University of Texas at Arlington
MavMatrix

2015 Spring Honors Capstone Projects Honors College

5-1-2015

ORBITAL ELEVATORS A PERFORMANCE ANALYSIS

Dierdra O'Donoghue

Recommended Citation

O'Donoghue, Dierdra, "ORBITAL ELEVATORS A PERFORMANCE ANALYSIS" (2015). *2015 Spring Honors Capstone Projects*. 11.
https://mavmatrix.uta.edu/honors_spring2015/11

ORBITAL ELEVATORS
A PERFORMANCE
ANALYSIS

by

DEIRDRA O'DONOGHUE

Presented to the Faculty of the Honors College of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

HONORS BACHELOR OF SCIENCE IN ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON
May 2015

ACKNOWLEDGMENTS

Acknowledgments go to my mentor Dr. Chudoba and my Engineering Group Project team, Stella Nova Aeronautics, especially the Performance group. Further acknowledgements go to DT, you know who you are.

May 08, 2015

ABSTRACT

ORBITAL ELEVATORS:
A PERFORMANCE
ANALYSIS

Deirdra O'Donoghue, AE

The University of Texas at Arlington, 2015

Faculty Mentor: Chudoba

The space race is reborn, with companies competing to be the first to send their paying passengers to space. Therefore, an economically viable route includes modifying an existing vehicle for this task. As such, the X–15 is the rocket plane for the job, as its capabilities include reaching the edge of space. The redesigned vehicle will be analyzed for performance characteristics from both the air drop and horizontal take–off single stage suborbital and orbital maneuvers. The applicable performance equations will be crafted into codes in Matlab, and first order estimations of the necessary performance parameters will be acquired. These results will be compared to competitors in order to determine feasibility and competitiveness. Performance results show that our vehicle requires a greater thrust than the original vehicle, but a similar burn time. Therefore, the redesigned space ship of tomorrow is a viable contender for cornering the market on space tourism.

TABLE OF CONTENTS

Appendix

LIST OF ILLUSTRATIONS

LIST OF TABLES

Table Page

CHAPTER 1

INTRODUCTION

As the age of space tourism arises, an economically viable transportation system is necessary. Consequently, for this design project, the X-15 will be designed to reach suborbital and orbital heights both from the originally designed B-52 drop off and a horizontal take-off and landing single stage configuration. Information necessary for the performance analysis of this vehicle will be acquired from previous NASA reports, space tourist companies, and other sources.

.

CHAPTER 2

LITERATURE REVIEW

2.1 X-15 History and Mission

“The X-15 from 1959 was built by North American in order to acquire data at hypersonic speeds of up to Mach 6.7. This information was necessary in order to advance the creation of the space shuttle enabling transport to the International Space Station [1]. Furthermore, the X-15 is capable of going speeds of up to 6,000 ft/s and reaching an altitude of 350,000 ft. In order to reach these high speeds, it was equipped with a single ‘Reaction Motors YLR99 variable-thrust, liquid-rocket engine’, that was capable of 50,000 pounds of thrust at sea level [2]. The X-15 airplane possesses a wedge shaped tail of 10o, as well as an all – moveable horizontal surface and removable lower rudder. Two of the three X- 15’s produced possess a stability augmentation system that provides damping through the control surfaces to increase stability. The third plane possesses a Honeywell adaptive control system that adapts to the control surface effectiveness to produce a constant response to pilot input [2]” [3].

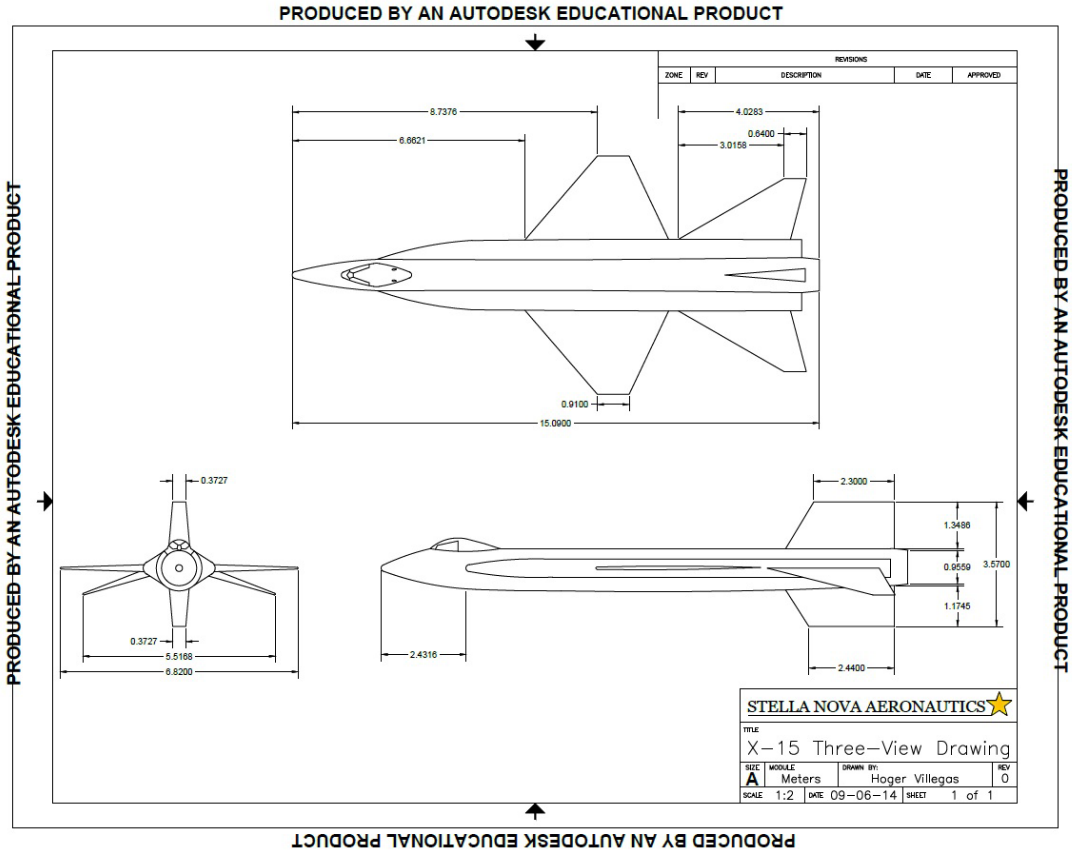


Figure 2.1: The Original X-15 (Created by Mr. Villegas) [3]

"As seen in Fig. 2.1 above, produced by Mr. Villegas, the X – 15 possesses an aft tail with a horizontal surface, as well as a detachable lower vertical surface." [3].

2.2 Space Tourism Today

In order to design a suborbital space vehicle, current designs will be used for comparison. The first vehicle under consideration is Virgin Galactic's Space Ship Two. As it is dropped from a height of 50,000 ft by another air vehicle, it will serve as an example for the first air drop mission of the X-15. Furthermore, the two pilots and six passengers configuration used by the Space Ship Two will be the payload basis for our suborbital mission [4].

The flight path, as seen below in Fig. 2.2, of the Space Ship Two will be used for the air drop mission redesign of the X-15.

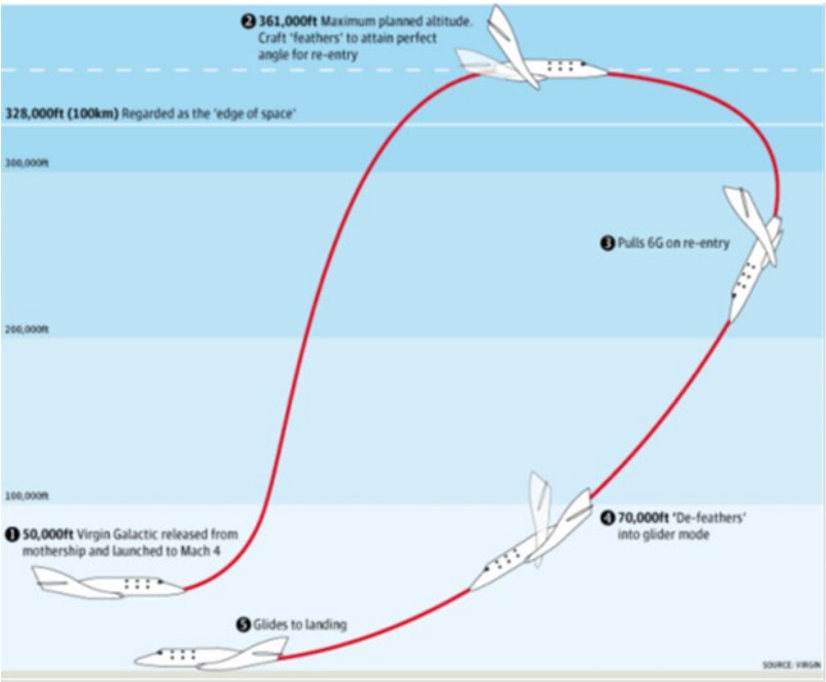


Figure 2.2: Air Drop Mission Flight path
(provided by Mr. Beassie courtesy of Virgin Galactic) [3]

Another vehicle designed for space tourism is Blue Origin's New Shepard. However, as it is a two stage vehicle and takes off vertically, rather than horizontally, it will not be used as a reference [5].

Although the vehicle designed by Bristol Spaceplanes, the Ascender, takes off horizontally, it does so using a turbofan engine rather than the rocket engine required for the second design. Furthermore, it only seats one passenger in addition to the pilot so space will be an issue [6].

Finally, XCOR Aerospace's Lynx takes off horizontally and lands horizontally, as required for the second suborbital mission, but it only seats one passenger in addition to the pilot. This design also possesses the capability of taking off at any airport with a 7,900 ft runway using rocket propulsion that matches the design requirements of the second mission [7].

For the horizontal take – off mission, our flight plan will resemble that of the Lynx, shown below in Fig. 2.3.

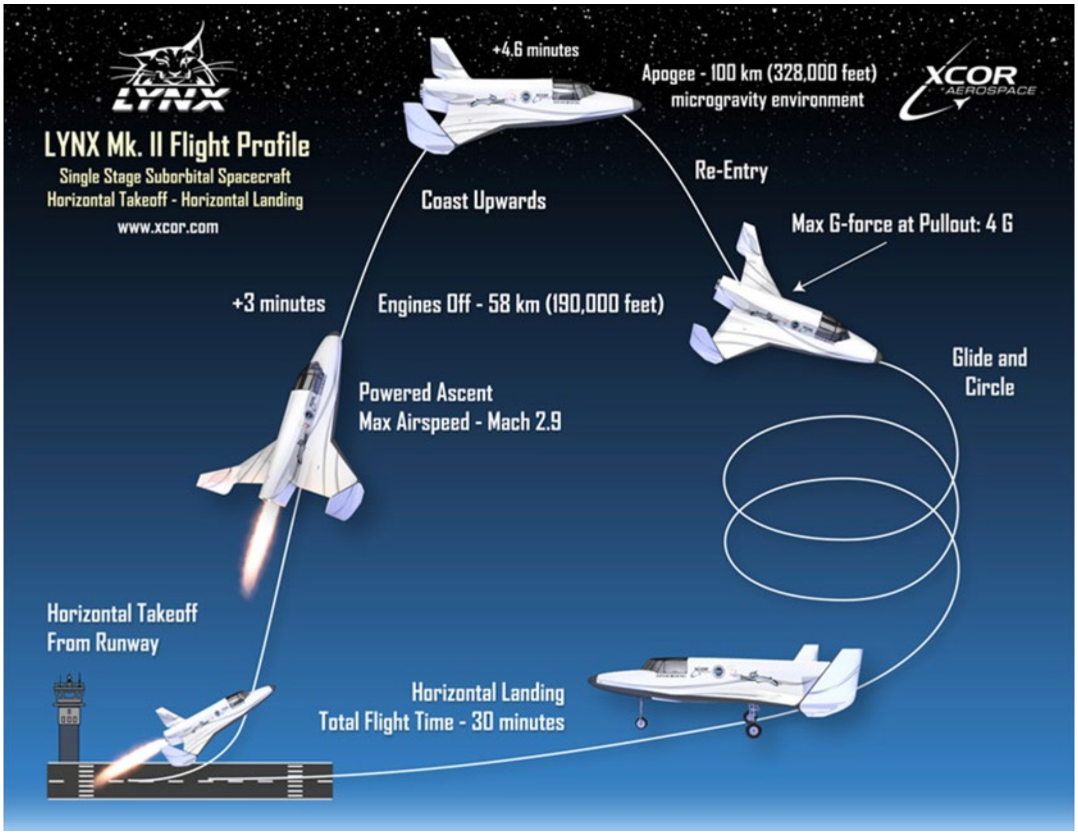
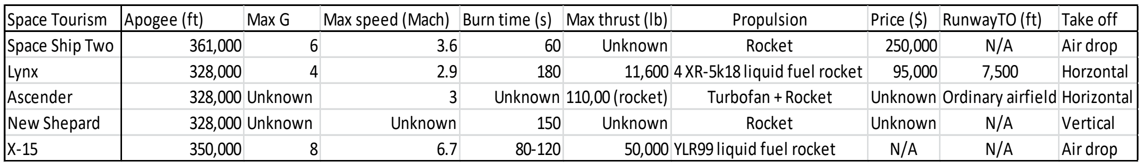


Figure 2.3: Horizontal Take – off mission flight plan [7]

The table below displays the information gleaned from the above space companies.

Table 2.1: Performance of Suborbital Competitors
[1,2,3,4,5,6,7,8,9, Mr. Pham of the Geometry Group, Mr. Villegas our Chief Engineer]



Space Tourism	Apogee (ft)	Max G	Max speed (Mach)	Burn time (s)	Max thrust (lb)	Propulsion	Price ($)	RunwayTO (ft)	Take off
Space Ship Two	361,000	6	3.6	60	Unknown	Rocket	250,000	N/A	Air drop
Lynx	328,000	4	2.9	180	11,600	4 XR-5k18 liquid fuel rocket	95,000	7,500	Horzontal
Ascender	328,000	Unknown	3	Unknown	110,00 (rocket)	Turbofan + Rocket	Unknown	Ordinary airfield	Horizontal
New Shepard	328,000	Unknown	Unknown	150	Unknown	Rocket	Unknown	N/A	Vertical
X-15	350,000	8	6.7	80-120	50,000	YLR99 liquid fuel rocket	N/A	N/A	Air drop

Therefore, based upon the above table, the only vehicles for consideration and comparison for the air drop mission are the original X – 15 and the Space Ship Two. Furthermore, the Lynx will be used for comparison on the horizontal take off mission as it fulfills the parameters.

2.3 Literature Overview

For the performance analysis of the new designs, information will be taken from NASA technical documents, corporate provided information, and previous design studies done on the X-15. Since the technical documents are provided by NASA, the veracity of the data is sound [1, 2]. However, the information provided by corporations about their own designs will be sparse and suspect at best [4, 5, 6, 7], and therefore more reliable sources analyzing and comparing these vehicles overall will be used, as in Ref. [10]. However, as this information is over a decade old, some more recent designs may not be included in the reference. Therefore, more recent information as provided by the FAA and NASA will be used as well [8,9,11]. In order to analyze the performance of the design, the course text [12], previous design studies [3], and texts from previous courses [13,14] will be used. The appropriate equations pulled from these sources will be valid as long as the necessary conditions are met for their application. Further sources detailing suborbital competitors will rely upon texts provided by the design lab, as these will be accurate and current [15]. Finally, information discussing performance equations acquired from previous courses in the form of notes will also be used in order to analyze the redesigned X – 15 [16]. Clearly, the veracity of these sources will depend upon the notes taken, but otherwise will be accurate. Also, previous equations and codes used by the performance team of last semester will be analyzed for applicability and used if this is the case [17]. The veracity of this source is good since the codes have already been tested out for the air drop mission configuration. Finally, comparison with FAA certifications will ensure safe passenger flight and certification in the future [18]. This source is reliable since it is supplied by the institution which certifies air vehicles. Further sources will be used in order to define terms [19]. Other sources analyzing either the X – 15 or current suborbital space transport were viewed in the course of designing a new vehicle, but these have not been included based upon redundancy, being unrelated to performance analysis, or unreliable sources.

2.4 Original Mission Values

The original unmodified X – 15 was capable of reaching the atmospheric height necessary for suborbital flight. However, modification is necessary in order to allow for more passengers and ground takeoff [1].

2.5 Lessons

During literature review, the integral relationship between vehicle requirements was demonstrated. Instances of decreased performance in order to increase other factors occur frequently. Therefore, performance will be optimized while compromising with other group requirements.

CHAPTER 3

METHODOLOGY: THE X-15 DESIGN MISSION

For this section we are to redesign the X – 15 research aircraft for two major missions. One includes the classical drop from the B – 52, with a horizontal landing, while the other will take off from the surface horizontally with a single stage and land horizontally as well. The intrinsic differences between these two are apparent, as the second will require more fuel, sturdier landing gear, and other factors due to the increase in weight necessary for this maneuver.

3.1 Team Division

The team will be divided into 7 groups, not including Chief Engineer, consisting of Synthesis, Structures and Geometry, Aerodynamics, Stability and Controls, Propulsion, Performance, and Costs and Certifications.

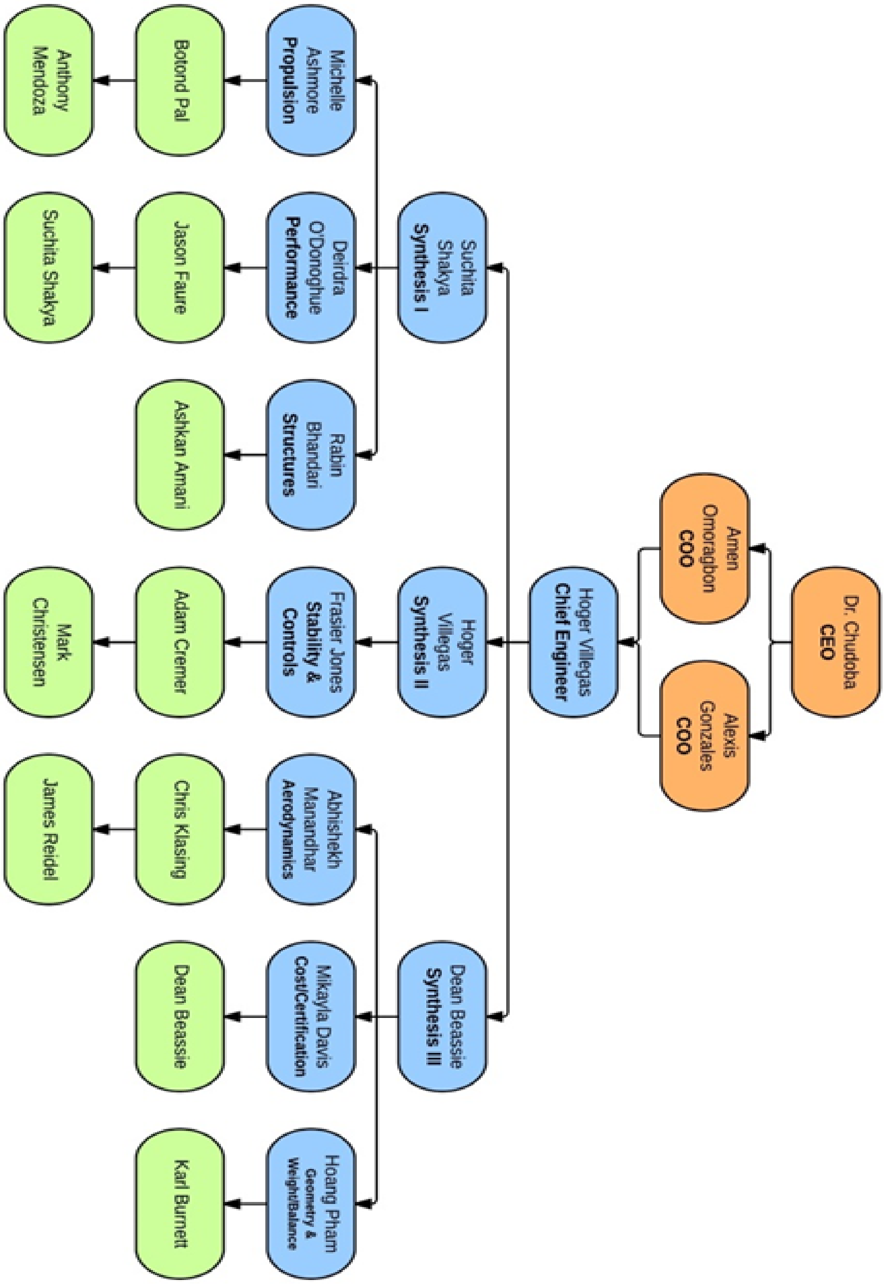


Figure 3.1: Overall Team Structure (Produced by Mr. Villegas)

Figure 3.1 above shows the team division and structure. The average subgroup consists of two to three members, with reshuffling occurring later on if necessary after future tasks are determined.

3.2 Performance: Individual Responsibilities and Scope

As a member of Performance, the main tasks required are defining the necessary variables, coordinating work with other group members, and sending our completed performance analysis results onto the necessary groups, such as Aerodynamics and Costs and Certification.

Furthermore, Propulsions, Aerodynamics, and Geometry and Structures will provide Performance with the necessary variables for analysis.

3.3 Team Responsibilities and Scope

"Since each group is necessary for the overall team to accomplish the project, the specific roles of each group are outlined below" [3].

"The Chief Engineer is our liaison with the professor and directs the team during meetings and other discussions. He decides the overall direction of the team" [3].

"The Synthesis Engineers assist the Chief Engineer in problem solving and are in charge of integrating their respective assigned groups into a cohesive whole" [3]. Furthermore, they will calculate the gross takeoff weights, wing, fuselage, and empennage sizing, and perform a feasibility study, as well as determine landing requirements.

Meanwhile, the Geometry and Structures group will determine the aerodynamic loads, the shear and moment, equilibrium temperature, stress limitations, wing and fuselage stress, center of gravity location, material properties, component weights, and inertias. Furthermore, they will create VN diagrams for the vehicle and a CAD model of our proposed design with its determined geometry.

Stability and Controls will size the empennage, determine the thrust and burn time of the reaction control system necessary, the tail volume coefficient, and the necessary coefficients for stability.

The Costs and Certification group will determine the required costs, the design tolerance, and airworthiness licensing. Secondly, they will perform simulation verification of the design.

Aerodynamics will calculate the necessary coefficients of lift, drag, and friction, determine the angle of attack, the location of the neutral point, and all necessary aerothermal values.

The Propulsion group will determine the fuel and system layout with the installed thrust, as well as other necessary values.

Finally, the Performance group will determine the rate of climb, range, altitude, landing gear analysis, loitering, glide, flight envelope, key angles of attack, takeoff and balance field length, and thrust required vs. thrust available. My personal responsibilities include assisting in defining, calculating, and analyzing the values necessary for a complete performance analysis.

3.4 Schedule

Our overall team schedule so far has determined that we will finish analyzing the horizontal take-off and landing single stage mission by April. After this the same design, with less fuel, will be used for the air drop mission.

