      %temp2(loop)=data(1,Time,loop)-launch_max(loop);
      temp3(:,loop)=data(:,Shield_Tempo,loop);

   end
  mesh(temp2, temp1, temp3);
   xlabel(xtext);
   ylabel('Altitude (m)');
   zlabel('Outer Shield Temperature (C)');
   title('Outer Shield Temperature Contour Plot');
   colorbar;
%     tempfile=sprintf('%sfig%d.fig',foldpath,fignum);
%     print -djpeg tempfile
   fignum=fignum+1;
%********************************
  figure(fignum)
  length=max(data(1,Lgth,:));
  temp1=[data(1:length,Alt,1)];
  for loop=1:depth
      temp3(:,loop)=data(:,Shield_Tempi,loop);
   end
  mesh(temp2, temp1, temp3);
   xlabel(xtext);
   ylabel('Altitude (m)');
   zlabel('Inner Shield Temperature (C)');
   title('Inner Shield Temperature Contour Plot');
   colorbar;
%     tempfile=sprintf('%sfig%d.fig',foldpath,fignum);
```

```
%     print -djpeg tempfile
   fignum=fignum+1;
%******************************
figure(fignum)
  for loop=1:depth
      temp3(:,loop)=data(:,Sensed_Acc,loop);
   end
   mesh(temp2, temp1, temp3);
   ylabel('Initial Altitude (m)');
   xlabel(xtext);
   zlabel('Acceleration (g''s)');
   title('Acceleration Contour Plot');
   colorbar;
%     tempfile=sprintf('%sfig%d.fig',foldpath,fignum);
%     print -djpeg tempfile
   fignum=fignum+1;
%************************************
  figure(fignum)
  for loop=1:depth
      temp3(:,loop)=data(:,Thrst,loop);
   end
   mesh(temp2, temp1, temp3);
   ylabel('Initial Altitude (m)');
   xlabel(xtext);
   zlabel('Thrust (g)');
   title('Acceleration Contour Plot');
   colorbar;
%     tempfile=sprintf('%sfig%d.fig',foldpath,fignum);
%     print -djpeg tempfile
   fignum=fignum+1;
%************************************

        if (depth < 6)
     fignum=fignum+1;
     figure(fignum)
     n=4;

     colors=['r ' 'y ' 'g ' 'b ' 'c ' 'm '];% 'ro' 'y*'];
     hold on %******************************
      subplot(n,1,1),plot(starttraj(:,Time),starttraj(:,Alt),'k');


      %xlabel('Time (s)');
```

```
        ylabel('Alt (m)');
        title('Simulation Results with Mock Baseline Trajectory')
        hold on %******************************
        subplot(n,1,2),plot(starttraj(:,Time),starttraj(:,Sensed_Acc),'k');
        %xlabel('Time (s)');
        ylabel('Acceleration (g s)');
        hold on %******************************
        subplot(n,1,3),plot(starttraj(:,Time),starttraj(:,Shield_Tempo),'k');


        %xlabel('Time (s)');
        ylabel('Temperature (C)');
        hold on %******************************
        subplot(n,1,4),plot(starttraj(:,Time),starttraj(:,DynP),'k');
        %xlabel('Time (s)');
        ylabel('Q (Pa)');
       hold on %******************************
       for loop=1:depth
          lnght=data(1,Lgth,loop);

subplot(n,1,1),plot(data(1:lnght,Time,loop),data(1:lnght,Alt,loop),colors(loop));
          %xlabel('Time (s)');
          ylabel('Alt (m)');
          hold on %******************************

subplot(n,1,2),plot(data(1:lnght,Time,loop),data(1:lnght,Sensed_Acc,loop),colors(loo
p));
          %xlabel('Time (s)');
          ylabel('Acceleration (g s)');
          hold on %******************************

subplot(n,1,3),plot(data(1:lnght,Time,loop),data(1:lnght,Shield_Tempo,loop),colors(l
oop));
          %xlabel('Time (s)');
          ylabel('Temp (C)');
          hold on %******************************

subplot(n,1,4),plot(data(1:lnght,Time,loop),data(1:lnght,DynP,loop),colors(loop));
          xlabel('Time (s)');
          ylabel('Q (Pa)');
        end %subplots
        hold off
%       tempfile=sprintf('%sfig%d.fig',foldpath,fignum);
```

```
%         print -djpeg tempfile
      fignum=fignum+1;
         end %if
end %Plots

   stats=depth;

%end %stats2
```

```
function density=Get_Density(alt);
%Last Update 1/6/05
%
%Given altitude above Earth returns Air Density
%
%Functions Called By:
%   aerocalcs
%   accelfunct
global track_dens denscount ndens