3.5 Group Contributions

After the necessary research is amassed, the group will redesign the X – 15 for the horizontal take off missions. After redesign is accomplished, it will be analyzed by the groups in tandem, and then used for the air drop mission and further analyzed.

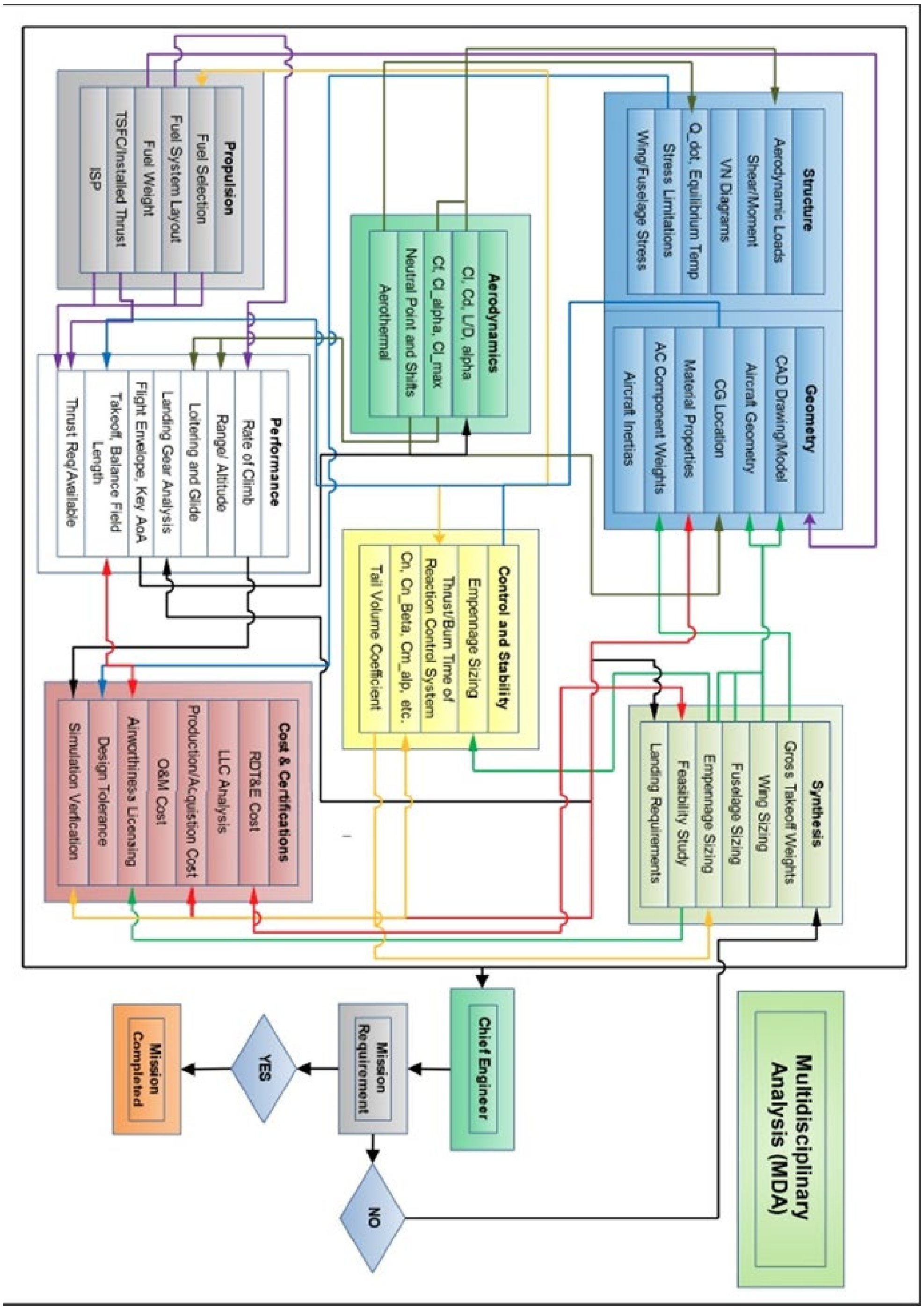


Figure 3.2: Overall Group MDA (Produced by Mr. Pham)

As seen in Fig. 3.2, produced by Mr. Pham, the interrelation of the design analysis is clear. Specifically, for performance analysis, we will acquire the necessary values from propulsions, geometry and structures, and aerodynamics, analyze the designs, and forward them onto the required groups, such as aerodynamics and costs and certifications as shown above in Fig. 3.2.

3.6 Performance Contributions

Currently the performance group consists of Ms. Shakya, Mr. Faure, and me. After researching current space tourism vehicles, we will acquire the necessary values from propulsion, geometry and structures, and aerodynamics, and analyze them using Matlab codes based upon equations acquired from the necessary texts [12, 13, 14]. The necessary outputs, as shown in our IDA and MDA in Fig. 4-3 below, will then be forwarded onto the groups who require this information for their analysis, specifically aerodynamics, costs and certifications, propulsions, and synthesis.

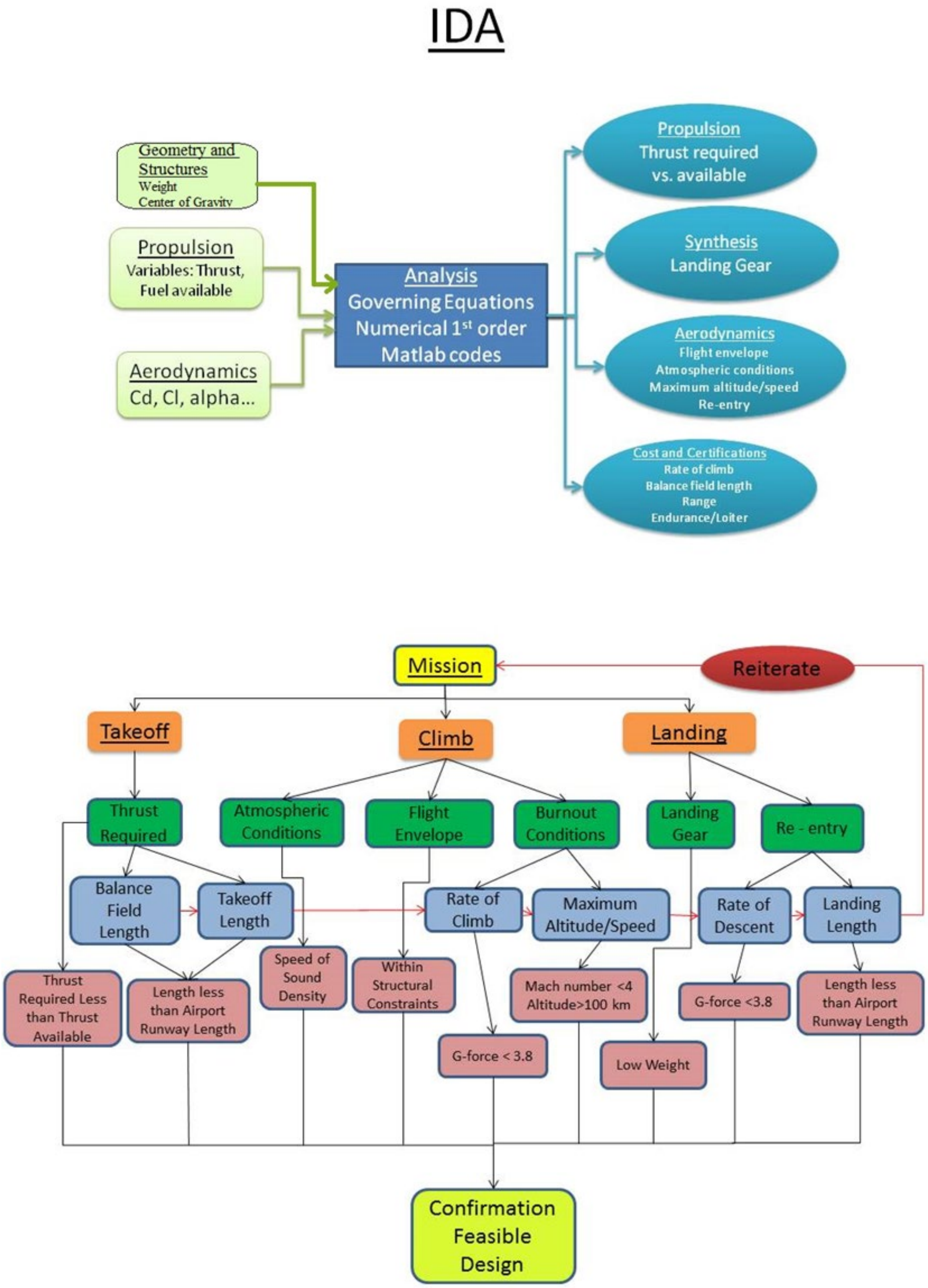


Figure 3.3: Performance IDA and MDA

CHAPTER 4

ANALYSIS

4.1 Performance Analysis of Suborbital Air Drop Mission Design

Based upon current research, it would appear that the Space Ship Two produced by Virgin Galactic would serve as a model, specifically in passenger requirements [4]. However, during early team discussions it was proposed that a double – delta wing, be added to the redesigned X – 15 instead of simply maximizing the wings in their current shape. The double – delta wing is a key feature of the Lynx by XCOR Aerospace that only carries one passenger and takes off horizontally [15]. This configuration was determined based upon the decrease in drag at supersonic speeds reported by the Aerodynamics team for this configuration as compared to the trapezoidal one.

The first values calculated by Performance will be the atmospheric values at certain heights. Therefore, a Matlab code, "geom224.m," using the equations of "standard atmosphere conditions" will be compiled and used at each specific height in order to determine the pressure, density, temperature, and viscosity ratio, along with the speed of sound at each height [16].

First, the geometric altitude will be converted to geopotential altitude, as seen in the equation below.

$$H = \frac{r*z}{r+z} \qquad [16]$$

After this the pressure ratio for a linear temperature gradient is found using the equation shown below.

$$\delta = \frac{P_{b_i}}{P_0}\left[1+\frac{K_{T_i}}{T_{b_i}}\left(H - H_{b_i}\right)\right]^{\frac{-Mg_0}{R^* K_{T_i}}} \qquad [16]$$

Then, the temperature ratio and density ratio are acquired from the perfect gas law, producing the following equations.

$$\theta = \frac{T_{b_i}}{T_0}\left[1+\frac{K_{T_i}}{T_{b_i}}\left(H - H_{b_i}\right)\right] \qquad [16]$$

$$\sigma = \frac{\rho_{b_i}}{\rho_0}\left[1+\frac{K_{T_i}}{T_{b_i}}\left(H - H_{b_i}\right)\right]^{\frac{-Mg_0}{R^* K_{T_i}} - 1} \qquad [16]$$

Therefore, the viscosity ratio is shown in the following equation.

$$\emptyset = \left(\frac{T}{T_0}\right)^{3/2}\left[1+\frac{K_{T_i}}{T_0 + S}\left(H - H_{b_i}\right)\right]^{-1} \qquad [16]$$

Finally, the speed of sound ratio can be determined from the temperature ratio, as shown in the equation below.

$$\chi = \theta^{1/2} \qquad [16]$$

Meanwhile, if the temperature is constant, the pressure ratio is found using the following equation.

$$\delta = \frac{P_{b_i}}{P_0}\, e^{-\left[\frac{Mg_0}{R^* T_{b_i}}\left(H - H_{b_i}\right)\right]} \qquad [16]$$

Therefore, from the perfect gas law and the assumption of constant temperature, the pressure, temperature, viscosity, and speed of sound ratio is determined as above. The code based upon these equations is located in Appendix C. The horizontal take – off mission design will be used with fuel modifications for the final air drop mission design in order to compare efficiency and performance between the two options.

The inputs are acquired from the Geometry team, with air – drop weight, empty weight, and center of gravity, the coefficient of drag, maximum coefficient of lift, and angle from Aerodynamics, and the thrust available and specific impulse from Propulsions. The atmospheric conditions are used as describe above.

Consequently, the codes and equations used for the horizontal performance analysis shown in Section 4.2 will be modified for use during the air drop mission. Since the vehicle is sized for horizontal take – off, the thrust will remain at 70,000 ft, but since we have decided to only reach the height achieved by the horizontal take – off mission, the burnout time will only be 95 s, thus saving approximately 1,000 lb of fuel. The burnout height, velocity, and maximum height are determined using “Thrust2h2.m,” from Appendix C which uses the same basic equations from Section 4.5 below, except with minor modifications taking into account the vehicle being dropped from a height of 50,000 ft. Therefore, the following equations are used.

$$h_{bo} = 15239 + c/m_e*(m_f*log(m_f/m_0) + m_0 - m_f) - 1/2*((m_0-m_f)/me)^2*g \quad [14]$$

$$V_{bo} = c*(log(n) - (g/me)*(m_0-m_f)) \quad [14]$$

$$h_{max} = 15239 + ½*c^2/g*ln^2(n) - c*m_0/me*(n*ln(n) - (n-1))/n \quad [14]$$

The maximum height reached is 340,000 ft, with a burnout velocity of 3200 ft/s and burnout height of 170,000 ft. Consequently it fulfills the flight condition of reaching greater than 100 km. The time in microgravity is determined based upon a set re – entry height of 150,000 ft due to an aerodynamic control condition from Stabilitiy and Controls. The equations below are used to find a microgravity time of 195 s. The code “Thrust2h2.m” also uses the following equation in order to find a re – entry velocity of 4500 ft/s.

$$t_{micro} = (2*(h_{max}-h_{re})/g_{re})^{1/2} \quad [14]$$

$$V_{re} = V_{hmx} - g_{re}*(t_{micro}) \quad [14]$$

Then “FlightPathAngle45dbld4h.m” in Appendix C is used to find the following values.

Table 4.1: Performance Values Suborbital Air – Drop Mission

Performance Values	
Vbo	3200 ft/s
Range burnout	27000 ft
Range	240,000 ft
Max altitude	340,000 ft
Angle	80 degrees

The range is also found using the following equation in

“FlightPathAngle45dbld4.m” and performing integration.

$$x^* = \frac{r}{r+h} * v * \cos(\gamma) \qquad [17]$$

Furthermore, the flight path is found with the above code as well, producing the figure below.

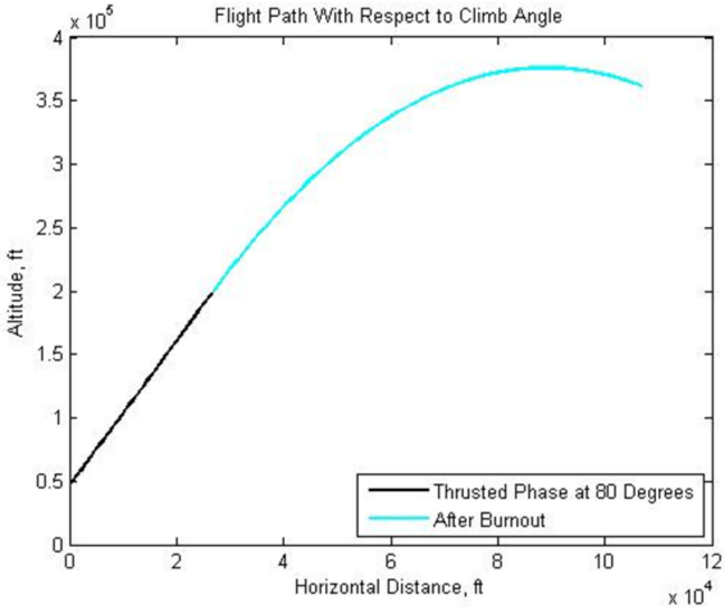


Figure 4.1: Flight Path Suborbital Air – Drop Mission

The balance field length, take – off velocity, and take – off length is not applicable to the air – drop mission. However, the rate of climb and descent are determined using the code "takeoffv2.m" and the following equations.

$$ROC = V*\sin(\gamma) \quad [12]$$

$$ROD = V*\sin(\gamma) \quad [12]$$

Therefore, the maximum rate of descent is 4500 ft/s, while the rate of climb is shown below in Fig. 4.2

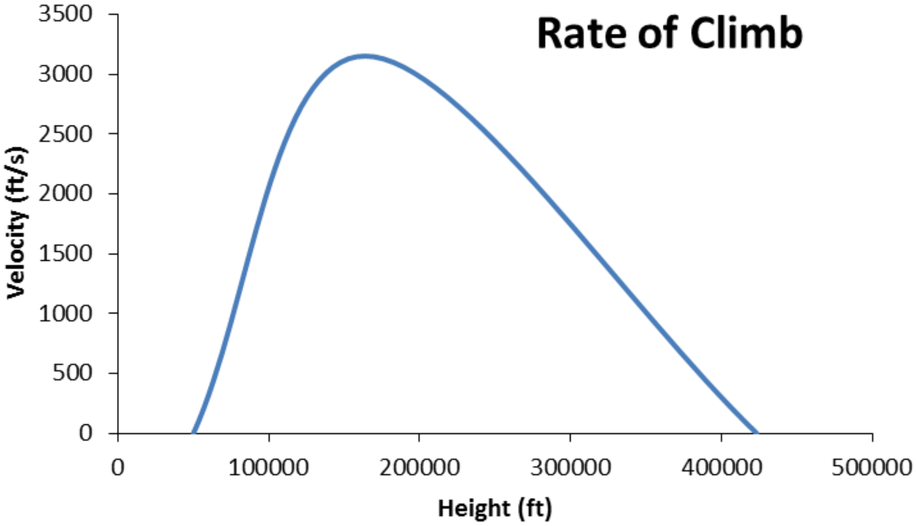


Figure 4.2: Rate of Climb Suborbital Air – Drop Mission

The landing distance is found using the equation below with the code "land2.m" from Appendix C as 2500 ft.

$$Sl_l = Sl_a + Sl_{fr} + Sl_{b2} \quad [12]$$

The landing gear is determined to be a tricycle landing gear by Ms. Sachya. Table 4.2 below shows further information on the landing gear.

Table 4.2: Landing Gear (Provided by Ms. Sachya)

Landing Gear				
Tricycle	Diameter (in)	Width (in)	Distance from nose (ft)	Load (lb)
Main Wheel	26	7.2	42	40,500
Nose Wheel	18	5.1	7	8600

The flight envelope is calculated using the following equations below.

$$V_{stall} = (W_{TO}/S*2/\rho/Clmax)^{1/2} \qquad [12]$$

$$V_2 = \left(2*\frac{q}{\rho}\right)^{1/2} \qquad [12]$$

$$V_{25} = \left(\frac{2*D}{S*\rho*\mathrm{Cd}}\right)^{1/2} \qquad [12]$$

$$V_4 = \left(\frac{qs}{3.21e-4*Cf*\rho}\right)^{1/3}*0.3048 \qquad [12]$$

Then the Mach number at each height is calculated using the following equation.

$$M = \frac{V}{a} \qquad [12]$$

Finally, the overall values are found in the code “envelopes42h.m” from Appendix C.

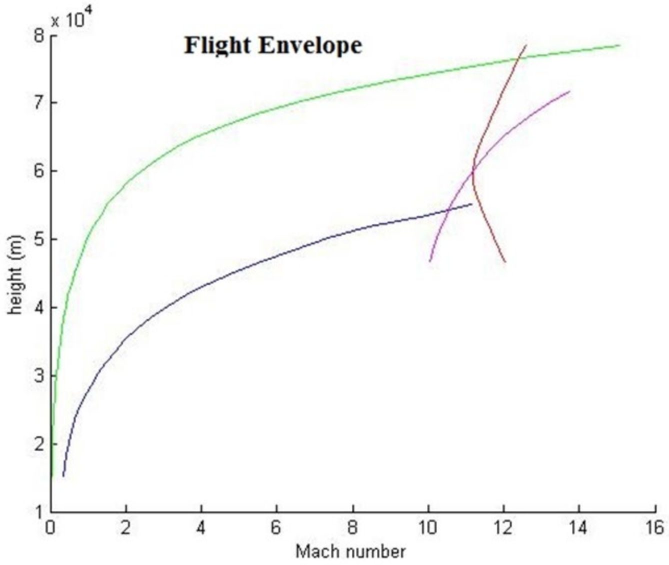


Figure 4.3: Flight Envelope Suborbital Air – Drop Mission

The vehicle operates once over 50,000 ft, so the envelope begins at that point. Furthermore, in order to complete the flight, the vehicle must decelerate in order to land safely. Therefore, banking turns in order to bleed off excess speed are called for. In order to determine the drag from these turns, and hence the decrease in velocity, the following equation is used in the code "turn2.m" in order to determine the banking angle and time necessary to decrease the speed.

$$D = q*S*(Cd+K_c*(n*W_f\,/\,(q*S))^2\,) + Cd_{trim} \qquad [14]$$

From this code, the bank angle is 5 degrees, with a time to bank of 476 s. Assuming a constant deceleration, the G-forces can be determined using the following equation.

$$A = (V_{re} - V_f)\,/\,t_{turn}$$

Once this value is divided by gravity, presuming gravity to be 32.2 ft/s2, the approximate g-forces exerted upon the passengers is 3g, which is less than the maximum allowed 3.8g specified by the FAA [18].

The thrust values will be sent to Propulsions, the landing gear will be sent to Synthesis, and the flight envelope, atmospheric conditions, maximum altitude, maximum speed, and re – entry values will be sent to Aerodynamics. Furthermore, the rate of climb, balance field length, range, and endurance and loiter values will be sent to the Costs and Certifications group in order to ensure safety and compliance.

4.2 Performance Analysis of Suborbital Horizontal Take – off Space Tourism Design

For the second mission, the X-15 will be redesigned for suborbital flight achieved from a horizontal take-off and landing, single stage, position. Preliminary research indicates the model for this design should be the Lynx by XCOR Aerospace as this is its flight path [7]. Furthermore, it possesses four XR-5K18 rocket engines powered by kerosene and liquid – oxygen, which would allow fast turnaround time due to the liquid fuel, allowing four flights a day if necessary [15]. It is composed completely of composites, and has a "thermal protection system" on the nose and leading edges in order to combat the heat from reentering the atmosphere [15]. Finally, it possesses a double – delta wing capable of landing at speeds of almost 90 knots [15].

However, the Lynx is only capable of ferrying one passenger to space, while our modified X-15 is required to ferry six [7]. Therefore, our current design for this mission is shown below in Fig. 4-5.