   H=7128.1;       %atm scale ht, m
   rho0=1.478;     %atm. dens, sea level
   scale_ht=exp(-alt/H);
   %alt
   density=rho0*scale_ht; %****************************
   if track_dens==1
      denscount=denscount+1;
      ndens(denscount,1)=alt;
      ndens(denscount,2)=density;
   end %track_dens
   %end Get_Density
```

```
function statedot=accelfunct(t, state)
%Last Update 2/24/04
%Functions Called By:
%   aerocalcs
%Calls Functions
%   Get_Density(alt)
%   GetGravity
%Globals*************************
global mu Re LoD d2r beta;
global Velocity Gamma Height Rt
global Rollvar

countdrag=1;
%*********************************
```

```
   density=Get_Density(state(Height));
   g=GetGravity(state(Height));
   alt=state(Height)+Re;
   vcirc=sqrt(mu/alt);
   %drag=0;
   mach=getMach(state(Velocity), alt);
   Cp0=Get_Cp0(mach);
   drag=density*state(Velocity)^2/(beta/Cp0);
   lift=drag*LoD;
   %gdot=((G/Velocity-Velocity/(Re+Alt))*COS(Gamma*d2r)-
G*Density*Velocity*LD*COS(Roll*d2r)/(3.7*beta/Cp0))
   %statedot(Gamma)=((g/state(Velocity)-state(Velocity)/alt)*cos(state(gamma)-
g*lift*cos(roll)
   statedot(Gamma)=(lift-(1-
(state(Velocity)/vcirc)^2)*g*cos(state(Gamma)))*cos(Rollvar)/state(Velocity);
   statedot(Velocity)=-drag-g*sin(state(Gamma));
   statedot(Height)=state(Velocity)*sin(state(Gamma));
   statedot(Rt)=state(Velocity)*cos(state(Gamma));
   statedot=statedot';
%end %accelfunct
```

```
%September 26,
2003****************************************************
function mach=getMach(Vel, Altitude)
   alt=1;
   speed=2;
   counter=0;
   maxalt=100000;
   for loop=0:500:3000;
      counter=counter+1;
      soundvel(counter,alt)=loop;
   end;
   for loop=4000:1000:20000
      counter=counter+1;
      soundvel(counter,alt)=loop;
   end
   for loop=25000:5000:maxalt
      counter=counter+1;
      soundvel(counter,alt)=loop;
   end
   soundvel(1,speed)=340.294;
   soundvel(2,speed)=338.37;
   soundvel(3,speed)=336.435;
   soundvel(4,speed)=334.489;
   soundvel(5,speed)=332.532;
   soundvel(6,speed)=330.563;
```

```
   soundvel(7,speed)=328.583;
   soundvel(8,speed)=324.589;
   soundvel(9,speed)=320.545;
   soundvel(10,speed)=316.452;
   soundvel(11,speed)=312.306;
   soundvel(12,speed)=308.105;
   soundvel(13,speed)=303.848;
   soundvel(14,speed)=299.532;
   soundvel(15,speed)=298.638;
   soundvel(16,speed)=297.745;
   soundvel(17,speed)=296.851;
   soundvel(18,speed)=295.958;
   soundvel(19,speed)=295.069;
   soundvel(20,speed)=294.466;
   soundvel(21,speed)=294.148;
   soundvel(22,speed)=294.452;
   soundvel(23,speed)=294.726;
   soundvel(24,speed)=295.069;
   soundvel(25,speed)=298.389;
   soundvel(26,speed)=301.709;
   soundvel(27,speed)=309.449;
   soundvel(28,speed)=317.189;
   soundvel(29,speed)=323.494;
   soundvel(30,speed)=329.799;
   soundvel(31,speed)=325.203;
   soundvel(32,speed)=320.606;
   soundvel(33,speed)=308.873;
   soundvel(34,speed)=297.139;
   soundvel(35,speed)=283.29;
   soundvel(36,speed)=269.44;
   soundvel(37,speed)=288.88;
   soundvel(38,speed)=308.31;
   soundvel(39,speed)=327.75;
   soundvel(40,speed)=347.18;

   if Altitude > maxalt
      Altitude = maxalt;
   end
   speedsound=interp1(soundvel(:,alt),soundvel(:,speed),Altitude);
%    temp=size(soundvel(:,alt));
%    for loop=1:temp
%        fprintf('%15.6f', soundvel(loop,:));
%        fprintf('\n');
%   end
   mach=Vel/speedsound;
   %end %getMach
```

```
function Cp=Get_Cp0(mach)

   Cp=1.35-(atan(1.96*(1-mach)))/pi;
%end %CP0
```

```
function density=Get_Density(alt);
%Last Update 1/6/05
%
%Given altitude above Earth returns Air Density
%
%Functions Called By:
%   aerocalcs
%   accelfunct
global track_dens denscount ndens