Figure 4.4: Suborbital Horizontal Take – off Mission Design
(provided courtesy of the Geometry Group)

The following analysis of the horizontal take – off mission is accomplished using the take – off weight, empty weight, and center of gravity values from the Geometry team, the coefficient of drag, maximum coefficient of lift, and angle from Aerodynamics, and the thrust available and specific impulse from the Propulsion group as inputs. Atmospheric conditions, as defined in Section 4.1, are applicable to both the air drop and horizontal take – off mission and will be used for both. The required thrust for the horizontal take-off is acquired from the code "Thrust2.m" located in Appendix C, created by Ms. Sachya, Mr. Faure, and I. This code is based upon the equations found in Ref. [14]. The equation used to determine thrust is shown below.

$$T_{req} = \frac{\Delta V}{\Delta t}(m_0) + D_0 + W_{TO}*\sin(\gamma_0)$$ [14]

After acquiring the required thrust, at 70,000 lbs due to a decrease in weight from Geometry, the burnout height, maximum height and burnout velocity are also found using “Thrust2.m” from Appendix C. This code uses the following equations to find the value of the maximum height and burnout velocity.

$$h_{bo} = c/m_e*(m_f*\log(m_f/m_0) + m_0 - m_f) - 1/2*((m_0-m_f)/me)^2*g$$ [14]

$$V_{bo} = c*(\log(n) - (g/me)*(m_0-m_f)$$ [14]

$$h_{max} = \tfrac{1}{2}*c^2/g*\ln^2(n) - c*m_0/me*(n*\ln(n) - (n-1))/n$$ [14]

This value is then compared to the flight requirement height of 100 km. If this value is not satisfied, the final velocity used to find the required thrust is increased, until the height is reached. At this point, the maximum height is reached, at 340,000 ft, and the thrust required is confirmed. The burnout velocity is 3500 ft/s, and the height at burnout is 140,000 ft. Additionally, in order to find the required performance values, the burnout time is calculated from the above code using the following equation and is found to be 100 seconds.

$$t_{bo} = (m_0-m_f)/m_e$$ [14]

The time in microgravity is found based upon the assumption that the re – entry height is set at 150,000 ft due to a Stability and Control requirement. Therefore, the following equation is used to find a microgravity time of 195 s. The equation below is used to find the speed after this time period using “Thrust2.m” from Appendix C.

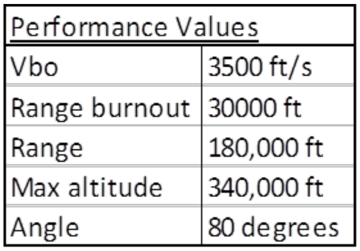
Consequently, the velocity at re – entry is found, using the equation shown below, to be 4500 ft/s.

$$t_{micro}= (2*(h_{max}-h_{re})/g_{re})^{1/2} \tag*{[14]}$$

$$V_{re} = V_{hmx} - g_{re}*(t_{micro}) \tag*{[14]}$$

After the results from the above code and the codes found in Appendix C “rcktperfdeltad.m” and “FlightPathAngle45dbld4.m” are compared, and adjusted until the maximum difference in value between the results is around 20%, the final values are accepted and averaged to produce the following table shown below. These differences in final values are primarily due to the different methods used in order to acquire them, and the assumptions made throughout.

Table 4.3: Performance Values Suborbital Horizontal Take – off Mission



Performance Values	
Vbo	3500 ft/s
Range burnout	30000 ft
Range	180,000 ft
Max altitude	340,000 ft
Angle	80 degrees

The range is found using the following equation in the code “FlightPathAngle45dbld4.m” and performing integration.

$$x^{*} = \frac{r}{r+h} *v*cos(\gamma) \tag*{[17]}$$

Finally, the flight path for the given thrust and velocity output by the code “FlightPathAngle45dbld4.m” is shown in the figure below.

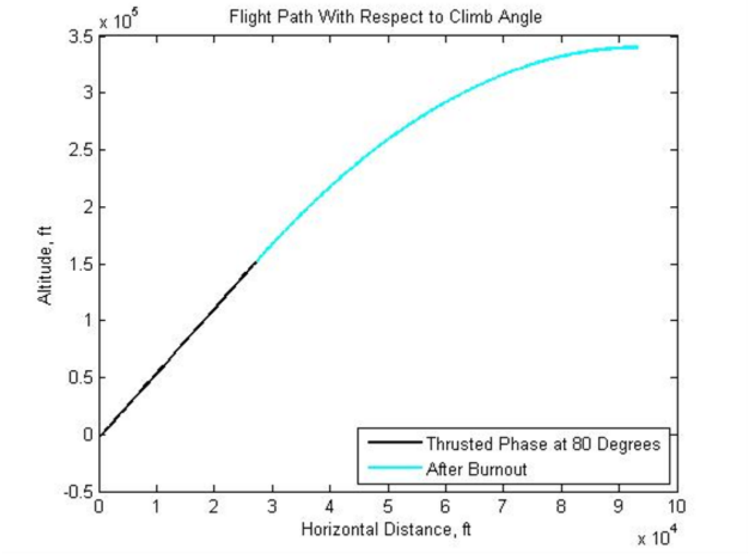


Figure 4.5: Flight Path Suborbital Horizontal Take – off Mission

Next, the balance field length is calculated by Mr. Faure of the Performance group using the following equation.

$$BFL = Sl_g + Sl_{r2} + Sl_{b2} \quad [12]$$

The code "balfield.m" as shown in Appendix C outputs a value of 1600 ft, due to the decrease in weight, as shown in Fig. 4.6 below.

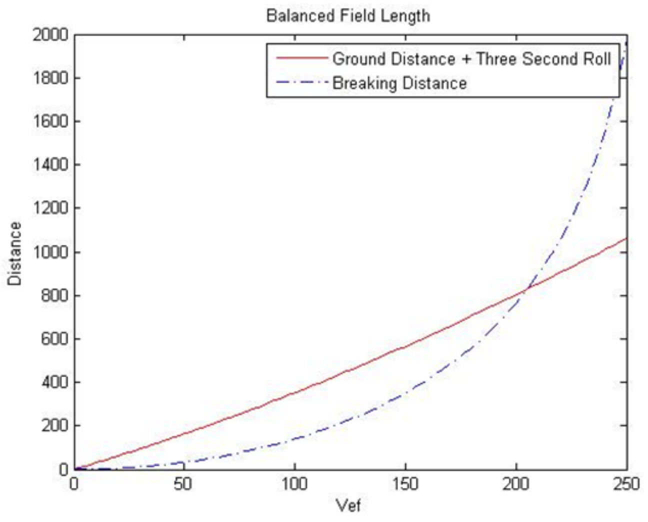


Figure 4.6: Balance Field Length (Provided by Mr. Faure)

The take – off velocity is determined to be 200 ft/s using the following equation.

$$V_{to} = 1.2*(W_0*2/S/\rho/Clmax)^{1/2} \quad [12]$$

The above, along with the rate of climb and rate of descent, using the following equations, are determined using the code "takeoffv2.m" as seen in Appendix C.

$$ROC = V*sin(\gamma) \quad [12]$$

$$ROD = V*sin(\gamma) \quad [12]$$

The maximum rate of descent is therefore determined to be 4500 ft/s, while the rate of climb is shown below in Fig. 4.7.

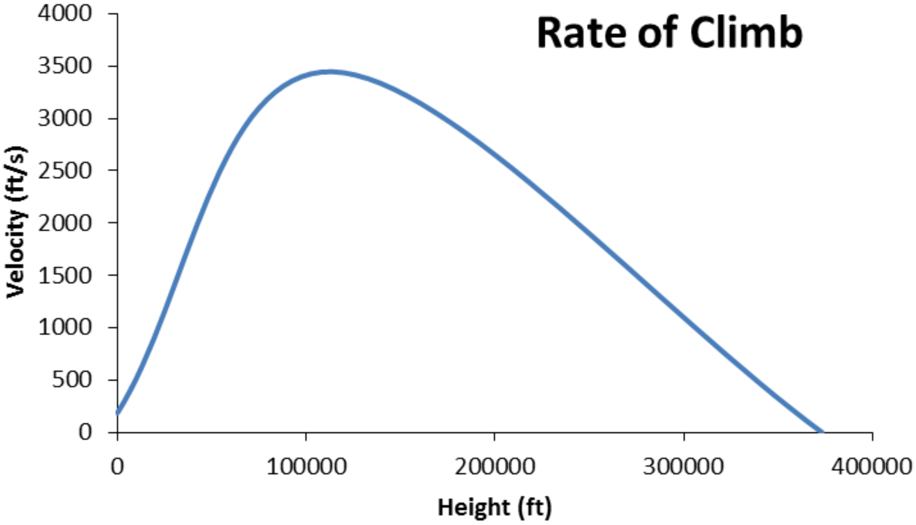


Figure 4.7: Rate of Climb Suborbital Horizontal Take – off Mission

Then the take – off field length is determined using the following equation in the code "takeofff.m" created by Mr. Faure from Appendix C.

$$Sl_{TO} = Sl_g + Sl_{r2} + Sl_{tr} + Sl_{cl} \quad [12]$$

Consequently, the takeoff distance is determined to be around 2600 ft. The landing distance is found to be about 2500 ft using the equation below in the code "land2.m" from Appendix C.

$$Sl_l=Sl_a+Sl_{fr}+Sl_{b2} \qquad [12]$$

The landing gear, as determined by Ms. Sachya of the Performance group, will be a tricycle landing gear. Table 4.2 above shows the landing gear information provided by Ms. Sachya.

The flight envelope is calculated using the equations below.

$$V_{stall} = (W_{TO}/S*2/\rho/Clmax)^{1/2} \qquad [12]$$

$$V_2 = \left(2*\frac{q}{\rho}\right)^{1/2} \qquad [12]$$

$$V_{25} = \left(\frac{2*D}{S*\rho*Cd}\right)^{1/2} \qquad [12]$$

$$V_4 = \left(\frac{qs}{3.21e-4*Cf*\rho}\right)^{1/3}*0.3048 \qquad [12]$$

Then the Mach number at each height is calculated using the following equation.

$$M = \frac{V}{a} \qquad [12]$$

Finally, the overall values are found in the code "envelopes42.m" from Appendix C and plotted in EXCEL to produce the figure below.

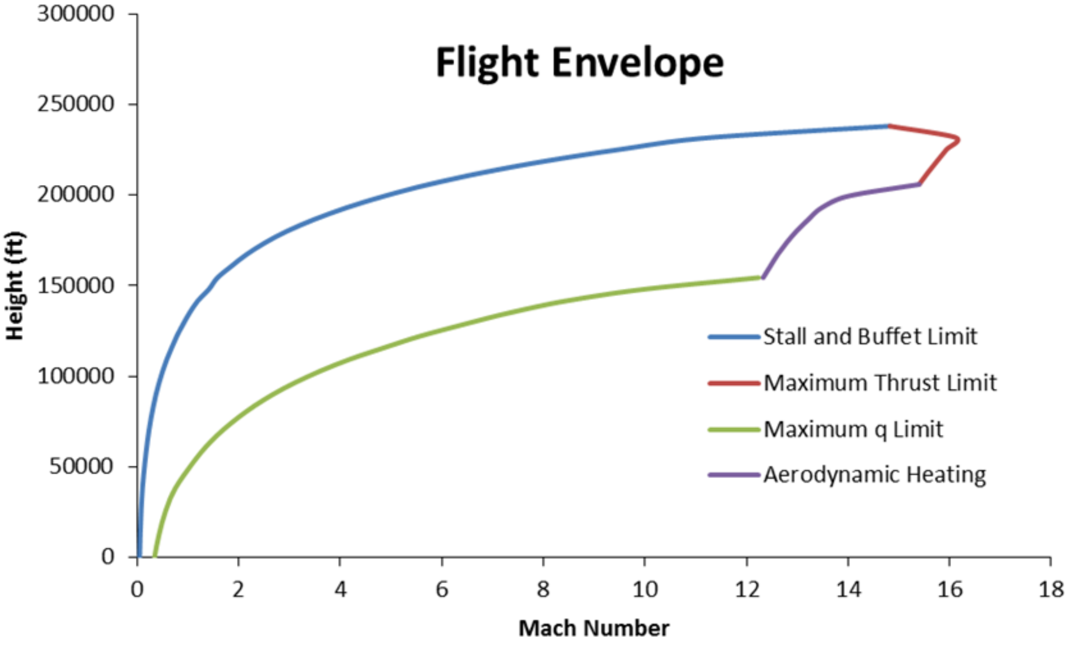


Figure 4.8: Flight Envelope Suborbital Horizontal Take – off Mission

The current maximum thrust and aerodynamic heating curves of the flight envelope are somewhat peculiar, and therefore further examination is called for. Furthermore, in order to complete the flight, the vehicle must decelerate in order to land safely. Therefore, banking turns in order to bleed off excess speed are called for. In order to determine the drag from these turns, and hence the decrease in velocity, the following equation is used in the code “turn2.m” in order to determine the banking angle and time necessary to decrease the speed.

$$D = q*S*(Cd+K_c*(n*W_f / (q*S))^2) + Cd_{trim} \qquad [14]$$

From this code, and a bank angle of 5 degrees, the time to bank is 476 s. Assuming a constant deceleration, the G-forces can be determined using the following equation.

$$A = (V_{re} - V_f) / t_{turn}$$

Once this value is divided by gravity, presuming gravity to be 32.2 ft/s^2, the approximate g-forces exerted upon the passengers is 3g that is less than the maximum allowed 3.8g specified by the FAA [18].

After completing the analysis, the thrust required values will be sent to the Propulsion group in order to verify the vehicle has the necessary thrust, while the landing gear information will be sent onto the Synthesis group for integration. Finally, the flight envelope, atmospheric conditions, maximum altitude, maximum speed, and re – entry values will be sent to aerodynamics in order to ensure feasibility, and the rate of climb, balance field length, range, and endurance and loiter values will be sent to the Costs and Certifications group in order to ensure they are within the necessary parameters.

4.3 Performance Analysis of Orbital Air – Drop Space Tourism Design

In order to fully analyze the vehicle, the orbital mission will also be examined. The inputs will once more come from the Geometry team, Aerodynamics, and Propulsions in the form of empty weight, air – drop weight, center of gravity, coefficient of drag, maximum coefficient of lift, angle, thrust available and specific impulse, with atmospheric conditions described from Section 4.4 above.

The above codes and equations can once more be used. As the vehicle is already sized, the thrust available is 70,000 lb. However, as the conditions for low earth orbit begins at around 460,000 ft, the burnout time must be increased for greater height [19]. Therefore, the burnout time is found to be 120 s from the code “Thrust2oh.m” from Appendix C which is increased until the maximum height is reached. The burnout height, velocity, and maximum height are also found from this code, using the equations shown below.

$$h_{bo} = 15239 + c/m_e*(m_f*log(m_f/m_0) + m_0 - m_f) - 1/2*((m_0-m_f)/me)^2*g \qquad [14]$$

$$V_{bo} = c*(log(n) - (g/me)*(m_0-m_f) \qquad [14]$$

$$h_{max} = 15239 + ½*c^2/g*ln^2(n) - c*m_0/me*(n*ln(n) - (n-1))/n \qquad [14]$$

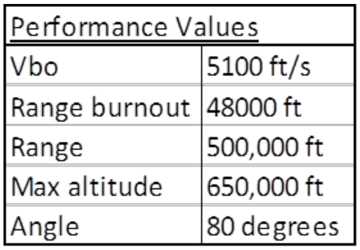
The maximum height reached is determined to be 650,000 ft, with a burnout velocity of 5100 ft/s and a burnout height of 296,000 ft. The re – entry altitude is set at 220,000 ft, which determines the time in microgravity. The code "Thrust2oh.m" is used to find the microgravity time and re – entry velocity using the equations below.

$$t_{micro} = (2*(h_{max}-h_{re})/g_{re})^{1/2} \qquad [14]$$

$$V_{re} = V_{hmx} - g_{re}*(t_{micro}) \qquad [14]$$

Consequently, the microgravity time is 400 s, while the re – entry velocity is 70,000 ft/s. The code "FlightPathAngle45dbld4ho.m" in Appendix C is used to find the following performance values.

Table 4.4: Performance Values Orbital Air - Drop Mission



Performance Values	
Vbo	5100 ft/s
Range burnout	48000 ft
Range	500,000 ft
Max altitude	650,000 ft
Angle	80 degrees

Furthermore, the range is also found using the above code with the following equation and integrating.

$$x^* = \frac{r}{r+h}*v*cos(\gamma) \qquad [17]$$

Flight path is also found with the code above, producing the figure shown below.

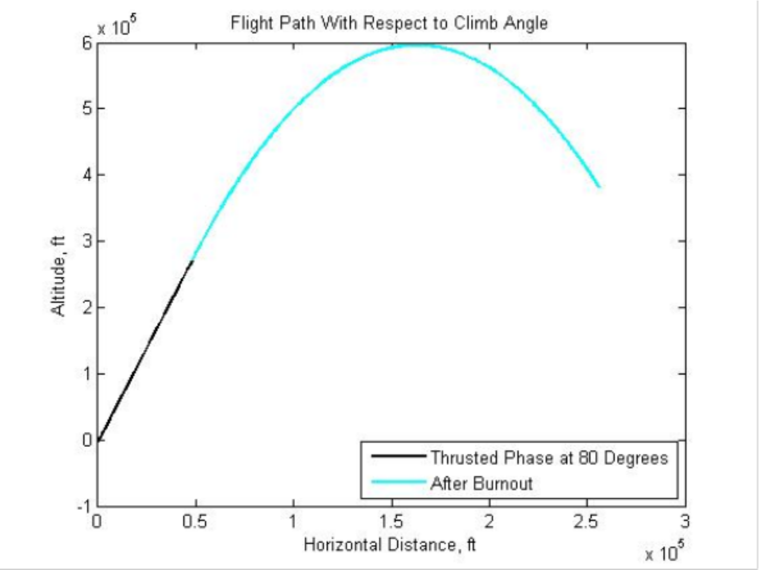


Figure 4.9: Flight Path Orbital Air – Drop Mission

As this is an air – drop mission, the balance field length, take – off velocity, and take – off length are not applicable. Consequently, the rate of climb and descent are determined using the code "takeoffv2.m" and the following equations.

$$ROC = V*\sin(\gamma) \quad [12]$$

$$ROD = V*\sin(\gamma) \quad [12]$$

Therefore, the maximum rate of descent is 7000 ft/s. The rate of climb is shown below in Fig. 4.10.

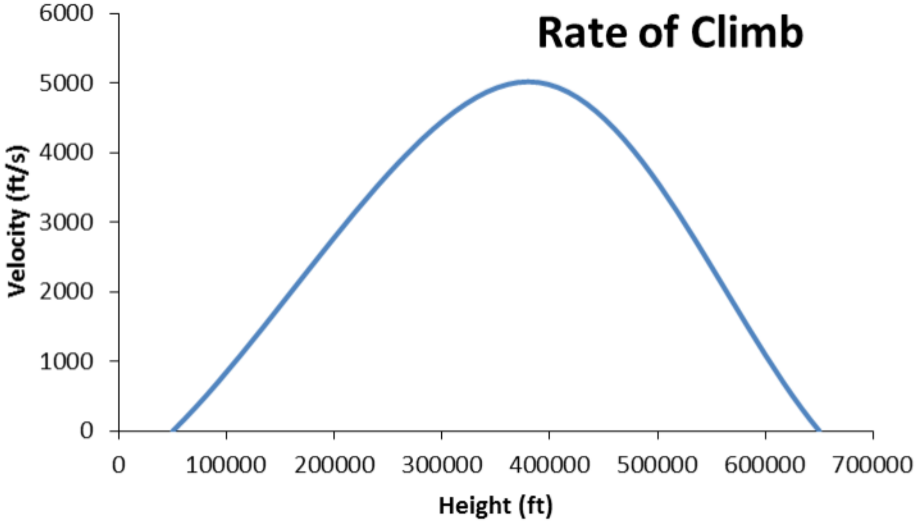


Figure 4.10: Rate of Climb Orbital Air – Drop Mission

The landing distance is found using the equation below with the code "land2.m" from Appendix C as 2500 ft.

$$Sl_l=Sl_a+Sl_{fr}+Sl_{b2} \quad [12]$$

The landing gear is determined to be a tricycle landing gear by Ms. Sachya. Table 4.2 above shows further information on the landing gear.

The flight envelope is once more calculated using the following equations below.

$$V_{stall} = (W_{TO}/S*2/\rho/Clmax)^{1/2} \quad [12]$$

$$V_2 = (2*\frac{q}{\rho})^{1/2} \quad [12]$$

$$V_{25} = (\frac{2*D}{S*\rho*Cd})^{1/2} \quad [12]$$

$$V_4 = (\frac{qs}{3.21e-4*Cf*\rho})^{1/3}*0.3048 \quad [12]$$

Then the Mach number at each height is calculated using the following equation.

$$M = \frac{V}{a} \quad [12]$$

Finally, the overall values are found in the code “envelopes42oh.m” from Appendix C.

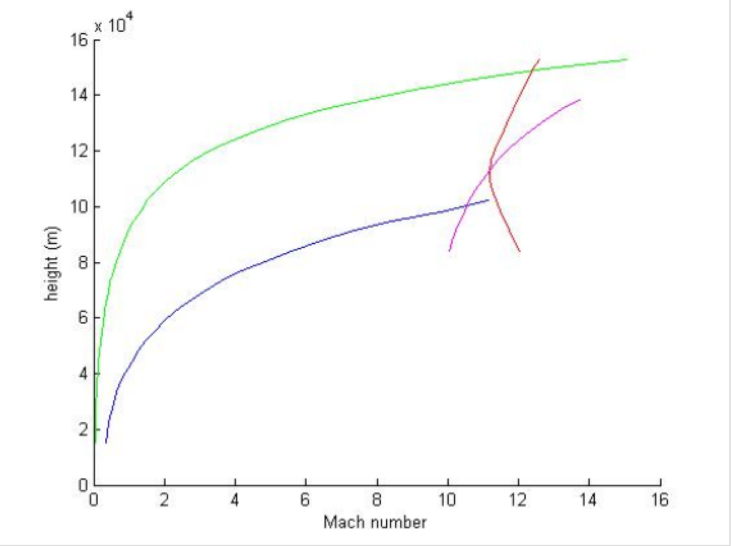


Figure 4.11: Flight Envelope Orbital Air – Drop Mission

The vehicle operates once over 50,000 ft, so the envelope begins at that point. Furthermore, in order to complete the flight, the vehicle must decelerate for a safe landing. Therefore, banking turns in order to bleed off speed are called for. In order to determine the drag and the decrease in velocity, the following equation is used in the code “turno2h.m” in order to determine the banking angle and time necessary to decrease the speed.

$$D = q*S*(Cd+K_c*(n*W_f / (q*S))^2) + Cd_{trim} \qquad [14]$$

From this code, and a bank angle of 30 degrees, the time to bank is 916 s. Assuming a constant deceleration, the G-forces can be determined using the following equation.

$$A = (V_{re} - V_f) / t_{turn}$$

Once this value is divided by gravity, presuming gravity to be 32.2 ft/s2, the approximate g-forces exerted upon the passengers is 2.3g, which is less than the maximum allowed 3.8g specified by the FAA [18].

These values would then be sent on to Propulsions, Synthesis, Aerodynamics, and Costs and Certifications in the form of thrust values, landing gear, flight envelope, atmospheric conditions, maximum altitude, maximum speed, re – entry values, rate of climb, range, endurance and loiter.

Furthermore, if we wished to remain in orbit, the vehicle must accelerate to the orbital velocity. These are determined using the following equations in the code "orbit2.m" from Appendix C.

$$h1 = (398600*r_{orb})^{1/2} \qquad [14]$$

$$v_{orb} = h1/r_{orb} \qquad [14]$$

Therefore, at this altitude, the vehicle must accelerate to 25,000 ft/s. Consequently more fuel is necessary, as determined by the following equation.

$$m_{ch} = (1 - exp\ (-v_{ch}\ /(Isp*g)))*m_0 \qquad [14]$$

Consequently, 380 lb of extra fuel would be needed on this mission in order to attain orbit. However, this might affect the overall flight path as this fuel will need to be brought up here as excess weight.

4.4 Performance Analysis of Orbital Horizontal Take – off Space Tourism Design

The next phase of orbital analysis is the horizontal take – off mission. Required input values will be acquired from Geometry, Aerodynamics, and Propulsion as take – off weight, empty weight, center of gravity, coefficient of drag and lift, angle, thrust available, and specific impulse. Atmospheric conditions are found using the form shown in Section 4.1 and the code “geom224.m”.

As the thrust and burn time have been determined to be 70,000 lb and 120 s, respectively, the burnout height, burnout velocity, and maximum height can be found using the following equations from “Thrust20.m” in Appendix C. These values are 246,000 ft, 5100 ft/s, and 600,000 ft.

$$h_{bo} = c/m_e*(m_f*log(m_f/m_0) + m_0 - m_f) - 1/2*((m_0-m_f)/me)^2*g$$ [14]

$$V_{bo} = c*(log(n) - (g/me)*(m_0-m_f)$$ [14]

$$h_{max} = \tfrac{1}{2}*c^2/g*ln^2(n) - c*m_0/me*(n*ln(n) - (n-1))/n$$ [14]

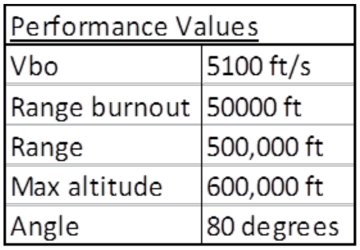
The microgravity time is found at a re – entry height of 220,000 ft, with the following equation used to find a value of 375 s. This and the re – entry speed from the equation below are also found using the above code. Therefore, the re – entry velocity is 6700 ft/s.

$$t_{micro}= (2*(h_{max}-h_{re})/g_{re})^{1/2}$$ [14]

$$V_{re} = V_{hmx} - g_{re}*(t_{micro})$$ [14]

Next, the code “FlightPathAngle45dbld4ho.m” from Appendix C is used to find the performance values shown in the table below.

Table 4.5: Performance Values Orbital Horizontal Take – off Mission



Performance Values	
Vbo	5100 ft/s
Range burnout	50000 ft
Range	500,000 ft
Max altitude	600,000 ft
Angle	80 degrees

Furthermore, the range is found using the following equation below in the code“FlightPathAngle45dbld4ho.m” and performing integration.

$$x^{*} = \frac{r}{r+h} * v * \cos(\gamma) \qquad [17]$$

The flight path is also found using the above code, producing the figure shown below.

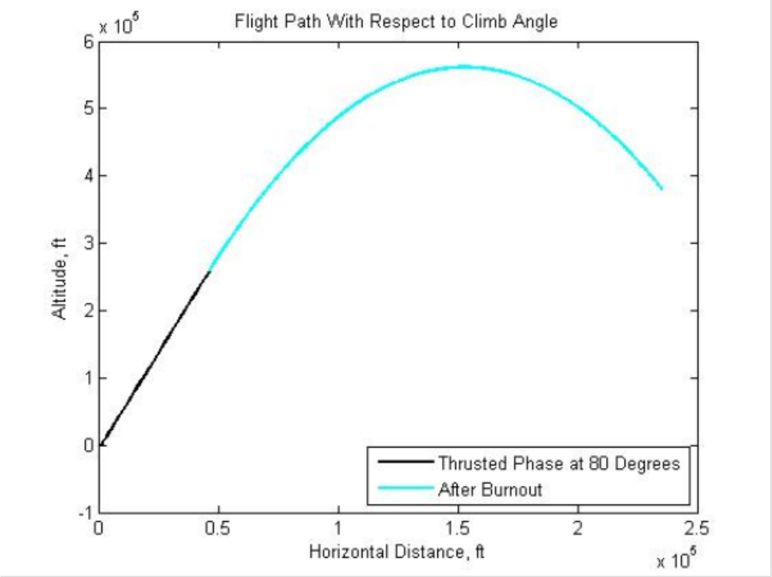


Figure 4.12: Flight Path Orbital Horizontal Take – off Mission

The balance field length is found using the following equation.

$$BFL = Sl_{g} + Sl_{r2} + Sl_{b2} \qquad [12]$$

The code “balfield.m” produced by Mr. Faure as shown in Appendix C outputs a value of 1600 ft, due to the decrease in weight, as shown in Fig. 4.13 below.

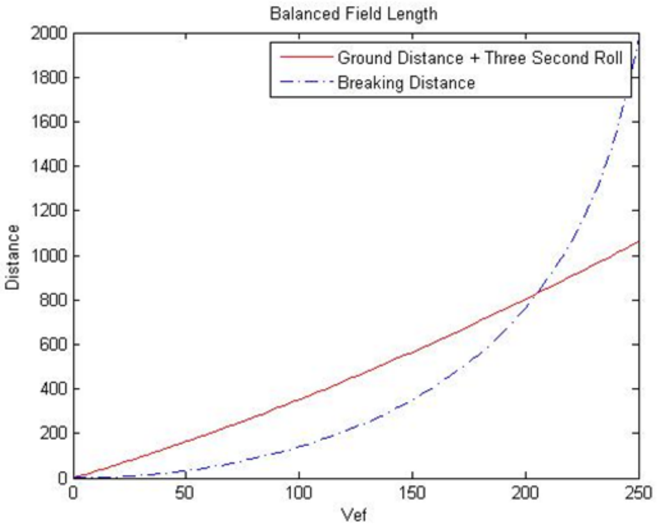


Figure 4.13: Balance Field Length Orbital (Provided by Mr. Faure)

Furthermore, the take – off velocity is found to be 200 ft/s using the following equation.

$$V_{to} = 1.2*(W_0*2/S/\rho/Clmax)^{1/2} \qquad [12]$$

The above, rate of climb and rate of descent, are found using the following equations, in the code “takeoffv2.m” as seen in Appendix C.

$$ROC = V*sin(\gamma) \qquad [12]$$

$$ROD = V*sin(\gamma) \qquad [12]$$

Therefore, the maximum rate of descent is 6700 ft/s, while the rate of climb is seen in the following Fig. 4.14.

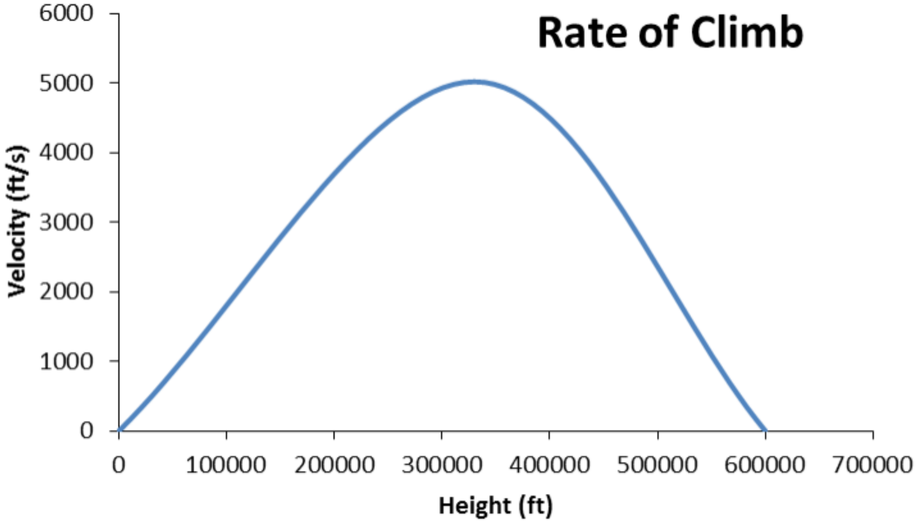


Figure 4.14: Rate of Climb Orbital Horizontal Take – off Mission

The take – off field length is once more determined using the following equation in the code "takeofff.m" created by Mr. Faure from Appendix C to be 2600 ft.

$$Sl_{TO} = Sl_g + Sl_{r2} + Sl_{tr} + Sl_{cl} \quad [12]$$

Also, the landing distance is found to be 2500 ft using the equation below and the code "land2.m" from Appendix C.

$$Sl_l = Sl_a + Sl_{fr} + Sl_{b2} \quad [12]$$

The vehicle will possess a tricycle landing gear. Table 4.2 above shows the landing gear information provided by Ms. Sachya.

The flight envelope is also calculated using the equations below.

$$V_{stall} = (W_{TO}/S*2/\rho/Clmax)^{1/2} \quad [12]$$

$$V_2 = (2*\frac{q}{\rho})^{1/2} \quad [12]$$

$$V_{25} = (\frac{2*D}{S*\rho*Cd})^{1/2} \quad [12]$$

$$V_4 = (\frac{qs}{3.21e-4*Cf*\rho})^{1/3}*0.3048 \quad [12]$$

Then the Mach number at each height is calculated using the following equation.

$$M = \frac{V}{a} \qquad [12]$$

Finally, the overall values are found in the code "envelopes42o.m" from Appendix C and shown in the figure below.

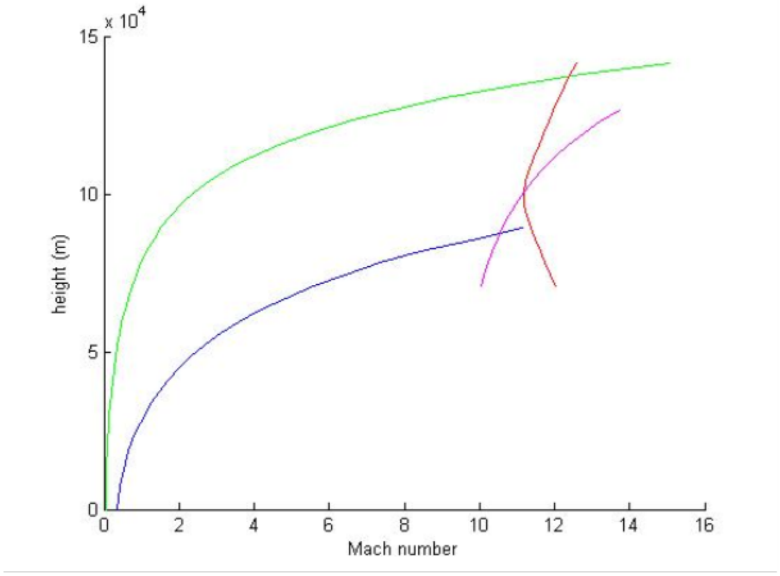


Figure 4.15: Flight Envelope Orbital Horizontal Take – off Mission

Deceleration is necessary in order to land safely. Banking turns in order to accomplish this are calculated using the following equations in the code "turno2.m" as shown in Appendix C.

$$D = q*S*(Cd+K_c*(n*W_f / (q*S))^2) + Cd_{trim} \qquad [14]$$

Therefore, the bank angle is 30 degrees, and the time to bank is 882 s. Assuming a constant deceleration, the G-forces can be determined using the following equation.

$$A = (V_{re} – V_f) / t_{turn}$$

Once this value is divided by gravity, presuming gravity to be 32.2 ft/s2, the approximate g-forces exerted upon the passengers is 2.4g that is less than the maximum allowed 3.8g specified by the FAA [18].

These completed analysis values will then be sent on to Propulsions, Synthesis, Aerodynamics, and Costs and Certifications in the form of the following values: thrust required, landing gear information, flight envelope, atmospheric conditions, maximum altitude, maximum speed, re – entry values, rate of climb, balance field length, range, and endurance and loiter values. This will ensure the vehicle remains within constraints.

However, in order to remain in orbit, the vehicle much reach a certain velocity, which is found using the following equations in the code "orbit.m" in Appendix C.

$$h1 = (398600*r_{orb})^{1/2} \qquad [14]$$

$$v_{orb} = h1/r_{orb} \qquad [14]$$

Therefore, the vehicle must accelerate to 25,000 ft/s. The fuel required to do this is found using the following equation in the same code as above.

$$m_{ch} = (1 - exp\ (-v_{ch}\ /(Isp*g)))*m_0 \qquad [14]$$

Consequently, 306 lb of extra fuel would be needed on this mission in order to attain orbit. However, this might affect the overall flight path as this fuel will need to be brought up there as excess weight.

CHAPTER 5

CONCLUSION

5.1 Results and Discussion

The results from the performance analysis performed above in Sections 4.1– 4.4 are clearly seen in Table 5.1 and Table 5.2. In Table 5.1, Run 1 is performed with initial assumptions and estimations. The changes from Run 1 to Run 2 are the result of a decrease in weight as well as code modifications. Run 3 is a result of estimated values being replaced by actual supplied values from other groups, as well as further code modifications and corrections, while Run 4 is performed with Stability and Control's desire to use aerodynamic controls, which requires a lower re – entry height of 150,000 ft, and further code corrections and new supplied values. This is the final run for the performance analysis of horizontal take – off suborbital mission analysis. Run 5 differs from Run 4 in that Run 5 is the first iteration results from the air – drop suborbital mission. Run 6 is the final performance results incorporating the 150,000 ft re – entry condition. Consequently, this vehicle is assumed to be dropped at 50,000 ft by another aircraft, from which it accelerates to the maximum height. Since the burn time is decreased for the air drop mission by 5 s, but the thrust is constant, approximately 1,000 lb of fuel is saved, decreasing the cost per trip. This leads to the same re – entry conditions for both missions, thereby requiring the same drag forces and G – forces for deceleration to a safe landing speed. However, both the suborbital horizontal take – off and air – drop missions will reach the required altitude in a reasonable amount of time with less than 3.8g.

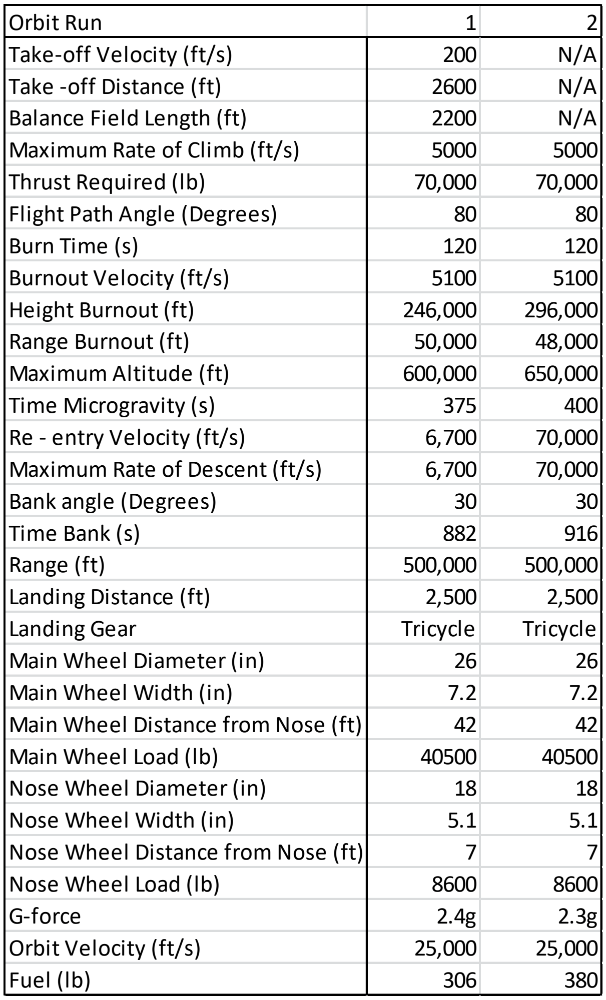
Table 5.1: Performance Suborbital Results

Run	1	2	3	4	5	6
Take-off Velocity (ft/s)	66	199	200	200	N/A	N/A
Take -off Distance (ft)	33	6700	980	2600	N/A	N/A
Balance Field Length (ft)	2485	2200	2200	1600	N/A	N/A
Maximum Rate of Climb (ft/s)	3250	2500	3500	3500	3500	3200
Thrust Required (lb)	107,200	70,000	70,000	70,000	70,000	70,000
Flight Path Angle (Degrees)	45	45	80	80	80	80
Burn Time (s)	60.5	80	100	100	100	95
Burnout Velocity (ft/s)	4662	3600	3500	3500	3500	3200
Height Burnout (ft)	100,000	110,000	140,000	140,000	190,000	170,000
Range Burnout (ft)	60170	89356	30000	30000	27000	27000
Maximum Altitude (ft)	373,200	330,000	340,000	340,000	380,000	340,000
Time Microgravity (s)	300	300	170	195	220	195
Re - entry Velocity (ft/s)	2180	3,400	3,700	4,500	4,500	4,500
Maximum Rate of Descent (ft/s)	1540	2,404	3608	4500	4500	4500
Bank angle (Degrees)	N/A	N/A	5	5	30	5
Time Bank (s)	N/A	N/A	271	476	680	476
Range (ft)	650,000	600,000	180,000	180,000	240,000	180,000
Landing Distance (ft)	26,000	1,700	2,500	2,500	2,500	2,500
Landing Gear	Tricycle	Tricycle	Tricycle	Tricycle	Tricycle	Tricycle
Main Wheel Diameter (in)	26	26	26	26	26	26
Main Wheel Width (in)	7.2	7.2	7.2	7.2	7.2	7.2
Main Wheel Distance from Nose (ft)	42	42	42	42	42	42
Main Wheel Load (lb)	40,500	40,500	40,500	40,500	40,500	40,500
Nose Wheel Diameter (in)	18	18	18	18	18	18
Nose Wheel Width (in)	5.1	5.1	5.1	5.1	5.1	5.1
Nose Wheel Distance from Nose (ft)	7	7	7	7	7	7
Nose Wheel Load (lb)	8600	8600	8600	8600	8600	8600
G-force	3.4g	2.4g	3.7g	3g	2.4g	3g

Table 5.2 below showcases the orbital performance results. Run 1 is for the horizontal take – off mission stage, while Run 2 is the air – drop mission. As seen in Table 5.2, the thrust is the same as that for the suborbital missions of Table 5.1, but the burnout time is increased from 100 s to 120 s in order to reach the low earth orbit. As with the suborbital mission, the air drop mission reaches greater heights when using the same burn time and achieves a longer microgravity time than the horizontal take – off mission due to the extra 50,000 ft supplied by the carrier aircraft. Interestingly enough,

the G – forces for the orbital mission are lower than those for the suborbital mission. This might imply that the relationship between deceleration and time to decelerate is logarithmic in nature, with sharp changes giving way to a gentle increase. Possibly leading to the faster re – entry yields a lower g – force seen below. Finally, although the vehicles reach slightly different altitudes requiring similar speeds, the fuel required by the air drop mission is greater than that for the horizontal take – off mission, implying a greater speed is required. However, both the horizontal and air drop orbital mission pass the low earth orbit range of 460,000 ft.

Table 5.2: Orbital Performance Results



Orbit Run	1	2
Take-off Velocity (ft/s)	200	N/A
Take -off Distance (ft)	2600	N/A
Balance Field Length (ft)	2200	N/A
Maximum Rate of Climb (ft/s)	5000	5000
Thrust Required (lb)	70,000	70,000
Flight Path Angle (Degrees)	80	80
Burn Time (s)	120	120
Burnout Velocity (ft/s)	5100	5100
Height Burnout (ft)	246,000	296,000
Range Burnout (ft)	50,000	48,000
Maximum Altitude (ft)	600,000	650,000
Time Microgravity (s)	375	400
Re - entry Velocity (ft/s)	6,700	70,000
Maximum Rate of Descent (ft/s)	6,700	70,000
Bank angle (Degrees)	30	30
Time Bank (s)	882	916
Range (ft)	500,000	500,000
Landing Distance (ft)	2,500	2,500
Landing Gear	Tricycle	Tricycle
Main Wheel Diameter (in)	26	26
Main Wheel Width (in)	7.2	7.2
Main Wheel Distance from Nose (ft)	42	42
Main Wheel Load (lb)	40500	40500
Nose Wheel Diameter (in)	18	18
Nose Wheel Width (in)	5.1	5.1
Nose Wheel Distance from Nose (ft)	7	7
Nose Wheel Load (lb)	8600	8600
G-force	2.4g	2.3g
Orbit Velocity (ft/s)	25,000	25,000
Fuel (lb)	306	380

Therefore, the vehicle reaches the flight conditions and maximum heights required for both the suborbital and orbital missions. These results are acquired based upon the inputs provided by Geometry, Aerodynamics, and Propulsion, and then sent off to the required teams, specifically Propulsion, Aerodynamics, Synthesis, and Costs and Certifications, in order to complete the vehicle analysis.

5.2 Conclusions and Recommendations

From the information shown above, the vehicle design for the horizontal take – off and air – drop suborbital and orbital missions is feasible. Some concerns and further analysis is necessary on the orbital speeds as compared to fuel required for these maneuvers. However, other than this issue, the horizontal design shown in Section 4-5 is sound and can be used for commercial purposes.

During our team's discussions, the designs changed somewhat over the course of the analysis. In the future, a recommendation would be that a design be locked in earlier and remain largely unchanged throughout, barring major flight issues. Furthermore, as work was begun upon the air drop mission prior to the horizontal take – off mission, with our emphasis changing midpoint, it would be recommended that we had begun analyzing the horizontal take – off mission prior to the air drop from the very beginning.

APPENDIX A

NOMENCLATURE

Variables

Δ	Change in
A	Acceleration
a	Speed of sound
δ	Pressure ratio
BFL	Balance Field Length
c	Exhaust velocity
Cd	Coefficient of Drag
Cf	Coefficient of Friction
Cl	Coefficient of Lift
D	Drag
e	Exponent
ft	Feet
g	Gravity
H	Geopotential height (m)
h	Height
h1	Angular momentum
Isp	Specific Impulse
K	Slope
M	Molecular weight of air (28.96 kg/kmol)
m	mass
n	Loading ratio
θ	Temperature ratio
σ	Density ratio

Subscripts

0	Sea level value
0	Initial values
2	At maximum dynamic pressure
25	At maximum thrust limit
4	At aerodynamic heating
*	Derivative
a	Air
b	Base value of the region considered
b2	Brake
bo	Burn-out
c	From Cl vs. Cd curve
ch	Change in
cl	Climb
coast	Climbing, no thrust
e	Exhaust mass flow rate
f	Final values
fr	Free roll
g	Ground
hmx	At maximum height
i	The region considered
l	Land
max	Maximum
orb	Orbital

∅	Viscosity ratio
P	Pressure (Pa)
ρ	Density (kg/m^3)
q	Maximum dynamic pressure
qs	Surface heat transfer
r	Radius of the Earth (6356.766 km)
R^*	Universal gas constant (8314.32 Nm/kmol/K)
ROC	Rate of Climb
ROD	Rate of Descent
S	Sutherland’s constant (110.4 K)
Sl	Distance
T	Temperature (K)
T	Thrust
t	time
V	Velocity
W	Weight
x	Range
χ	Speed of sound ratio
γ	Flight path angle
z	Geometric height (m)
	Values [13]

r	Required
r2	Rotation
re	Re – entry values
stall	Stall
T	Of temperature change
TO	Take-off
tr	Transition
trim	From Trim
turn	From turning
ybo	Burnout value on the y–axis

APPENDIX B

SCHEDULING

Entire Team: Our team’s overall schedule is to fulfill analysis of the horizontal takeoff/landing single stage mission by April, with the redesign for the air drop analyzed after this design is completed.