   H=7128.1;       %atm scale ht, m
   rho0=1.478;     %atm. dens, sea level
   scale_ht=exp(-alt/H);
   %alt
   density=rho0*scale_ht; %****************************
   if track_dens==1
      denscount=denscount+1;
      ndens(denscount,1)=alt;
      ndens(denscount,2)=density;
   end %track_dens
   %end Get_Density
```

```
function drag=Get_Drag(v, density)
%Last Update 1/12/04
%Functions Called By:
%   accelfunct

   global beta Re
   global track_drag ndrag dragcount
   global Velocity

   earthrate=7.292115e-5;
   drag=.5*density*v^2;
   %end %Get Drag
```

```
function [kappav, kappac, cvv, cvc]=Get_Kappas(temp)
%Given the thermal coefficients of an ablative substance and the
%temperature will find the thermal conductivity of both charred and virgin
%material
```

```
minchar=37.96;
maxvirg=560.1833;

temperature=1;
virgval=2;
charval=3;
vheatcap=4;
cheatcap=5;

kappas(1,:)=[-100.928 .02130881 nan 1171.52 nan];
kappas(2,:)=[-17.59444444 0.02878558 nan 1171.52 nan];
kappas(3,:)=[37.96111111 0.033645483 0.059814192 1171.52 857.72];
kappas(4,:)=[143.5166667 0.042991451 0.059814192 1171.52 857.72];
kappas(5,:)=[171.2944444 0.046729838 0.059814192 1171.52 857.72];
kappas(6,:)=[282.4055556 0.046729838 0.059814192 1171.52 857.72];
kappas(7,:)=[560.1833333 nan 0.059814192 nan 857.72];
kappas(8,:)=[699.0722222 nan 0.065421773 nan 1096.208];
kappas(9,:)=[837.9611111 nan 0.072898547 nan 1138.048];
kappas(10,:)=[976.85 nan 0.083739869 nan 1158.968];
kappas(11,:)=[1185.183333 nan 0.102805643 nan 1204.992];
kappas(12,:)=[1254.627778 nan 0.121497578 nan 1213.36];
kappas(13,:)=[1393.516667 nan 0.142058706 nan 1213.36];
kappas(14,:)=[1532.405556 nan 0.177573383 nan 1213.36];
kappas(15,:)=[1671.294444 nan 0.201872899 nan 1213.36];

kappas(:,2:5)=kappas(:,2:5)/1000;
if temp < maxvirg
   kappav=interp1(kappas(:,temperature),kappas(:,virgval),temp);
   cvv=interp1(kappas(:,temperature),kappas(:,vheatcap),temp);
else
   kappav=nan;
   cvv=nan;
end
if temp > minchar
   kappac=interp1(kappas(:,temperature),kappas(:,charval),temp);
   cvc=interp1(kappas(:,temperature),kappas(:,cheatcap),temp);
else
   kappac=nan;
end
```

```
function g=GetGravity(height)
%Last Update 2/24/04
%Functions Called By:
%   accelfunct
%Calls Functions
```

```
%Globals*************************
global gravity Re
%*********************************
g=gravity*(Re^2)/((height+Re)^2);
%end GetGravity
```

```
function roll=GetRoll(oldroll, time)
%
%function roll=GetRoll(time)
%Last Update 4/14/04
%Functions Called By:
%   aerocalcs

   global rolleqtn Rollcount

   tou=174.336478873392;
   %tou=100;
% ****** rolleqtn='90*(sin(time/tou))^1';*******

%     if ((abs(abs(oldroll)-90)< 1) && (rem(Rollcount,50)~=0))
%        Rollcount=Rollcount+1;
%        roll=oldroll;
%     else
%        roll=eval(rolleqtn);
%     end %if statement
roll=eval(rolleqtn);
%end %GetRoll
```