APPENDIX C

MATLAB CODES

Code geom224.m is to be used to determine atmospheric coefficients.

```
function [x,y,c,a,b]=geom224(z)
h=(6356.766e3/(6356.766e3+z))*z;

if(h<11000)
x=(1-2.2557e-5*h)^5.25644
y=(1-2.2557e-5*h)^4.25644
c=(1-2.2557e-5*h)
a=(1-2.2557e-5*h)*(1-1.6309e-5*h)^-1
b=sqrt(c)*340.2612

elseif(11000<=h)(h<20000)
x=0.2234*exp(1.73467-0.157697e-3*h)
y=0.297*exp(1.73467-0.157697e-3*h)
c=0.75187
a=0.79448
b=sqrt(c)*340.2612

elseif(20e3<=h)(h<32e3)
x1=0.2234*exp(1.73467-0.157697e-3*20e3)
y1=0.297*exp(1.73467-0.157697e-3*20e3)
x=x1*(1+1e-3/216.65*(h-20e3))^(-34.1698/1)
y=y1*(1+1e-3/216.65*(h-20e3))^(-34.1698/1-1)
c=216.65/288.15*(1+1e-3/216.65*(h-20e3))
a=c*(1+1e-3/(288.15+110.4)*(h-20e3))^-1
b=sqrt(c)*340.2612

elseif(32e3<=h)(h<47e3)
x1=0.2234*exp(1.73467-0.157697e-3*20e3)
y1=0.297*exp(1.73467-0.157697e-3*20e3)
x2=x1*(1+1e-3/216.65*(32e3-20e3))^(-34.1698/1)
y2=y1*(1+1e-3/216.65*(32e3-20e3))^(-34.1698/1-1)
x=x2*(1+2.8e-3/228.65*(h-32e3))^(-34.1698/2.8)
y=y2*(1+2.8e-3/228.65*(h-32e3))^(-34.1698/2.8-1)
c=228.65/288.15*(1+2.8e-3/228.65*(h-32e3))
a=c*(1+2.8e-3/(288.15+110.4)*(h-32e3))^-1
b=sqrt(c)*340.2612

elseif(47e3<=h)(h<52e3)
x1=0.2234*exp(1.73467-0.157697e-3*20e3)
x2=x1*(1+1e-3/216.65*(32e3-20e3))^(-34.1698/1)
x3=x2*(1+2.8e-3/228.65*(47e3-32e3))^(-34.1698/2.8)
x=x3*exp(-0.0341698/270.65*(h-47e3))
c=270.65/288.15
y=x/c
a=c^1.5*((288.15+110.4)/(270.65+110.4))
b=sqrt(c)*340.2612

elseif(52e3<=h)(h<61e3)
x1=0.2234*exp(1.73467-0.157697e-3*20e3)
x2=x1*(1+1e-3/216.65*(32e3-20e3))^(-34.1698/1)
x3=x2*(1+2.8e-3/228.65*(47e3-32e3))^(-34.1698/2.8)
x4=x3*exp(-0.0341698/270.65*(52e3-47e3))
c1=270.65/288.15
y4=x4/c1
x=x4*(1-2.0e-3/270.65*(h-52e3))^(-34.1698/-2.0)
y=y4*(1-2.0e-3/270.65*(h-52e3))^(-34.1698/-2.0-1)
c=270.65/288.15*(1-2.0e-3/270.65*(h-52e3))
a=c*(1-2.0e-3/(288.15+110.4)*(h-52e3))^-1
b=sqrt(c)*340.2612
```

```
elseif(61e3<=h)(h<79e3)
x1=0.2234*exp(1.73467-0.157697e-3*20e3)
x2=x1*(1+1e-3/216.65*(32e3-20e3))^(-34.1698/1)
x3=x2*(1+2.8e-3/228.65*(47e3-32e3))^(-34.1698/2.8)
x4=x3*exp(-0.0341698/270.65*(52e3-47e3))
c1=270.65/288.15
y4=x4/c1
x5=x4*(1-2.0e-3/270.65*(61e3-52e3))^(-34.1698/-2.0)
y5=y4*(1-2.0e-3/270.65*(61e3-52e3))^(-34.1698/-2.0-1)
x=x5*(1-4.0e-3/252.65*(h-61e3))^(-34.1698/-4.0)
y=y5*(1-4.0e-3/252.65*(h-61e3))^(-34.1698/-4.0-1)
c=252.65/288.15*(1-4.0e-3/252.65*(h-61e3))
a=c*(1-4.0e-3/(288.15+110.4)*(h-61e3))^-1
b=sqrt(c)*340.2612

else
x1=0.2234*exp(1.73467-0.157697e-3*20e3)
x2=x1*(1+1e-3/216.65*(32e3-20e3))^(-34.1698/1)
x3=x2*(1+2.8e-3/228.65*(47e3-32e3))^(-34.1698/2.8)
x4=x3*exp(-0.0341698/270.65*(52e3-47e3))
x5=x4*(1-2.0e-3/270.65*(61e3-52e3))^(-34.1698/-2.0)
x6=x5*(1-4.0e-3/252.65*(79e3-61e3))^(-34.1698/-4.0)
x=x6*exp(-0.0341698/180.65*(h-79e3))
c=180.65/288.15
y=x/c
a=(c)^(3/2)*((288.15+110.4)/(180.65+110.4))
b=sqrt(c)*340.2612

end
pressureratio=x
densityratio=y
temperatureratio=c
viscosityratio=a
speedofsound=b
end.
```

Code Thrust2.m, created by Ms. Sachya, Mr. Faure, and I is used to find the Thrust required [14].

```
% Velocity at 123000 ft
% The thrust calculation for the s

M = 41612.3*0.45;       %mass in kg
g = 9.8;        %m/s^2
theta = 90;     %flight path angle
C_d = 0.03;     %co-efficient of delta
rho = 5.19e-3;      %density in the altitude
V= 940;         %                 m/s
A = 81;                                 %area of the wing m^2

Isp=320;   %s

   %kg/s

mf = 21612.3*0.45;   %kg

tbo=100
t = tbo;     %burn time (designed)
me=(M-mf)/t   %kg/s

F = M*g*sind(0)+C_d*rho*V*V*A/2 + M*V/t;     %Force in newtons
F_lbs =F *0.2248                          %Thrust in lbs
```

```
F2 = M*g*sind(45)+C_d*rho*V*V*A/2 + M*V/t;     %Force in newtons
F_lbs2 =F2 *0.2248                          %Thrust in lbs

F25 = (M-mf)/2*g*sind(80)+C_d*rho*V*V*A/2 + (M-mf)/2*V/t;
F_lbs25 =F25 *0.2248

c = Isp*g

vbo = (c*log(M/mf) - g/me*(M-mf))
hbo = (c/me*(mf*log(mf/M) + M - mf)-1/2*((M-mf)/me)^2*g)

tcoast=c/g*log(M/mf)-tbo  %s

n=M/mf

hmax= (1/2*c^2/g*((log(n))^2)-c*M/me*(n*log(n)-(n-1))/n)

re=6378e3

gB = g/(1+hmax/re)^2
hA =90000 %m
h2=90000
gc = g/(1+hA/re)^2

tmicro= 2*sqrt(2*(hmax-hA)/gB)
tmicro2 = sqrt(2*(hA-h2)/gc)
tmicrototal = tmicro+tmicro2

vre1 = -gB*(tmicro/2+tmicro2)

h4=45720
gd = g/(1+h4/re)^2
tmicro4 = sqrt(2*(h2-h4)/gc)
D = C_d*rho*vre1*vre1*A/2
a = (D - mf*gd*sind(theta))/mf

vre = vre1 +a*tmicro4


hre = hmax - 1/2*gB*(tmicro2)^2

tmicrototal = tmicro+tmicro2+tmicro4

Wf = 21612.3*4.448   %N

D = C_d*rho*vre*vre*A/2
a = (D - Wf*sind(theta))/mf

AR=2.4
K=1/pi/AR
```

Code rcktperfdeltad.m is used to find the first iteration performance values [17].

```
%%%%%%%%%%Rocket Performance Calculations%%%%%%%%%%%%%%%%

%Givens:
w0 = 41612.3;        % Launch Weight [lbs]
wl = 21612.3;       % Landing Weight [lbs]
wf = w0-wl;         % Fuel Weight [lbs]
wp = 0;             % Payload Weight [lbs]
```

```
g0 = 32.174;        %[ft/s^2]
rE = 6378*3280.84;  %[ft]
G = 6.6742e-11;     % Universal Constant [m^3/kg*s^2]
Isp = 320;          % Specific Impulse at 47000 ft [sec]
Ispsl = 276;        % Specific Impulse at sea-level [sec]
T = 70000;          % Thrust for XLR-99 at 328084 ft  [lbs]
Tsl = 92000;        % Thrust for XLR-99 at sea-level  [lbs]
z0 = 0;             % Launch elevation
t0 = 0;             % [sec]
tf = 1800;          % [sec]
z1=0;

%Calculations:
m0 = w0/32.174;     % [slugs]
mf = wf/32.174;     % [slugs]
ml = wl/32.174;     % [slugs]
mp = wp/32.174;     % [slugs]

g = g0/((1+(z1/rE))^2);      % gravitational force at specific altitude [ft/s^2]
z1 = z0-((g/2)*(t1^2));    % ignition elevation [ft]

c = Isp*g;          % exhaust velocity [ft/sec]
n = m0/mf;                  % loading ratio

Dme = T/(Isp*g);            % exhaust mass flow rate [slug/s]
tbo = (m0-mf)/Dme;          % burnout time [sec]

hbo = (c/Dme)*((mf*log(mf/m0))+m0-mf)-(((1/2)*(((m0-mf)/Dme))^2)*g);    %burnout height [ft]

vbo = (c*(log(n)))-((g/Dme)*(m0-mf));                              % burnout velocity [ft/s]

hmax = ((1/2)*((c^2)/g)*((log(n))^2))-(((c*m0)/Dme)*(((n*log(n))-(n-1))/n)); %max altitude [ft]

fprintf('Exhaust Velocity = %.2f [ft/s]\n', c); %Display the initial guess
fprintf('Fuel Mass = %.2f [slugs]\n', mf); %Display the initial guess
fprintf('Exhaust Mass Flow Rate = %.2f [slugs/sec]\n', Dme); %Display the initial guess
fprintf('Burnout Time = %.1f [sec]\n', tbo); %Display the initial guess
fprintf('Burnout Height = %.2f [ft]\n', hbo); %Display the initial guess
fprintf('Burnout Velocity = %.2f [ft/s]\n', vbo); %Display the initial guess
fprintf('Max Height = %.2f [ft]\n', hmax); %Display the initial guess
```

Code FlightPathAngle45dbld4.m is used to determine the Performance values of the design [17].

```
%%%%%%%%%Rocket Performance Calculations%%%%%%%%%%%%%%%
function FlightPathAngle45dbld

%Givens:
w0 = 41612.3;        % Launch Weight [lbs]
wl = 21612.3;       % Landing Weight [lbs]
wf = w0 - wl;       % Fuel Weight [lbs]
S  = 00;            % Extra Fuel for Commercial W/D=9900,  Wo/D=0
w0 = w0 + S;        % Launch Weight accounting for added fuel
wf = wf + S;        % Fuel Weight with addition for Commercial
wp = 0;             % Payload Weight [lbs]
g0 = 32.174;        %[ft/s^2]
rE = 6378*3280.84;  %[ft]
% Isp = 284;          % Specific Impulse at 45000 ft for XLR-99 Engine  [sec]
T  = 70000;     % Thrust for XLR-99 at 45000 ft  [lbs]
h0 = 0;         % Launch elevation [ft]
x0 = 0;            % Initial Distance [ft]
V0 = 0;           % Initial Velocity [ft/s]
```

```
t1 = 15;            % [sec]
h  = 0;         % Ignition Elevation [ft]
gamma = 80;         % Flight Path Angle (assume constant) [degrees]
gammar = gamma*pi/180; % Flight Path Angle [radians]
D  = w0*cosd(gamma)/9.1;          % Drag Force [lbs]

hE = 360892;
gE = g0/((1+(hE/rE))^2);
g = g0/((1+(h/rE))^2);    % gravitational force at specific altitude [ft/s^2]
hI = h0-((g/2)*(t1^2));   % ignition elevation [ft]

%Calculations:
m0 = w0/32.174;     % Launch Mass[slugs]
mf = wf/32.174;     % Fuel Mass[slugs]  ***Replaced by later value***
ml = wl/32.174;     % Landing (Empty) Mass[slugs]
mp = wp/32.174;     % Payload Mass[slugs]

%Dme = T/(Isp*g);    % exhaust mass flow rate [slug/s]
tbo1 = 100;       % Burn Time from rcktperfdlbd [s]
Dme = (m0-ml)/tbo1; % Mass Flow Rate Defined by Tbo
Isp = T/(g*Dme);    % Specific Impulse Defined by Tbo and T
%mf = -tbo1*Dme+m0;  % Fuel Mass [slugs] **Corrected Value**
%wf = mf*32.174
% tbo1 = (m0-ml)/Dme; % burnout time [sec]

c = T/Dme;          % exhaust velocity (equal to Isp*g)[ft/sec]
n = m0/ml;          % mass ratio
tspan = [0, tbo1];  % Time Span

fprintf('Known Values:\n');
fprintf('Takeoff Weight =          %.2f [lbs]\n', w0);
fprintf('Landing Weight =          %.2f [lbs]\n', wl);
fprintf('Fuel Weight =             %.2f [lbs]\n', wf);
fprintf('Exhaust Mass Flow Rate =   %.2f [slugs/sec]\n', Dme);
fprintf('Time at Burnout =         %.2f [sec]\n', tbo1);
fprintf('ISP =                     %.2f [s]\n', Isp);
fprintf('Thrust Required =         %.2f [lbs]\n\n', T);
% fprintf('Fuel Weight Estimation =   %.2f [lbs]\n\n', wf);

vbocalc = (c*(log(n)))-((g0/Dme)*(m0-ml));         % burnout velocity [ft/s]
hbocalc = (c/Dme)*((ml*log(ml/m0))+m0-ml)-((1/2)*(tbo1^2)*g0*sind(gamma));   %burnout height [ft]

% Apex Height
b = vbocalc*sin(gammar);% Quadtratic Equation
tA = (-b - (b^2-4*0*gE/2)^(1/2))/(2*-gE/2);    % gEntry is used since accurate height isn't known yet
hA1 = b*tA/2 - gE/2*(tA/2)^2;
hmax1 = hbocalc + hA1;  % Max height using no Drag after burnout
hA = hmax1;
gA = g0/((1+(hA/rE))^2);

% Re-entry Velocity
hRe = hE - hbocalc;
tE = (-b - (b^2-4*hRe*gE/2)^(1/2))/(2*-gE/2);
VE = b - gE*tE;

fprintf('Values Calculated from Rocket Equations\n');
fprintf('Burnout Altitude =        %.2f [ft]\n', hbocalc);
fprintf('Burnout Velocity =        %.2f [ft/s]\n', vbocalc);
fprintf('Maximum Altitude =        %.2f [ft]\n', hA);
% fprintf('Reentry Altitude =        %.2f [ft]\n', hE);
fprintf('Reentry Velocity =        %.2f [ft/s]\n\n', VE);
```

```
IC = [V0; x0; hI];
% Velocity Distance Altitude %
[t,f] = ode45(@rates, tspan, IC);

v = f(:,1);
x = f(:,2);
h = f(:,3);
function dydt = rates(t,y)
dydt = zeros(3,1);
v = y(1);
x = y(2);
h = y(3);

  if t < tbo1
    m = m0 - Dme*t;
    T = T;
  else
    m = m0 - Dme*tbo1;
    T = 0;
  end

v_dot = T/m - D/m - g*sin(gammar);
x_dot = (rE/(rE + h)*v*cos(gammar));
h_dot = v*sin(gammar);

dydt(1) = v_dot;
dydt(2) = x_dot;
dydt(3) = h_dot;
end

xsize = size(x);
hsize = size(h);
vsize = size(v);
xbo = x(xsize(1));      % Taken from ode45 function, final value
vbo = v(vsize(1));      % Taken form ode45 funciton, final value
hbo = h(hsize(1));      % Taken form ode45 function, final value

hRe = hE - hbo;
cE = -hRe;
bE = vbo*sin(gammar);
aE = -gE/2;
TABE1 = (-bE-(bE^2-4*aE*cE)^(1/2))/(2*aE);
TABE2 = (-bE+(bE^2-4*aE*cE)^(1/2))/(2*aE);
if TABE2 > TABE1;
   TABE = TABE2;
else TABE = TABE1;
end

TABEALL = (-bE-(bE^2-4*0*aE)^(1/2))/(2*aE);

% ***After Burnout to Re-Entry***
% With gEntry & Integrated hbo
tAB = 0:0.5:TABE;   % Time Period After Burnout to Entry
tABsize = size(tAB);% Time Length for Re-Entry Velocity
Tsize = size(tAB);  % Time Length for Q Calculations
xAB = xbo + vbo.*cos(gammar).*tAB;
yAB = hbo + vbo.*sin(gammar).*tAB - gE./2.*tAB.^2; % gEntry is used instead of actively changing Gravity
yABhalf = hbo + vbo.*sin(gammar).*(TABEALL/2) - gE./2.*(TABEALL/2).^2;
vxAB = vbo.*cos(gammar);
vyAB = vbo.*sin(gammar) - gE.*tAB;

% ***After Burnout to Re-Entry***
```

```
% With gApex & Integrated hbo
xAB2 = xbo + vbo.*cos(gammar).*tAB;
yAB2 = h(hsize(1)) + vbo.*sin(gammar).*tAB - gA./2.*tAB.^2; % gApex is used instead of actively changing Gravity
yABhalf2 = hbo + vbo.*sin(gammar).*(TABEALL/2) - gA./2.*(TABEALL/2).^2;
vxAB2 = vbo.*cos(gammar);
vyAB2 = vbo.*sin(gammar) - gA.*tAB;

% Averaging the Values to account for Gravity Difference
xABa = (xAB + xAB2)./2;
yABa = (yAB + yAB2)./2;
vxABa = (vxAB + vxAB2)./2;     % x-direction velocity
vyABa = (vyAB + vyAB2)./2;     % y-direction velocity
VAB = sqrt(vxABa.^2+vyABa.^2);  % tangent velocity in direction of flight path
yMAX = (yABhalf + yABhalf2)/2;

% figure
plot(x,h, 'k', 'linewidth', 2)
hold on
plot(xABa, yABa, 'c','linewidth', 2)
title('Flight Path With Respect to Climb Angle')
xlabel('Horizontal Distance, ft')
ylabel('Altitude, ft')
legend('Thrusted Phase at 45 Degrees','After Burnout','Thrusted Phase at 60 Degrees','After Burnout','Thrusted Phase at
55 Degrees','After Burnout','Thrusted Phase at 57 Degrees','After Burnout','Thrusted Phase at 59 Degrees','After
Burnout','Location','southeast')

% ***Gravity Comparison***
% figure
% plot(xAB, yAB, 'b')
% hold on
% plot(xAB2, yAB2, 'g')
% plot(xABa, yABa, 'k')
% title('Comparison of Gravity Values')
% xlabel('Horizontal Distance, ft')
% ylabel('Altitude, ft')
% legend('gEntry','gApex','gAverage', 'Location','southeast')
% axis([330000, 410000, 293000 300000])

fprintf('Interated Values Using ode45\n');
fprintf('Distance at Burnout =      %.2f [ft]\n', x(xsize(1)));
fprintf('Alititude at Burnout =     %.2f [ft]\n', h(hsize(1)));
fprintf('Velocity at Burnout =      %.2f [ft/s]\n', v(vsize(1)));
fprintf('Altitude at Apex =         %.2f [ft]\n', yMAX)
% fprintf('Altitude at Re-Entry =     %.2f [ft]\n', hE);
fprintf('Velocity at Re-Entry =     %.2f [ft/s]\n', VAB(tABsize(2)));
fprintf('Time after Burnout =       %.1f [s]\n\n\n', TABE);

end
```

Code balfield.m to calculate balance field length by Mr. Faure

```
function [ BFL, Sg, Sb, Sr ] = balfield( Vef,T,Cd0,Cdflap,Cdgear,K,Clg,Cdspoilers,Cdmisc,Sref,mu,mu2,rho,Wto,g )
%   Calculates balanced field length. Works for either metric or English
%   units.
%       Vef= arbitrary engine failure speed
%       T= engine thrust during takeoff
%       Cd0= zero lift drag
%       Cdflap= drag due to flap deflection
%       Cdgear= drag due to landing gear. use 0.01 (Fig 10.5 in chudoba
%       notes
%       K=1/2*1/(AR*e) adjust for ground effect AR/ARef~=0.4
```

```
%	Clg=Clmax adjusted for ground effect
%	Cdspoilers & Cdmisc = drag due to spoilers/testing equipment/etc
%	mu= brakes off ground coefficient of friction
%	mu2= brakes on friction coefficient
%	rho= air density SLS=0.00237
%	Wto= takeoff weight
%	g= Metric: 9.18     English: 32.174 (adjust for runway altitude


% acceleration at 0.707*Vef
V=0.707*Vef;
Cd=Cd0+Cdflap+Cdgear+Cdgear+K*Clg^2;

d=1/2*rho*V^2*Sref*Cd;
l=1/2*rho*V^2*Sref*Clg;
F=mu*(Wto-l);
a=g/Wto*(T-F-d);

Sg=1/2*V^2/a; %approximated with acceleration held constant

Cd=Cd+Cdspoilers+Cdmisc; %add air braking if necessary

Sb=Wto/(g*mu2*rho*Sref*(Cd/mu2-Clg))*log(1+rho*Sref/2/Wto*(Cd/mu2-Clg)*V^2);

Sr=3*Vef; %3 second roll after deciding to abort takeoff


BFL=Sg+Sr+Sb;

End
```

Code takeoffv2.m is used to determine takeoff velocity, rate of climb, and rate of descent.

```
w0=41612.3*4.448; %N
S = 796.53*0.09290304; %m^2
p=1.225; %kg/m^3
Clmx=1.6; %estimate

vs = sqrt(w0*2/S/p/Clmx)

vto=1.2*vs%m/s
ROC = vto*sind(45)%m/s
V=3400 %ft/s
ROD = V*sind(45)%ft/s
```

Code takeofff.m by Mr. Faure is used to determine takeoff distance.

```
function [ Sto, tto, htr,f,t1 ] = takeoff( Vto,Cd0,Cdflap,Cdgear,rho,K,Clg,Sref,mu,Wto,T,g,thetadot )
%UNTITLED2 Summary of this function goes here
%   Detailed explanation goes here
V=0.707*Vto;
Cd=Cd0+Cdflap+Cdgear+K*Clg^2;
T=T;
d=1/2*rho*V^2*Sref*Cd;
l=1/2*rho*V^2*Sref*Clg;
F=mu*(Wto-l);
a=g/Wto*(T-F-d);
[Temp,Psls,rhosls,gsls]=atmos(0);
Ae=1.5;
Sg=1/2*V^2/a;
tg=V/a;
```

```
%Sr=2*Vto
m=Wto/g;
%D=1/2*rho*Vto^2*Sref*Cd;
%gammar=0;
%Str=(Vto^2/(0.15*g))*sin(thetacl);
%htr=(Vto^2/(0.15*g))*cos(thetacl);
IC=[Vto;Sg;0;0];
tspan=[tg 7.6];

[t1,f]=ode45(@rotate, tspan, IC);

v=f(:,1);
x=f(:,2);
h=f(:,3);
function dydt = rotate(t,y)
dydt = zeros(4,1);
v = y(1);
x = y(2);
h = y(3);
gammar=y(4);
    [Temp,Pe,rho,g]=atmos(h);
   D=1/2*rho*v^2*Sref*(Cd+K*Clg^2);
   L=1/2*rho*v^2*Sref*Clg;
  gammadot=thetacalc(T,L,m,7*pi/180,g,v,h,gammar);



v_dot=T/m-D/m-g*sin(gammar);

x_dot = v*cos(gammar);
h_dot = v*sin(gammar);

dydt(1) = v_dot;
dydt(2) = x_dot;
dydt(3) = h_dot;
dydt(4) = gammadot;
end
htr=h(end);
if htr>50
    Scl=0;
else
    Scl=(50-htr)/tan(f(4,end));
end

Sto=x(end)+Scl;
%htr=h(end);
tto=16;
end
```

Code land2.m used to determine the landing distance

```
w0=41612.3*4.448; %N
wl=21612.3*4.448; %N
wf=w0-wl;
wfl=wl+wf/2;
S = 796.53*0.09290304; %m^2
p=1.225; %kg/m^3
Clmx=1.6; %estimate
g=9.81; %m/s^2
Cd=0.08;
Vs = sqrt(wfl*2/S/p/Clmx);
```

```
Vf=1.3*Vs;
Vtd=1.15*Vs;
mu=0.6;
cl=1.6;
clg=0.092;

Sa=50/tand(3)/3.281;
Sfr=3*Vtd;
Sb=wfl/(g*mu*p*S)*log(1+p/2*S/wfl*(Cd/mu-clg)*Vtd^2);

S=Sa+Sb+Sfr
```

Code envelopes42.m is used to find the Maximum Mach numbers at specific heights for the flight envelope.

```
 hi = 0;            %m
hf = 100000;       %m
wi = 41612.3*4.448;      %N
wf = 21612.3*4.448;   %N
p0 = 1.225;                  %kg/m^3
S  = 796.53*0.3048^2;                                  %m^2

h = linspace(hi,hf,52);
W = linspace(wi,wf,52);
y=[1,0.8249,0.6744,0.5459,0.4371,0.3458,0.2642,0.1941,0.1427,0.1049,0.0772,0.0567,0.0417,0.0307,0.0226,0.0166,0.
0122,0.0090,0.0066,0.0049,0.0036,0.0027,0.0020,0.0014,0.0011,7.8452e-04,5.7863e-04,4.2686e-04,3.1495e-
04,2.3243e-04,1.7156e-04,1.2665e-04,9.3518e-05,6.9065e-05,5.1016e-05,3.7690e-05,2.7851e-05,2.0584e-05,1.5216e-
05,1.1249e-05,8.3187e-06,6.1527e-06,4.5514e-06,3.3675e-06,2.4920e-06,1.8445e-06,1.3655e-06,1.0110e-06,7.4874e-
07,5.5459e-07,4.1086e-07,3.0443e-07];
% y from geom224.m
p=y*p0;

Clmax=linspace(16,16/3,52)
Vstall = sqrt(W/S*2./(p.*Clmax));
b = [355.3 347.9 340.3 332.5 324.6 316.5 308.1 299.5 295.1 295.1 295.1 295.1 295.1 296.4 297.7 299.1 300.4 301.7
303 306.5 310.1 313.7 317.2 320.7 324.1 327.5 329.8 329.8 328.8 325.4 322 318.6 315.1 311.6 308 304.4 300.8 297.1
293.4 290.8 288 285.3 282.5 279.8 276.9 274 274 274 274.1 274.7 275.8 277.6];
%b from geom224.m;
M = Vstall./b
hold

plot(M(1,1:39),h(1,1:39),'g')

xlabel('Mach number')
ylabel('height (m)')


q = 1800*4.88;               %kg/m^2
V2 = sqrt(2*q./p);

M2 = V2./b

plot(M2(1,1:25),h(1,1:25),'b')
hold

T = 311.36e3;                %N
alpha = 10;        %degrees
y0 = 45;          %degrees
```

```
cd0 = 0.03;
AR = 1.95;
e = 1;
Cl = Clmax;          %estimate

K = 1/(pi*AR*e);
cd = cd0 + K.*Cl.^2;
D = T*cosd(alpha) - W*sind(y0);
V25 = sqrt(2*D./S./p/cd);

M25=V25./b
hold
plot(M25(1,20:39),h(1,20:39),'r')

w = 800+459.67; %deg. R
em = 0.8;       %emissivity
vsb = 0.481e-12; %constant
Cf = 0.0008;    %estimate

qs = w^4*em*vsb

v4 = (qs/3.21e-4/Cf./p).^(1/3)*0.3048

M4 = v4./b

plot(M4(1,20:35),h(1,20:35), 'm')
hold
```

Code turn2.m is used to determine the bank angle and banking time.

```
vre = 1.385e3;  %m/s
ph = 5;
W = 21612.3*4.448; %N
g=9.81
Cd = 0.03;
Cdt = 0.001;
S = 81;
AR = 2.4;
cl = 1.6

K=1/pi/AR;

n=1/cosd(ph);

p = 0.00196265;
q = 1/2*p*vre^2;

D = q*S*(Cd+K*(n*W/q/S)^2)+Cdt

m=W/g;
a = D/m
t=10;
hre = 45720;

v1 = vre - a*t

q1 = 1/2*p*v1^2;
D1 = q1*S*(Cd+K*(n*W/q1/S)^2)+Cdt
a1 = D1/m
v2 = v1 - a1*t
```

```
q2 = 1/2*p*v2^2;
D2 = q2*S*(Cd+K*(n*W/q2/S)^2)+Cdt
a2 = D2/m
v25 = v2 - a2*t

q25 = 1/2*p*v25^2;
D25 = q25*S*(Cd+K*(n*W/q25/S)^2)+Cdt
a25 = D25/m
v4 = v25 - a25*t

q4 = 1/2*p*v4^2;
D4 = q4*S*(Cd+K*(n*W/q4/S)^2)+Cdt
a4 = D4/m
v5 = v4 - a4*t

q5 = 1/2*p*v5^2;
D5 = q5*S*(Cd+K*(n*W/q5/S)^2)+Cdt
a5 = D5/m
v6 = v5 - a5*t

q6 = 1/2*p*v6^2;
D6 = q6*S*(Cd+K*(n*W/q6/S)^2)+Cdt
a6 = D6/m
v7 = v6 - a6*t

q7 = 1/2*p*v7^2;
D7 = q7*S*(Cd+K*(n*W/q7/S)^2)+Cdt
a7 = D7/m
v8 = v7 - a7*t

q8 = 1/2*p*v8^2;
D8 = q8*S*(Cd+K*(n*W/q8/S)^2)+Cdt
a8 = D8/m
v9 = v8 - a8*t

q9 = 1/2*p*v9^2;
D9 = q9*S*(Cd+K*(n*W/q9/S)^2)+Cdt
a9 = D9/m
v10 = v9 - a9*t

q10 = 1/2*p*v10^2;
D10 = q10*S*(Cd+K*(n*W/q10/S)^2)+Cdt
a10 = D10/m
v11 = v10 - a10*t

q11 = 1/2*p*v11^2;
D11 = q11*S*(Cd+K*(n*W/q11/S)^2)+Cdt
a11 = D11/m
v12 = v11 - a11*t

q12 = 1/2*p*v12^2;
D12 = q12*S*(Cd+K*(n*W/q12/S)^2)+Cdt
a12 = D12/m
v125 = v12 - a12*t

q125 = 1/2*p*v125^2;
D125 = q125*S*(Cd+K*(n*W/q125/S)^2)+Cdt
a125 = D125/m
v14 = v125 - a125*t

q14 = 1/2*p*v14^2;
D14 = q14*S*(Cd+K*(n*W/q14/S)^2)+Cdt
```

```
a14 = D14/m
v15 = v14 - a14*t

q15 = 1/2*p*v15^2;
D15 = q15*S*(Cd+K*(n*W/q15/S)^2)+Cdt
a15 = D15/m
v16 = v15 - a15*t

q16 = 1/2*p*v16^2;
D16 = q16*S*(Cd+K*(n*W/q16/S)^2)+Cdt
a16 = D16/m
v17 = v16 - a16*t

q17 = 1/2*p*v17^2;
D17 = q17*S*(Cd+K*(n*W/q17/S)^2)+Cdt
a17 = D17/m
v18 = v17 - a17*t

q18 = 1/2*p*v18^2;
D18 = q18*S*(Cd+K*(n*W/q18/S)^2)+Cdt
a18 = D18/m
v19 = v18 - a18*t

q19 = 1/2*p*v19^2;
D19 = q19*S*(Cd+K*(n*W/q19/S)^2)+Cdt
a19 = D19/m
v20 = v19 - a19*t

q20 = 1/2*p*v20^2;
D20 = q20*S*(Cd+K*(n*W/q20/S)^2)+Cdt
a20 = D20/m
v21 = v20 - a20*t

q21 = 1/2*p*v21^2;
D21 = q21*S*(Cd+K*(n*W/q21/S)^2)+Cdt
a21 = D21/m
v22 = v21 - a21*t

q22 = 1/2*p*v22^2;
D22 = q22*S*(Cd+K*(n*W/q22/S)^2)+Cdt
a22 = D22/m
v23 = v22 - a22*t

q23 = 1/2*p*v23^2;
D23 = q23*S*(Cd+K*(n*W/q23/S)^2)+Cdt
a23 = D23/m
v24 = v23 - a23*t

q24 = 1/2*p*v24^2;
D24 = q24*S*(Cd+K*(n*W/q24/S)^2)+Cdt
a24 = D24/m
v25 = v24 - a24*t

q25 = 1/2*p*v25^2;
D25 = q25*S*(Cd+K*(n*W/q25/S)^2)+Cdt
a25 = D25/m
v26 = v25 - a25*t

q26 = 1/2*p*v26^2;
D26 = q26*S*(Cd+K*(n*W/q26/S)^2)+Cdt
a26 = D26/m
v27 = v26 - a26*t
```

```
q27 = 1/2*p*v27^2;
D27 = q27*S*(Cd+K*(n*W/q27/S)^2)+Cdt
a27 = D27/m
v28 = v27 - a27*t

q28 = 1/2*p*v28^2;
D28 = q28*S*(Cd+K*(n*W/q28/S)^2)+Cdt
a28 = D28/m
v29 = v28 - a28*t

q29 = 1/2*p*v29^2;
D29 = q29*S*(Cd+K*(n*W/q29/S)^2)+Cdt
a29 = D29/m
v30 = v29 - a29*t

q30 = 1/2*p*v30^2;
D30 = q30*S*(Cd+K*(n*W/q30/S)^2)+Cdt
a30 = D30/m
v31 = v30 - a30*t

q31 = 1/2*p*v31^2;
D31 = q31*S*(Cd+K*(n*W/q31/S)^2)+Cdt
a31 = D31/m
v32 = v31 - a31*t

q32 = 1/2*p*v32^2;
D32 = q32*S*(Cd+K*(n*W/q32/S)^2)+Cdt
a32 = D32/m
v33 = v32 - a32*t

q33 = 1/2*p*v33^2;
D33 = q33*S*(Cd+K*(n*W/q33/S)^2)+Cdt
a33 = D33/m
v34 = v33 - a33*t

q34 = 1/2*p*v34^2;
D34 = q34*S*(Cd+K*(n*W/q34/S)^2)+Cdt
a34 = D34/m
v35 = v34 - a34*t

q35 = 1/2*p*v35^2;
D35 = q35*S*(Cd+K*(n*W/q35/S)^2)+Cdt
a35 = D35/m
v36 = v35 - a35*t

q36 = 1/2*p*v36^2;
D36 = q36*S*(Cd+K*(n*W/q36/S)^2)+Cdt
a36 = D36/m
v37 = v36 - a36*t

q37 = 1/2*p*v37^2;
D37 = q37*S*(Cd+K*(n*W/q37/S)^2)+Cdt
a37 = D37/m
v38 = v37 - a37*t

q38 = 1/2*p*v38^2;
D38 = q38*S*(Cd+K*(n*W/q38/S)^2)+Cdt
a38 = D38/m
v39 = v38 - a38*t

q39 = 1/2*p*v39^2;
D39 = q39*S*(Cd+K*(n*W/q39/S)^2)+Cdt
```

```
a39 = D39/m
v40 = v39 - a39*t

q40 = 1/2*p*v40^2;
D40 = q40*S*(Cd+K*(n*W/q40/S)^2)+Cdt
a40 = D40/m
v41 = v40 - a40*t

q41 = 1/2*p*v41^2;
D41 = q41*S*(Cd+K*(n*W/q41/S)^2)+Cdt
a41 = D41/m
v42 = v41 - a41*t

q42 = 1/2*p*v42^2;
D42 = q42*S*(Cd+K*(n*W/q42/S)^2)+Cdt
a42 = D42/m
v43 = v42 - a42*t

q43 = 1/2*p*v43^2;
D43 = q43*S*(Cd+K*(n*W/q43/S)^2)+Cdt
a43 = D43/m
v44 = v43 - a43*t

q44 = 1/2*p*v44^2;
D44 = q44*S*(Cd+K*(n*W/q44/S)^2)+Cdt
a44 = D44/m
v45 = v44 - a44*t

q45 = 1/2*p*v45^2;
D45 = q45*S*(Cd+K*(n*W/q45/S)^2)+Cdt
a45 = D45/m
v46 = v45 - a45*t

q46 = 1/2*p*v46^2;
D46 = q46*S*(Cd+K*(n*W/q46/S)^2)+Cdt
a46 = D46/m
v47 = v46 - a46*t

t2=6
q47 = 1/2*p*v47^2;
D47 = q47*S*(Cd+K*(n*W/q47/S)^2)+Cdt
a47 = D47/m
v48 = v47 - a47*t2
ttotal = t*47+t2

L = 1/2*p*v48^2*cl*S
D = 1/2*p*v48^2*Cd*S
af = (L-W*sind(7))/m

t5 =275.802
h = hre - v48*sind(7)*t5 + 1/2*af*t5^2

vfin = v48 - 1/2*af*cosd(7)*t5

talll = t5+ttotal

R = ttotal*(vre+v48)/2*57.3/10

gforce = (vre-v48)/ttotal
```

Code Thrust2h2.m is used to find maximum height, burn out velocity, burn out height

```
% Velocity at 123000 ft
% The thrust calculation for the s

M = 41612.3*0.45;       %mass in kg
g = 9.8;        %m/s^2
theta = 90;     %flight path angle
C_d = 0.03;     %co-efficient of delta
rho = 5.19e-3;      %density in the altitude
V= 940;         %                   m/s
A = 81;                                 %area of the wing m^2

Isp=320;   %s

   %kg/s

mf = 21612.3*0.45;   %kg

mf=10176
tbo=95
t = tbo;     %burn time (designed)
me=(M-mf)/t   %kg/s

F = M*g*sind(0)+C_d*rho*V*V*A/2 + M*V/t;     %Force in newtons
F_lbs =F *0.2248                          %Thrust in lbs

F2 = M*g*sind(45)+C_d*rho*V*V*A/2 + M*V/t;     %Force in newtons
F_lbs2 =F2 *0.2248                          %Thrust in lbs

F25 = (M-mf)/2*g*sind(80)+C_d*rho*V*V*A/2 + (M-mf)/2*V/t;
F_lbs25 =F25 *0.2248

c = Isp*g

vbo = (c*log(M/mf) - g/me*(M-mf))
hbo = 15239+(c/me*(mf*log(mf/M) + M - mf)-1/2*((M-mf)/me)^2*g)

tcoast=c/g*log(M/mf)-tbo  %s

n=M/mf

hmax= 15239+(1/2*c^2/g*((log(n))^2)-c*M/me*(n*log(n)-(n-1))/n)

re=6378e3

gB = g/(1+hmax/re)^2
hA =90000 %m
h2=90000
gc = g/(1+hA/re)^2

tmicro= 2*sqrt(2*(hmax-hA)/gB)
tmicro2 = sqrt(2*(hA-h2)/gc)
tmicrototal = tmicro+tmicro2

vre1 = -gB*(tmicro/2+tmicro2)

h4=45720
gd = g/(1+h4/re)^2
tmicro4 = sqrt(2*(h2-h4)/gc)
D = C_d*rho*vre1*vre1*A/2
a = (D - mf*gd*sind(theta))/mf

vre = vre1 +a*tmicro4
```

```
hre = hmax - 1/2*gB*(tmicro2)^2

tmicrototal = tmicro+tmicro2+tmicro4


Wf = 21612.3*4.448   %N

D = C_d*rho*vre*vre*A/2
a = (D + Wf*sind(theta))/mf

AR=2.4
K=1/pi/AR
```

Code FlightPathFlightAngle45dbld4h.m is used to determine the flight path

```
%%%%%%%%%Rocket Performance Calculations%%%%%%%%%%%%%%%
function FlightPathAngle45dbld

%Givens:
w0 = 41612.3;         % Launch Weight [lbs]
wl = 21612.3;       % Landing Weight [lbs]
wf = w0 - wl;       % Fuel Weight [lbs]
S  = 00;            % Extra Fuel for Commercial W/D=9900,  Wo/D=0
w0 = w0 + S;        % Launch Weight accounting for added fuel
wf = wf + S;        % Fuel Weight with addition for Commercial
wp = 0;             % Payload Weight [lbs]
g0 = 32.174;        %[ft/s^2]
rE = 6378*3280.84;  %[ft]
% Isp = 284;          % Specific Impulse at 45000 ft for XLR-99 Engine  [sec]
T  = 70000;      % Thrust for XLR-99 at 45000 ft  [lbs]
h0 = 50000;         % Launch elevation [ft]
x0 = 0;            % Initial Distance [ft]
V0 = 0;           % Initial Velocity [ft/s]
t1 = 15;            % [sec]
h  = 0;         % Ignition Elevation [ft]
gamma = 80;         % Flight Path Angle (assume constant) [degrees]
gammar = gamma*pi/180; % Flight Path Angle [radians]
D  = w0*cosd(gamma)/3.1;          % Drag Force [lbs]

hE = 360892;
gE = g0/((1+(hE/rE))^2);
g = g0/((1+(h/rE))^2);    % gravitational force at specific altitude [ft/s^2]
hI = h0-((g/2)*(t1^2));   % ignition elevation [ft]

%Calculations:
m0 = w0/32.174;     % Launch Mass[slugs]
mf = wf/32.174;     % Fuel Mass[slugs]  ***Replaced by later value***
ml = 10176*0.0685;     % Landing (Empty) Mass[slugs]
mp = wp/32.174;     % Payload Mass[slugs]

%Dme = T/(Isp*g);    % exhaust mass flow rate [slug/s]
tbo1 = 100;       % Burn Time from rcktperfdlbd [s]
Dme = (m0-ml)/tbo1; % Mass Flow Rate Defined by Tbo
Isp = T/(g*Dme);    % Specific Impulse Defined by Tbo and T
%mf = -tbo1*Dme+m0;  % Fuel Mass [slugs] **Corrected Value**
%wf = mf*32.174
% tbo1 = (m0-ml)/Dme; % burnout time [sec]

c = T/Dme;          % exhaust velocity (equal to Isp*g)[ft/sec]
n = m0/ml;          % mass ratio
tspan = [0, tbo1];  % Time Span
```

```
fprintf('Known Values:\n');
fprintf('Takeoff Weight =          %.2f [lbs]\n', w0);
fprintf('Landing Weight =          %.2f [lbs]\n', wl);
fprintf('Fuel Weight =             %.2f [lbs]\n', wf);
fprintf('Exhaust Mass Flow Rate =  %.2f [slugs/sec]\n', Dme);
fprintf('Time at Burnout =         %.2f [sec]\n', tbo1);
fprintf('ISP =                     %.2f [s]\n', Isp);
fprintf('Thrust Required =         %.2f [lbs]\n\n', T);
% fprintf('Fuel Weight Estimation =  %.2f [lbs]\n\n', wf);

vbocalc = (c*(log(n)))-((g0/Dme)*(m0-ml));        % burnout velocity [ft/s]
hbocalc = (c/Dme)*((ml*log(ml/m0))+m0-ml)-((1/2)*(tbo1^2)*g0*sind(gamma));   %burnout height [ft]

% Apex Height
b = vbocalc*sin(gammar);% Quadtratic Equation
tA = (-b - (b^2-4*0*gE/2)^(1/2))/(2*-gE/2);    % gEntry is used since accurate height isn't known yet
hA1 = b*tA/2 - gE/2*(tA/2)^2;
hmax1 = hbocalc + hA1;  % Max height using no Drag after burnout
hA = hmax1;
gA = g0/((1+(hA/rE))^2);

% Re-entry Velocity
hRe = hE - hbocalc;
tE = (-b - (b^2-4*hRe*gE/2)^(1/2))/(2*-gE/2);
VE = b - gE*tE;

fprintf('Values Calculated from Rocket Equations\n');
fprintf('Burnout Altitude =        %.2f [ft]\n', hbocalc);
fprintf('Burnout Velocity =        %.2f [ft/s]\n', vbocalc);
fprintf('Maximum Altitude =        %.2f [ft]\n', hA);
% fprintf('Reentry Altitude =      %.2f [ft]\n', hE);
fprintf('Reentry Velocity =        %.2f [ft/s]\n\n', VE);


IC = [V0; x0; hI];
% Velocity Distance Altitude %
[t,f] = ode45(@rates, tspan, IC);

v = f(:,1);
x = f(:,2);
h = f(:,3);
function dydt = rates(t,y)
dydt = zeros(3,1);
v = y(1);
x = y(2);
h = y(3);

    if t < tbo1
        m = m0 - Dme*t;
        T = T;
    else
        m = m0 - Dme*tbo1;
        T = 0;
    end

v_dot = T/m - D/m - g*sin(gammar);
x_dot = (rE/(rE + h)*v*cos(gammar));
h_dot = v*sin(gammar);

dydt(1) = v_dot;
dydt(2) = x_dot;
dydt(3) = h_dot;
```

```
end

xsize = size(x);
hsize = size(h);
vsize = size(v);
xbo = x(xsize(1));      % Taken from ode45 function, final value
vbo = v(vsize(1));      % Taken form ode45 funciton, final value
hbo = h(hsize(1));      % Taken form ode45 function, final value

hRe = hE - hbo;
cE = -hRe;
bE = vbo*sin(gammar);
aE = -gE/2;
TABE1 = (-bE-(bE^2-4*aE*cE)^(1/2))/(2*aE);
TABE2 = (-bE+(bE^2-4*aE*cE)^(1/2))/(2*aE);
if TABE2 > TABE1;
   TABE = TABE2;
else TABE = TABE1;
end

TABEALL = (-bE-(bE^2-4*0*aE)^(1/2))/(2*aE);

% ***After Burnout to Re-Entry***
% With gEntry & Integrated hbo
tAB = 0:0.5:TABE;   % Time Period After Burnout to Entry
tABsize = size(tAB);% Time Length for Re-Entry Velocity
Tsize = size(tAB);  % Time Length for Q Calculations
xAB = xbo + vbo.*cos(gammar).*tAB;
yAB = hbo + vbo.*sin(gammar).*tAB - gE./2.*tAB.^2; % gEntry is used instead of actively changing Gravity
yABhalf = hbo + vbo.*sin(gammar).*(TABEALL/2) - gE./2.*(TABEALL/2).^2;
vxAB = vbo.*cos(gammar);
vyAB = vbo.*sin(gammar) - gE.*tAB;

% ***After Burnout to Re-Entry***
% With gApex & Integrated hbo
xAB2 = xbo + vbo.*cos(gammar).*tAB;
yAB2 = h(hsize(1)) + vbo.*sin(gammar).*tAB - gA./2.*tAB.^2; % gApex is used instead of actively changing Gravity
yABhalf2 = hbo + vbo.*sin(gammar).*(TABEALL/2) - gA./2.*(TABEALL/2).^2;
vxAB2 = vbo.*cos(gammar);
vyAB2 = vbo.*sin(gammar) - gA.*tAB;

% Averaging the Values to account for Gravity Difference
xABa = (xAB + xAB2)./2;
yABa = (yAB + yAB2)./2;
vxABa = (vxAB + vxAB2)./2;      % x-direction velocity
vyABa = (vyAB + vyAB2)./2;      % y-direction velocity
VAB = sqrt(vxABa.^2+vyABa.^2);  % tangent velocity in direction of flight path
yMAX = (yABhalf + yABhalf2)/2;

% figure
plot(x,h, 'k', 'linewidth', 2)
hold on
plot(xABa, yABa, 'c','linewidth', 2)
title('Flight Path With Respect to Climb Angle')
xlabel('Horizontal Distance, ft')
ylabel('Altitude, ft')
legend('Thrusted Phase at 80 Degrees','After Burnout','Thrusted Phase at 60 Degrees','After Burnout','Thrusted Phase at
55 Degrees','After Burnout','Thrusted Phase at 57 Degrees','After Burnout','Thrusted Phase at 59 Degrees','After
Burnout','Location','southeast')

% ***Gravity Comparison***
% figure
```

```
% plot(xAB, yAB, 'b')
% hold on
% plot(xAB2, yAB2, 'g')
% plot(xABa, yABa, 'k')
% title('Comparison of Gravity Values')
% xlabel('Horizontal Distance, ft')
% ylabel('Altitude, ft')
% legend('gEntry','gApex','gAverage', 'Location','southeast')
% axis([330000, 410000, 293000 300000])

fprintf('Interated Values Using ode45\n');
fprintf('Distance at Burnout =      %.2f [ft]\n', x(xsize(1)));
fprintf('Alititude at Burnout =     %.2f [ft]\n', h(hsize(1)));
fprintf('Velocity at Burnout =      %.2f [ft/s]\n', v(vsize(1)));
fprintf('Altitude at Apex =         %.2f [ft]\n', yMAX)
% fprintf('Altitude at Re-Entry =     %.2f [ft]\n', hE);
fprintf('Velocity at Re-Entry =     %.2f [ft/s]\n', VAB(tABsize(2)));
fprintf('Time after Burnout =       %.1f [s]\n\n\n', TABE);

end
```