```
%September 26 2003*********************************************
function shieldtemp=getShield(heatrate)
   temp=((229100580000 * heatrate/2)^.25)-460;
   if temp < 60
      shieldtemp=60;
   else
      shieldtemp=temp;
   end
   %end%shieldtemp
```

Glossary

ACES	Advanced Crew Escape Suit
AMROC	American Rocket Company
AOA	Abort Once Around
ATO	Abort To Orbit
CAIB	Columbia Accident Investigation Board
deltaT	Simulation Time Interval
EI	Entry Interface
GPC	General Purpose Computer
h	Altitude
L/D	Lift to Drag Ratio
LAS	Launch Abort System
LES	Launch Escape System
LES	Launch and Entry Suit
max q	Maximum Dynamic Pressure
MECO	Main Engine Cut Off
MOOSE	Manned Orbital Operations Safety Equipment
OAMS	Orbital Attitude Maneuvering System
OMS	Orbital Maneuvering System
PEAP	Personel Egress Air Pack
RCS	Reentry Control System
RTSL	Return to Launch Site
SRB	Solid Rocket Booster

TAL	Transatlantic Abort Landing
US	United States
v	Velocity
γ	Flight Path Angle

Bibliography

[1] Purser, Paul E. Maxime A. Faget, and Norman F. Smith. Manned Spacecraft: Engineering Design and Operation. Fairchild Publications Inc. New York, 1964.

[2] "The Space Educators Handbook" http://vesuvius.jsc.nasa.gov/er/seh/crewcap.html

[3] Wade, Mark. "Encyclopedia Astronautica" www.astronautix.com

[4] Newkirk, Dennis. "Almanac of Soviet Manned Space Flight" Gulf Publishing Co. Houston, TX, 1990.

[5] Von Braun, Wernher. Space Frontier. Holt, Rinhart and Winston, New York, 1963.

[6] Brodsky, R.F. "Space Station Escape Vehicle - 40 Years Later." Proceedings 43rd Annual AIAA Sciences Meeting and Exhibit, (10-13 January, 2005, Reno NV), Paper Number 2005-704.

[7] Harland, David M. The Story of the Space Shuttle. Praxis Publishing, Chichester, UK, 2004.

[8] "NSTS 1988 News Reference Manual" http://science.ksc.nasa.gov/shuttle/technology/sts-newsref/stsref-toc.html cited March 15, 2005.

[9] Purser, Paul E. Maxime A. Faget and Norman F. Smith. Manned Spacecraft: Engineering Design and Operation. Fairchild Publications Inc. New York, 1964.

[10] Catchpole, John. Project Mercury, NASA's First Manned Space Programme. Praxis Publishing, Chichester, UK, 2001.

[11] Furniss, Tim. Manned Space Flight Log. Van Nostrand Reinhold Co Inc. Boston, MA, 1983.

[12] Hacker, Barton C. and James M. Grimwood. On the Shoulders of Titans: A History of Project Gemini. NASA Scientific and Technical Information Office, Washington DC, 1977.

[13] Shayler, David J. Gemini, Steps to the Moon. Praxis Publishing, Chichester, UK, 2001.

[14] Siddiq, Asif A. The Soviet Space Race With Apollo. University Press of Florida, Gainsville, Florida, 2000.

[15] Bond, Peter. Heroes in Space, From Gagarin to Challenger. Basil Blackwell Inc. Oxford, UK, 1987.

[16] Kerwin, Joseph P. Memo to RADM Richard H. Truly. July 28, 1986. " http://history.nasa.gov/kerwin.html"

[17] 2003 Space Shuttle Columbia Trajedy: Columbia Accident Investigation Board (CAIB) Final Report, August 2003, The Gehman Board Report to NASA on the Space Shuttle Program. Progressive Management, 2003.

[18] "STS-107 Timeline-Rev 15 Summary Time Line." "http://history.nasa.gov/columbia/Troxell/Columbia%20Web%20Site/Timelines/summary_timeline.htm"

[19] Troxell, Jennifer. "Columbia Recommendation Spreadsheet" http://history.nasa.gov/columbia/Troxell/Columbia%20Web%20Site/CAIB/COLUMB%7E1.HTM

[20] Howard, Russ. “Reentry Spreadsheet” Space Systems Laboratory, University of Maryland, College Park, MD, 1988. (unpublished)

[21] Hale, Francis J. Introduction to Space Flight. Prentice Hall, Englewood Cliffs, NJ, 1994. pgs. 225-234.

[22] Chapman, Dean R. “An approximate Analytical Method for Studying Entry into Planetary Atmospheres” NACA TN-4276. URL: http://naca.larc.nasa.gov/reports/1958/naca-tn-4276/ [cited November 2, 2004]

[23] Swenson, Loyd S. James M. Grimwood and Charles C. Alexander. This New Ocean: A History of Project Mercury. NASA Scientific and Technical Information Office, Washington DC, 1966.

[24] "NASA-STD-3000, Man-System Integration Standards Revision B" NASA JSC, July 1995, Vol. 1. http://msis.jsc.nasa.gov/Volume1.htm [cited April 13, 2005].

[25] Wilde, Detlef and Stephan Walther “Inflatable Reentry and Descent Technology (IRDT) – Further Developments” published in “2nd International Symposium of Atmospheric Reentry Vehicles and Systems” Arcachon, France, 26-29 March 2001.

[26] Adams, Douglas. The Ultimate Hitchhiker's Guide. Wings Publishing, San Antonio TX, 1996.