Code envelopes42h.m is used to determine the flight envelope

```
hi = 15239;             %m
hf = 100000;        %m
wi = 41612.3*4.448;       %N
wf = 21612.3*4.448;   %N
p0 = 1.225;                     %kg/m^3
S  = 81;                        %m^2

h = linspace(hi,hf,52);
W = linspace(wi,wf,52);
y=[1,0.8249,0.6744,0.5459,0.4371,0.3458,0.2642,0.1941,0.1427,0.1049,0.0772,0.0567,0.0417,0.0307,0.0226,0.0166,0.
0122,0.0090,0.0066,0.0049,0.0036,0.0027,0.0020,0.0014,0.0011,7.8452e-04,5.7863e-04,4.2686e-04,3.1495e-
04,2.3243e-04,1.7156e-04,1.2665e-04,9.3518e-05,6.9065e-05,5.1016e-05,3.7690e-05,2.7851e-05,2.0584e-05,1.5216e-
05,1.1249e-05,8.3187e-06,6.1527e-06,4.5514e-06,3.3675e-06,2.4920e-06,1.8445e-06,1.3655e-06,1.0110e-06,7.4874e-
07,5.5459e-07,4.1086e-07,3.0443e-07];
% y from geom224.m
p=y*p0;

Clmax=linspace(16,16/3,52);
Vstall = sqrt(W/S*2./(p.*Clmax));
b = [355.3 347.9 340.3 332.5 324.6 316.5 308.1 299.5 295.1 295.1 295.1 295.1 295.1 296.4 297.7 299.1 300.4 301.7
303 306.5 310.1 313.7 317.2 320.7 324.1 327.5 329.8 329.8 328.8 325.4 322 318.6 315.1 311.6 308 304.4 300.8 297.1
293.4 290.8 288 285.3 282.5 279.8 276.9 274 274 274 274.1 274.7 275.8 277.6];
%b from geom224.m;
M = Vstall./b

hold

plot(M(1,1:39),h(1,1:39),'g')

xlabel('Mach number')
ylabel('height (m)')


q = 1800*4.88;                  %kg/m^2
V2 = sqrt(2*q./p);

M2 = V2./b
```

```
plot(M2(1,1:25),h(1,1:25),'b')
hold

T = 311.36e3;                  %N
alpha = 10;         %degrees
y0 = 90;            %degrees
cd0 = 0.03;
AR = 2.4;
e = 1;
Cl = Clmax;           %estimate

K = 1/(pi*AR*e);
cd = cd0 + K.*Cl.^2;
D = T*cosd(alpha) - W*sind(y0);
V25 = sqrt(2*D./S./p/cd);

M25=V25./b
hold
plot(M25(1,20:39),h(1,20:39),'r')

w = 1000+459.67; %deg. R
em = 0.8;       %emissivity
vsb = 0.481e-12; %constant
Cf = 0.0008;    %estimate

qs = w^4*em*vsb

v4 = (qs/3.21e-4/Cf./p).^(1/3)*0.3048;

M4 = v4./b+9

plot(M4(1,20:35),h(1,20:35), 'm')
hold
```

Code Thrust20h.m is used to determine the orbital burn time and maximum height

```
% Velocity at 123000 ft
% The thrust calculation for the s

M = 41612.3*0.45;       %mass in kg
g = 9.8;        %m/s^2
theta = 90;     %flight path angle
C_d = 0.03;     %co-efficient of delta
rho = 0.00001846;      %density in the altitude
V= 940;         %                  m/s
A = 81;                                 %area of the wing m^2

Isp=500;   %s

   %kg/s

mf = 21612.3*0.45;   %kg

tbo=120
t = tbo;     %burn time (designed)
me=(M-mf)/t   %kg/s

F = M*g*sind(0)+C_d*rho*V*V*A/2 + M*V/t;     %Force in newtons
F_lbs =F *0.2248                          %Thrust in lbs

F2 = M*g*sind(45)+C_d*rho*V*V*A/2 + M*V/t;     %Force in newtons
```

```
F_lbs2 =F2 *0.2248                    %Thrust in lbs

F25 = (M-mf)/2*g*sind(80)+C_d*rho*V*V*A/2 + (M-mf)/2*V/t;
F_lbs25 =F25 *0.2248
Isp=70000*4.448/me/g
c = Isp*g


vbo = (c*log(M/mf) - g/me*(M-mf))
hbo = 15239+(c/me*(mf*log(mf/M) + M - mf)-1/2*((M-mf)/me)^2*g)

tcoast=c/g*log(M/mf)-tbo  %s

n=M/mf

hmax= 15239+(1/2*c^2/g*((log(n))^2)-c*M/me*(n*log(n)-(n-1))/n)

re=6378e3

gB = g/(1+hmax/re)^2
hA =90000 %m
h2=66000
gc = g/(1+hA/re)^2

tmicro= 2*sqrt(2*(hmax-hA)/gB)
tmicro2 = sqrt(2*(hA-h2)/gc)
tmicrototal = tmicro+tmicro2

vre = -gB*(tmicro/2+tmicro2)
hre = hmax - 1/2*gB*(tmicro2)^2

Wf = 21612.3*4.448   %N

D = C_d*rho*vre*vre*A/2
a = (D + Wf*sind(theta))/mf

AR=2.4
K=1/pi/AR
```

Code FlightPathAngle45dbld4ho.m is used to find the flight path

```
%%%%%%%%%%Rocket Performance Calculations%%%%%%%%%%%%%%%%%
function FlightPathAngle45dbld

%Givens:
w0 = 41612.3;        % Launch Weight [lbs]
wl = 21612.3;       % Landing Weight [lbs]
wf = w0 - wl;       % Fuel Weight [lbs]
S  = 00;           % Extra Fuel for Commercial W/D=9900,  Wo/D=0
w0 = w0 + S;        % Launch Weight accounting for added fuel
wf = wf + S;        % Fuel Weight with addition for Commercial
wp = 0;            % Payload Weight [lbs]
g0 = 32.174;        %[ft/s^2]
rE = 6378*3280.84;  %[ft]
% Isp = 284;          % Specific Impulse at 45000 ft for XLR-99 Engine  [sec]
T  = 70000;      % Thrust for XLR-99 at 45000 ft  [lbs]
h0 = 0;        % Launch elevation [ft]
x0 = 0;           % Initial Distance [ft]
V0 = 0;          % Initial Velocity [ft/s]
t1 = 15;          % [sec]
h  = 0;        % Ignition Elevation [ft]
gamma = 80;        % Flight Path Angle (assume constant) [degrees]
```

```
gammar = gamma*pi/180; % Flight Path Angle [radians]
D  = w0*cosd(gamma)/3.1;          % Drag Force [lbs]

hE = 360892;
gE = g0/((1+(hE/rE))^2);
g = g0/((1+(h/rE))^2);    % gravitational force at specific altitude [ft/s^2]
hI = h0-((g/2)*(t1^2));   % ignition elevation [ft]

%Calculations:
m0 = w0/32.174;     % Launch Mass[slugs]
mf = wf/32.174;     % Fuel Mass[slugs]  ***Replaced by later value***
ml = wl/32.174;     % Landing (Empty) Mass[slugs]
mp = wp/32.174;     % Payload Mass[slugs]

%Dme = T/(Isp*g);    % exhaust mass flow rate [slug/s]
tbo1 = 130;       % Burn Time from rcktperfdlbd [s]
Dme = (m0-ml)/tbo1; % Mass Flow Rate Defined by Tbo
Isp = T/(g*Dme);    % Specific Impulse Defined by Tbo and T
%mf = -tbo1*Dme+m0;  % Fuel Mass [slugs] **Corrected Value**
%wf = mf*32.174
% tbo1 = (m0-ml)/Dme; % burnout time [sec]

c = T/Dme;          % exhaust velocity (equal to Isp*g)[ft/sec]
n = m0/ml;          % mass ratio
tspan = [0, tbo1];  % Time Span

fprintf('Known Values:\n');
fprintf('Takeoff Weight =          %.2f [lbs]\n', w0);
fprintf('Landing Weight =          %.2f [lbs]\n', wl);
fprintf('Fuel Weight =             %.2f [lbs]\n', wf);
fprintf('Exhaust Mass Flow Rate =   %.2f [slugs/sec]\n', Dme);
fprintf('Time at Burnout =         %.2f [sec]\n', tbo1);
fprintf('ISP =                     %.2f [s]\n', Isp);
fprintf('Thrust Required =         %.2f [lbs]\n\n', T);
% fprintf('Fuel Weight Estimation =   %.2f [lbs]\n\n', wf);

vbocalc = (c*(log(n)))-((g0/Dme)*(m0-ml));         % burnout velocity [ft/s]
hbocalc = (c/Dme)*((ml*log(ml/m0))+m0-ml)-((1/2)*(tbo1^2)*g0*sind(gamma));   %burnout height [ft]

% Apex Height
b = vbocalc*sin(gammar);% Quadtratic Equation
tA = (-b - (b^2-4*0*gE/2)^(1/2))/(2*-gE/2);    % gEntry is used since accurate height isn't known yet
hA1 = b*tA/2 - gE/2*(tA/2)^2;
hmax1 = hbocalc + hA1;  % Max height using no Drag after burnout
hA = hmax1;
gA = g0/((1+(hA/rE))^2);

% Re-entry Velocity
hRe = hE - hbocalc;
tE = (-b - (b^2-4*hRe*gE/2)^(1/2))/(2*-gE/2);
VE = b - gE*tE;

fprintf('Values Calculated from Rocket Equations\n');
fprintf('Burnout Altitude =         %.2f [ft]\n', hbocalc);
fprintf('Burnout Velocity =         %.2f [ft/s]\n', vbocalc);
fprintf('Maximum Altitude =         %.2f [ft]\n', hA);
% fprintf('Reentry Altitude =         %.2f [ft]\n', hE);
fprintf('Reentry Velocity =         %.2f [ft/s]\n\n', VE);


IC = [V0; x0; hI];
% Velocity Distance Altitude %
```

```
[t,f] = ode45(@rates, tspan, IC);

v = f(:,1);
x = f(:,2);
h = f(:,3);
function dydt = rates(t,y)
dydt = zeros(3,1);
v = y(1);
x = y(2);
h = y(3);

   if t < tbo1
      m = m0 - Dme*t;
      T = T;
   else
      m = m0 - Dme*tbo1;
      T = 0;
   end

v_dot = T/m - D/m - g*sin(gammar);
x_dot = (rE/(rE + h)*v*cos(gammar));
h_dot = v*sin(gammar);

dydt(1) = v_dot;
dydt(2) = x_dot;
dydt(3) = h_dot;
end

xsize = size(x);
hsize = size(h);
vsize = size(v);
xbo = x(xsize(1));      % Taken from ode45 function, final value
vbo = v(vsize(1));      % Taken form ode45 funciton, final value
hbo = h(hsize(1));      % Taken form ode45 function, final value

hRe = hE - hbo;
cE = -hRe;
bE = vbo*sin(gammar);
aE = -gE/2;
TABE1 = (-bE-(bE^2-4*aE*cE)^(1/2))/(2*aE);
TABE2 = (-bE+(bE^2-4*aE*cE)^(1/2))/(2*aE);
if TABE2 > TABE1;
   TABE = TABE2;
else TABE = TABE1;
end

TABEALL = (-bE-(bE^2-4*0*aE)^(1/2))/(2*aE);

% ***After Burnout to Re-Entry***
% With gEntry & Integrated hbo
tAB = 0:0.5:TABE;   % Time Period After Burnout to Entry
tABsize = size(tAB);% Time Length for Re-Entry Velocity
Tsize = size(tAB);  % Time Length for Q Calculations
xAB = xbo + vbo.*cos(gammar).*tAB;
yAB = hbo + vbo.*sin(gammar).*tAB - gE./2.*tAB.^2; % gEntry is used instead of actively changing Gravity
yABhalf = hbo + vbo.*sin(gammar).*(TABEALL/2) - gE./2.*(TABEALL/2).^2;
vxAB = vbo.*cos(gammar);
vyAB = vbo.*sin(gammar) - gE.*tAB;

% ***After Burnout to Re-Entry***
% With gApex & Integrated hbo
xAB2 = xbo + vbo.*cos(gammar).*tAB;
```

```
yAB2 = h(hsize(1)) + vbo.*sin(gammar).*tAB - gA./2.*tAB.^2; % gApex is used instead of actively changing Gravity
yABhalf2 = hbo + vbo.*sin(gammar).*(TABEALL/2) - gA./2.*(TABEALL/2).^2;
vxAB2 = vbo.*cos(gammar);
vyAB2 = vbo.*sin(gammar) - gA.*tAB;

% Averaging the Values to account for Gravity Difference
xABa = (xAB + xAB2)./2;
yABa = (yAB + yAB2)./2;
vxABa = (vxAB + vxAB2)./2;      % x-direction velocity
vyABa = (vyAB + vyAB2)./2;      % y-direction velocity
VAB = sqrt(vxABa.^2+vyABa.^2);  % tangent velocity in direction of flight path
yMAX = (yABhalf + yABhalf2)/2;

% figure
plot(x,h, 'k', 'linewidth', 2)
hold on
plot(xABa, yABa, 'c','linewidth', 2)
title('Flight Path With Respect to Climb Angle')
xlabel('Horizontal Distance, ft')
ylabel('Altitude, ft')
legend('Thrusted Phase at 80 Degrees','After Burnout','Thrusted Phase at 60 Degrees','After Burnout','Thrusted Phase at
55 Degrees','After Burnout','Thrusted Phase at 57 Degrees','After Burnout','Thrusted Phase at 59 Degrees','After
Burnout','Location','southeast')

% ***Gravity Comparison***
% figure
% plot(xAB, yAB, 'b')
% hold on
% plot(xAB2, yAB2, 'g')
% plot(xABa, yABa, 'k')
% title('Comparison of Gravity Values')
% xlabel('Horizontal Distance, ft')
% ylabel('Altitude, ft')
% legend('gEntry','gApex','gAverage', 'Location','southeast')
% axis([330000, 410000, 293000 300000])

fprintf('Interated Values Using ode45\n');
fprintf('Distance at Burnout =      %.2f [ft]\n', x(xsize(1)));
fprintf('Alititude at Burnout =     %.2f [ft]\n', h(hsize(1)));
fprintf('Velocity at Burnout =      %.2f [ft/s]\n', v(vsize(1)));
fprintf('Altitude at Apex =         %.2f [ft]\n', yMAX)
% fprintf('Altitude at Re-Entry =     %.2f [ft]\n', hE);
fprintf('Velocity at Re-Entry =     %.2f [ft/s]\n', VAB(tABsize(2)));
fprintf('Time after Burnout =       %.1f [s]\n\n\n', TABE);

end
```

Code envelopes42oh.m is used to find the flight envelope for orbital air drop conditions

```
hi = 15239;             %m
hf = 200000;        %m
wi = 41612.3*4.448;       %N
wf = 21612.3*4.448;   %N
p0 = 1.225;                   %kg/m^3
S  = 81;                      %m^2

h = linspace(hi,hf,52);
W = linspace(wi,wf,52);
y=[1,0.8249,0.6744,0.5459,0.4371,0.3458,0.2642,0.1941,0.1427,0.1049,0.0772,0.0567,0.0417,0.0307,0.0226,0.0166,0.
0122,0.0090,0.0066,0.0049,0.0036,0.0027,0.0020,0.0014,0.0011,7.8452e-04,5.7863e-04,4.2686e-04,3.1495e-
04,2.3243e-04,1.7156e-04,1.2665e-04,9.3518e-05,6.9065e-05,5.1016e-05,3.7690e-05,2.7851e-05,2.0584e-05,1.5216e-
```

```
05,1.1249e-05,8.3187e-06,6.1527e-06,4.5514e-06,3.3675e-06,2.4920e-06,1.8445e-06,1.3655e-06,1.0110e-06,7.4874e-
07,5.5459e-07,4.1086e-07,3.0443e-07];
% y from geom224.m
p=y*p0;

Clmax=linspace(16,16/3,52);
Vstall = sqrt(W/S*2./(p.*Clmax));
b = [355.3 347.9 340.3 332.5 324.6 316.5 308.1 299.5 295.1 295.1 295.1 295.1 295.1 296.4 297.7 299.1 300.4 301.7
303 306.5 310.1 313.7 317.2 320.7 324.1 327.5 329.8 329.8 328.8 325.4 322 318.6 315.1 311.6 308 304.4 300.8 297.1
293.4 290.8 288 285.3 282.5 279.8 276.9 274 274 274 274.1 274.7 275.8 277.6];
%b from geom224.m;
M = Vstall./b

hold

plot(M(1,1:39),h(1,1:39),'g')

xlabel('Mach number')
ylabel('height (m)')


q = 1800*4.88;              %kg/m^2
V2 = sqrt(2*q./p);

M2 = V2./b

plot(M2(1,1:25),h(1,1:25),'b')
hold

T = 311.36e3;               %N
alpha = 10;        %degrees
y0 = 90;           %degrees
cd0 = 0.03;
AR = 2.4;
e = 1;
Cl = Clmax;          %estimate

K = 1/(pi*AR*e);
cd = cd0 + K.*Cl.^2;
D = T*cosd(alpha) - W*sind(y0);
V25 = sqrt(2*D./S./p/cd);

M25=V25./b
hold
plot(M25(1,20:39),h(1,20:39),'r')

w = 1000+459.67; %deg. R
em = 0.8;      %emissivity
vsb = 0.481e-12; %constant
Cf = 0.0008;   %estimate

qs = w^4*em*vsb

v4 = (qs/3.21e-4/Cf./p).^(1/3)*0.3048;

M4 = v4./b+9

plot(M4(1,20:35),h(1,20:35), 'm')
hold
```

Code turno2h.m is used to find the banking angle and turning time

```
vre = 2.15e3;  %m/s
ph = 30;
W = 21612.3*4.448; %N
g=9.81
Cd = 0.03;
Cdt = 0.001;
S = 81;
AR = 2.4;
cl = 1.6

K=1/pi/AR;

n=1/cosd(ph);

p = 0.00196265;
q = 1/2*p*vre^2;

D = q*S*(Cd+K*(n*W/q/S)^2)+Cdt

m=W/g;
a = D/m
t=25;
hre = 66000;

v1 = vre - a*t

q1 = 1/2*p*v1^2;
D1 = q1*S*(Cd+K*(n*W/q1/S)^2)+Cdt
a1 = D1/m
v2 = v1 - a1*t

q2 = 1/2*p*v2^2;
D2 = q2*S*(Cd+K*(n*W/q2/S)^2)+Cdt
a2 = D2/m
v25 = v2 - a2*t

q25 = 1/2*p*v25^2;
D25 = q25*S*(Cd+K*(n*W/q25/S)^2)+Cdt
a25 = D25/m
v4 = v25 - a25*t

q4 = 1/2*p*v4^2;
D4 = q4*S*(Cd+K*(n*W/q4/S)^2)+Cdt
a4 = D4/m
v5 = v4 - a4*t

q5 = 1/2*p*v5^2;
D5 = q5*S*(Cd+K*(n*W/q5/S)^2)+Cdt
a5 = D5/m
v6 = v5 - a5*t

q6 = 1/2*p*v6^2;
D6 = q6*S*(Cd+K*(n*W/q6/S)^2)+Cdt
a6 = D6/m
v7 = v6 - a6*t
q7 = 1/2*p*v7^2;
D7 = q7*S*(Cd+K*(n*W/q7/S)^2)+Cdt
a7 = D7/m
v8 = v7 - a7*t

q8 = 1/2*p*v8^2;
D8 = q8*S*(Cd+K*(n*W/q8/S)^2)+Cdt
```

```
a8 = D8/m
v9 = v8 - a8*t

q9 = 1/2*p*v9^2;
D9 = q9*S*(Cd+K*(n*W/q9/S)^2)+Cdt
a9 = D9/m
v10 = v9 - a9*t

q10 = 1/2*p*v10^2;
D10 = q10*S*(Cd+K*(n*W/q10/S)^2)+Cdt
a10 = D10/m
v11 = v10 - a10*t

q11 = 1/2*p*v11^2;
D11 = q11*S*(Cd+K*(n*W/q11/S)^2)+Cdt
a11 = D11/m
v12 = v11 - a11*t

q12 = 1/2*p*v12^2;
D12 = q12*S*(Cd+K*(n*W/q12/S)^2)+Cdt
a12 = D12/m
v125 = v12 - a12*t

q125 = 1/2*p*v125^2;
D125 = q125*S*(Cd+K*(n*W/q125/S)^2)+Cdt
a125 = D125/m
v14 = v125 - a125*t

q14 = 1/2*p*v14^2;
D14 = q14*S*(Cd+K*(n*W/q14/S)^2)+Cdt
a14 = D14/m
v15 = v14 - a14*t

q15 = 1/2*p*v15^2;
D15 = q15*S*(Cd+K*(n*W/q15/S)^2)+Cdt
a15 = D15/m
v16 = v15 - a15*t

q16 = 1/2*p*v16^2;
D16 = q16*S*(Cd+K*(n*W/q16/S)^2)+Cdt
a16 = D16/m
v17 = v16 - a16*t

q17 = 1/2*p*v17^2;
D17 = q17*S*(Cd+K*(n*W/q17/S)^2)+Cdt
a17 = D17/m
v18 = v17 - a17*t

q18 = 1/2*p*v18^2;
D18 = q18*S*(Cd+K*(n*W/q18/S)^2)+Cdt
a18 = D18/m
v19 = v18 - a18*t

q19 = 1/2*p*v19^2;
D19 = q19*S*(Cd+K*(n*W/q19/S)^2)+Cdt
a19 = D19/m
v20 = v19 - a19*t

q20 = 1/2*p*v20^2;
D20 = q20*S*(Cd+K*(n*W/q20/S)^2)+Cdt
a20 = D20/m
v21 = v20 - a20*t
```

```
q21 = 1/2*p*v21^2;
D21 = q21*S*(Cd+K*(n*W/q21/S)^2)+Cdt
a21 = D21/m
v22 = v21 - a21*t

q22 = 1/2*p*v22^2;
D22 = q22*S*(Cd+K*(n*W/q22/S)^2)+Cdt
a22 = D22/m
v23 = v22 - a22*t

q23 = 1/2*p*v23^2;
D23 = q23*S*(Cd+K*(n*W/q23/S)^2)+Cdt
a23 = D23/m
v24 = v23 - a23*t

q24 = 1/2*p*v24^2;
D24 = q24*S*(Cd+K*(n*W/q24/S)^2)+Cdt
a24 = D24/m
v25 = v24 - a24*t

q25 = 1/2*p*v25^2;
D25 = q25*S*(Cd+K*(n*W/q25/S)^2)+Cdt
a25 = D25/m
v26 = v25 - a25*t

q26 = 1/2*p*v26^2;
D26 = q26*S*(Cd+K*(n*W/q26/S)^2)+Cdt
a26 = D26/m
v27 = v26 - a26*t


q27 = 1/2*p*v27^2;
D27 = q27*S*(Cd+K*(n*W/q27/S)^2)+Cdt
a27 = D27/m
v28 = v27 - a27*t

q28 = 1/2*p*v28^2;
D28 = q28*S*(Cd+K*(n*W/q28/S)^2)+Cdt
a28 = D28/m
v29 = v28 - a28*t

q29 = 1/2*p*v29^2;
D29 = q29*S*(Cd+K*(n*W/q29/S)^2)+Cdt
a29 = D29/m
v30 = v29 - a29*t

q30 = 1/2*p*v30^2;
D30 = q30*S*(Cd+K*(n*W/q30/S)^2)+Cdt
a30 = D30/m
v31 = v30 - a30*t

q31 = 1/2*p*v31^2;
D31 = q31*S*(Cd+K*(n*W/q31/S)^2)+Cdt
a31 = D31/m
v32 = v31 - a31*t

q32 = 1/2*p*v32^2;
D32 = q32*S*(Cd+K*(n*W/q32/S)^2)+Cdt
a32 = D32/m
v33 = v32 - a32*t
```

```
q33 = 1/2*p*v33^2;
D33 = q33*S*(Cd+K*(n*W/q33/S)^2)+Cdt
a33 = D33/m
v34 = v33 - a33*t

q34 = 1/2*p*v34^2;
D34 = q34*S*(Cd+K*(n*W/q34/S)^2)+Cdt
a34 = D34/m
v35 = v34 - a34*t


q35 = 1/2*p*v35^2;
D35 = q35*S*(Cd+K*(n*W/q35/S)^2)+Cdt
a35 = D35/m
v36 = v35 - a35*t

t2=16.5
q36 = 1/2*p*v36^2;
D36 = q36*S*(Cd+K*(n*W/q36/S)^2)+Cdt
a36 = D36/m
v37 = v36 - a36*t2

ttotal = t*36+t2

L = 1/2*p*v37^2*cl*S
D = 1/2*p*v37^2*Cd*S
af = (L-W*sind(45))/m

t5 =137.894
h = hre - v37*sind(7)*t5 + 1/2*af*t5^2

vfin = v37 - 1/2*af*cosd(7)*t5

talll = t5+ttotal

R = ttotal*(vre+v37)/2*57.3/10

gforce = (vre-v37)/ttotal
```

Code orbit2.m is used to find the orbital velocity and fuel required

```
ra1 = 6578; %km
rp1 = ra1;
rp2 = 6378;
ra2 = 6578;
ra = 6378;
rp = ra;
eb = (ra2-rp2)/(ra2+rp2)

hb = sqrt(2*398600)*sqrt(rp2*ra2/(ra2+rp2))

h1 = sqrt(398600*ra)

va1 = hb/rp2
vb1 = hb/ra2
vb2 = h1/ra2
cv=vb2-vb1

mc = 1-exp(cv*1e3/340/9.81)
mc*21612*0.45
```

Code “Thrust2o.m“ is used to find the maximum height.

```
% Velocity at 123000 ft
% The thrust calculation for the s

M = 41612.3*0.45;       %mass in kg
g = 9.8;        %m/s^2
theta = 90;     %flight path angle
C_d = 0.03;     %co-efficient of delta
rho = 0.00001846;      %density in the altitude
V= 940;         %                  m/s
A = 81;                                    %area of the wing m^2

Isp=500;   %s

   %kg/s

mf = 21612.3*0.45;   %kg

tbo=120
t = tbo;     %burn time (designed)
me=(M-mf)/t   %kg/s

F = M*g*sind(0)+C_d*rho*V*V*A/2 + M*V/t;     %Force in newtons
F_lbs =F *0.2248                          %Thrust in lbs

F2 = M*g*sind(45)+C_d*rho*V*V*A/2 + M*V/t;     %Force in newtons
F_lbs2 =F2 *0.2248                          %Thrust in lbs

F25 = (M-mf)/2*g*sind(80)+C_d*rho*V*V*A/2 + (M-mf)/2*V/t;
F_lbs25 =F25 *0.2248
Isp=70000*4.448/me/g
c = Isp*g


vbo = (c*log(M/mf) - g/me*(M-mf))
hbo = (c/me*(mf*log(mf/M) + M - mf)-1/2*((M-mf)/me)^2*g)

tcoast=c/g*log(M/mf)-tbo  %s

n=M/mf

hmax= (1/2*c^2/g*((log(n))^2)-c*M/me*(n*log(n)-(n-1))/n)

re=6378e3

gB = g/(1+hmax/re)^2
hA =90000 %m
h2=66000
gc = g/(1+hA/re)^2

tmicro= 2*sqrt(2*(hmax-hA)/gB)
tmicro2 = sqrt(2*(hA-h2)/gc)
tmicrototal = tmicro+tmicro2

vre = -gB*(tmicro/2+tmicro2)
hre = hmax - 1/2*gB*(tmicro2)^2

Wf = 21612.3*4.448   %N

D = C_d*rho*vre*vre*A/2
a = (D + Wf*sind(theta))/mf
```

```
AR=2.4
K=1/pi/AR
```

Code “FlightPathAngle45dbld4ho.m“ is used to find the flight path.

```
%%%%%%%%%Rocket Performance Calculations%%%%%%%%%%%%%%%
function FlightPathAngle45dbld

%Givens:
w0 = 41612.3;        % Launch Weight [lbs]
wl = 21612.3;      % Landing Weight [lbs]
wf = w0 - wl;      % Fuel Weight [lbs]
S  = 00;           % Extra Fuel for Commercial W/D=9900,  Wo/D=0
w0 = w0 + S;       % Launch Weight accounting for added fuel
wf = wf + S;       % Fuel Weight with addition for Commercial
wp = 0;            % Payload Weight [lbs]
g0 = 32.174;       %[ft/s^2]
rE = 6378*3280.84;  %[ft]
% Isp = 284;         % Specific Impulse at 45000 ft for XLR-99 Engine  [sec]
T  = 70000;     % Thrust for XLR-99 at 45000 ft  [lbs]
h0 = 0;        % Launch elevation [ft]
x0 = 0;            % Initial Distance [ft]
V0 = 0;           % Initial Velocity [ft/s]
t1 = 15;           % [sec]
h  = 0;        % Ignition Elevation [ft]
gamma = 80;        % Flight Path Angle (assume constant) [degrees]
gammar = gamma*pi/180; % Flight Path Angle [radians]
D  = w0*cosd(gamma)/3.1;         % Drag Force [lbs]

hE = 360892;
gE = g0/((1+(hE/rE))^2);
g = g0/((1+(h/rE))^2);    % gravitational force at specific altitude [ft/s^2]
hI = h0-((g/2)*(t1^2));   % ignition elevation [ft]

%Calculations:
m0 = w0/32.174;     % Launch Mass[slugs]
mf = wf/32.174;     % Fuel Mass[slugs]  ***Replaced by later value***
ml = wl/32.174;     % Landing (Empty) Mass[slugs]
mp = wp/32.174;     % Payload Mass[slugs]

%Dme = T/(Isp*g);    % exhaust mass flow rate [slug/s]
tbo1 = 130;       % Burn Time from rcktperfdlbd [s]
Dme = (m0-ml)/tbo1; % Mass Flow Rate Defined by Tbo
Isp = T/(g*Dme);    % Specific Impulse Defined by Tbo and T
%mf = -tbo1*Dme+m0;  % Fuel Mass [slugs] **Corrected Value**
%wf = mf*32.174
% tbo1 = (m0-ml)/Dme; % burnout time [sec]

c = T/Dme;          % exhaust velocity (equal to Isp*g)[ft/sec]
n = m0/ml;          % mass ratio
tspan = [0, tbo1];  % Time Span

fprintf('Known Values:\n');
fprintf('Takeoff Weight =          %.2f [lbs]\n', w0);
fprintf('Landing Weight =          %.2f [lbs]\n', wl);
fprintf('Fuel Weight =             %.2f [lbs]\n', wf);
fprintf('Exhaust Mass Flow Rate =   %.2f [slugs/sec]\n', Dme);
fprintf('Time at Burnout =         %.2f [sec]\n', tbo1);
fprintf('ISP =                     %.2f [s]\n', Isp);
fprintf('Thrust Required =         %.2f [lbs]\n\n', T);
% fprintf('Fuel Weight Estimation =   %.2f [lbs]\n\n', wf);
```

```
vbocalc = (c*(log(n)))-((g0/Dme)*(m0-ml));        % burnout velocity [ft/s]
hbocalc = (c/Dme)*((ml*log(ml/m0))+m0-ml)-((1/2)*(tbo1^2)*g0*sind(gamma));   %burnout height [ft]

% Apex Height
b = vbocalc*sin(gammar);% Quadtratic Equation
tA = (-b - (b^2-4*0*gE/2)^(1/2))/(2*-gE/2);    % gEntry is used since accurate height isn't known yet
hA1 = b*tA/2 - gE/2*(tA/2)^2;
hmax1 = hbocalc + hA1;  % Max height using no Drag after burnout
hA = hmax1;
gA = g0/((1+(hA/rE))^2);

% Re-entry Velocity
hRe = hE - hbocalc;
tE = (-b - (b^2-4*hRe*gE/2)^(1/2))/(2*-gE/2);
VE = b - gE*tE;

fprintf('Values Calculated from Rocket Equations\n');
fprintf('Burnout Altitude =        %.2f [ft]\n', hbocalc);
fprintf('Burnout Velocity =        %.2f [ft/s]\n', vbocalc);
fprintf('Maximum Altitude =        %.2f [ft]\n', hA);
% fprintf('Reentry Altitude =        %.2f [ft]\n', hE);
fprintf('Reentry Velocity =        %.2f [ft/s]\n\n', VE);


IC = [V0; x0; hI];
% Velocity Distance Altitude %
[t,f] = ode45(@rates, tspan, IC);

v = f(:,1);
x = f(:,2);
h = f(:,3);
function dydt = rates(t,y)
dydt = zeros(3,1);
v = y(1);
x = y(2);
h = y(3);

   if t < tbo1
      m = m0 - Dme*t;
      T = T;
   else
      m = m0 - Dme*tbo1;
      T = 0;
   end

v_dot = T/m - D/m - g*sin(gammar);
x_dot = (rE/(rE + h)*v*cos(gammar));
h_dot = v*sin(gammar);

dydt(1) = v_dot;
dydt(2) = x_dot;
dydt(3) = h_dot;
end

xsize = size(x);
hsize = size(h);
vsize = size(v);
xbo = x(xsize(1));     % Taken from ode45 function, final value
vbo = v(vsize(1));     % Taken form ode45 funciton, final value
hbo = h(hsize(1));     % Taken form ode45 function, final value

hRe = hE - hbo;
```

```
cE = -hRe;
bE = vbo*sin(gammar);
aE = -gE/2;
TABE1 = (-bE-(bE^2-4*aE*cE)^(1/2))/(2*aE);
TABE2 = (-bE+(bE^2-4*aE*cE)^(1/2))/(2*aE);
if TABE2 > TABE1;
   TABE = TABE2;
else TABE = TABE1;
end

TABEALL = (-bE-(bE^2-4*0*aE)^(1/2))/(2*aE);

% ***After Burnout to Re-Entry***
% With gEntry & Integrated hbo
tAB = 0:0.5:TABE;   % Time Period After Burnout to Entry
tABsize = size(tAB);% Time Length for Re-Entry Velocity
Tsize = size(tAB);  % Time Length for Q Calculations
xAB = xbo + vbo.*cos(gammar).*tAB;
yAB = hbo + vbo.*sin(gammar).*tAB - gE./2.*tAB.^2; % gEntry is used instead of actively changing Gravity
yABhalf = hbo + vbo.*sin(gammar).*(TABEALL/2) - gE./2.*(TABEALL/2).^2;
vxAB = vbo.*cos(gammar);
vyAB = vbo.*sin(gammar) - gE.*tAB;

% ***After Burnout to Re-Entry***
% With gApex & Integrated hbo
xAB2 = xbo + vbo.*cos(gammar).*tAB;
yAB2 = h(hsize(1)) + vbo.*sin(gammar).*tAB - gA./2.*tAB.^2; % gApex is used instead of actively changing Gravity
yABhalf2 = hbo + vbo.*sin(gammar).*(TABEALL/2) - gA./2.*(TABEALL/2).^2;
vxAB2 = vbo.*cos(gammar);
vyAB2 = vbo.*sin(gammar) - gA.*tAB;

% Averaging the Values to account for Gravity Difference
xABa = (xAB + xAB2)./2;
yABa = (yAB + yAB2)./2;
vxABa = (vxAB + vxAB2)./2;      % x-direction velocity
vyABa = (vyAB + vyAB2)./2;      % y-direction velocity
VAB = sqrt(vxABa.^2+vyABa.^2);  % tangent velocity in direction of flight path
yMAX = (yABhalf + yABhalf2)/2;

% figure
plot(x,h, 'k', 'linewidth', 2)
hold on
plot(xABa, yABa, 'c','linewidth', 2)
title('Flight Path With Respect to Climb Angle')
xlabel('Horizontal Distance, ft')
ylabel('Altitude, ft')
legend('Thrusted Phase at 80 Degrees','After Burnout','Thrusted Phase at 60 Degrees','After Burnout','Thrusted Phase at 55 Degrees','After Burnout','Thrusted Phase at 57 Degrees','After Burnout','Thrusted Phase at 59 Degrees','After Burnout','Location','southeast')

% ***Gravity Comparison***
% figure
% plot(xAB, yAB, 'b')
% hold on
% plot(xAB2, yAB2, 'g')
% plot(xABa, yABa, 'k')
% title('Comparison of Gravity Values')
% xlabel('Horizontal Distance, ft')
% ylabel('Altitude, ft')
% legend('gEntry','gApex','gAverage', 'Location','southeast')
% axis([330000, 410000, 293000 300000])
```

```
fprintf('Interated Values Using ode45\n');
fprintf('Distance at Burnout =      %.2f [ft]\n', x(xsize(1)));
fprintf('Alititude at Burnout =     %.2f [ft]\n', h(hsize(1)));
fprintf('Velocity at Burnout =      %.2f [ft/s]\n', v(vsize(1)));
fprintf('Altitude at Apex =         %.2f [ft]\n', yMAX)
% fprintf('Altitude at Re-Entry =     %.2f [ft]\n', hE);
fprintf('Velocity at Re-Entry =     %.2f [ft/s]\n', VAB(tABsize(2)));
fprintf('Time after Burnout =       %.1f [s]\n\n\n', TABE);


end
```

Code “envelopes42o.m“ is used to find the flight envelope.

```
hi = 0;             %m
hf = 190000;        %m
wi = 41612.3*4.448;       %N
wf = 21612.3*4.448;   %N
p0 = 1.225;                     %kg/m^3
S  = 81;                        %m^2

h = linspace(hi,hf,52);
W = linspace(wi,wf,52);
y=[1,0.8249,0.6744,0.5459,0.4371,0.3458,0.2642,0.1941,0.1427,0.1049,0.0772,0.0567,0.0417,0.0307,0.0226,0.0166,0.
0122,0.0090,0.0066,0.0049,0.0036,0.0027,0.0020,0.0014,0.0011,7.8452e-04,5.7863e-04,4.2686e-04,3.1495e-
04,2.3243e-04,1.7156e-04,1.2665e-04,9.3518e-05,6.9065e-05,5.1016e-05,3.7690e-05,2.7851e-05,2.0584e-05,1.5216e-
05,1.1249e-05,8.3187e-06,6.1527e-06,4.5514e-06,3.3675e-06,2.4920e-06,1.8445e-06,1.3655e-06,1.0110e-06,7.4874e-
07,5.5459e-07,4.1086e-07,3.0443e-07];
% y from geom224.m
p=y*p0;

Clmax=linspace(16,16/3,52);
Vstall = sqrt(W/S*2./(p.*Clmax));
b = [355.3 347.9 340.3 332.5 324.6 316.5 308.1 299.5 295.1 295.1 295.1 295.1 295.1 296.4 297.7 299.1 300.4 301.7
303 306.5 310.1 313.7 317.2 320.7 324.1 327.5 329.8 329.8 328.8 325.4 322 318.6 315.1 311.6 308 304.4 300.8 297.1
293.4 290.8 288 285.3 282.5 279.8 276.9 274 274 274 274.1 274.7 275.8 277.6];
%b from geom224.m;
M = Vstall./b

hold

plot(M(1,1:39),h(1,1:39),'g')

xlabel('Mach number')
ylabel('height (m)')


q = 1800*4.88;                  %kg/m^2
V2 = sqrt(2*q./p);

M2 = V2./b

plot(M2(1,1:25),h(1,1:25),'b')
hold

T = 311.36e3;                   %N
alpha = 10;         %degrees
y0 = 90;            %degrees
cd0 = 0.03;
AR = 2.4;
e = 1;
```

```
Cl = Clmax;           %estimate

K = 1/(pi*AR*e);
cd = cd0 + K.*Cl.^2;
D = T*cosd(alpha) - W*sind(y0);
V25 = sqrt(2*D./S./p/cd);

M25=V25./b
hold
plot(M25(1,20:39),h(1,20:39),'r')

w = 1000+459.67; %deg. R
em = 0.8;       %emissivity
vsb = 0.481e-12; %constant
Cf = 0.0008;    %estimate

qs = w^4*em*vsb

v4 = (qs/3.21e-4/Cf./p).^(1/3)*0.3048;

M4 = v4./b+9

plot(M4(1,20:35),h(1,20:35), 'm')
hold
```

Code “turno2.m“ is used to find the banking angle and turning time.

```
vre = 2.1e3;  %m/s
ph = 30;
W = 21612.3*4.448; %N
g=9.81
Cd = 0.03;
Cdt = 0.001;
S = 81;
AR = 2.4;
cl = 1.6

K=1/pi/AR;

n=1/cosd(ph);

p = 0.00196265;
q = 1/2*p*vre^2;

D = q*S*(Cd+K*(n*W/q/S)^2)+Cdt

m=W/g;
a = D/m
t=25;
hre = 66000;

v1 = vre - a*t

q1 = 1/2*p*v1^2;
D1 = q1*S*(Cd+K*(n*W/q1/S)^2)+Cdt
a1 = D1/m
v2 = v1 - a1*t

q2 = 1/2*p*v2^2;
D2 = q2*S*(Cd+K*(n*W/q2/S)^2)+Cdt
a2 = D2/m
```

```
v25 = v2 - a2*t

q25 = 1/2*p*v25^2;
D25 = q25*S*(Cd+K*(n*W/q25/S)^2)+Cdt
a25 = D25/m
v4 = v25 - a25*t

q4 = 1/2*p*v4^2;
D4 = q4*S*(Cd+K*(n*W/q4/S)^2)+Cdt
a4 = D4/m
v5 = v4 - a4*t

q5 = 1/2*p*v5^2;
D5 = q5*S*(Cd+K*(n*W/q5/S)^2)+Cdt
a5 = D5/m
v6 = v5 - a5*t

q6 = 1/2*p*v6^2;
D6 = q6*S*(Cd+K*(n*W/q6/S)^2)+Cdt
a6 = D6/m
v7 = v6 - a6*t

q7 = 1/2*p*v7^2;
D7 = q7*S*(Cd+K*(n*W/q7/S)^2)+Cdt
a7 = D7/m
v8 = v7 - a7*t

q8 = 1/2*p*v8^2;
D8 = q8*S*(Cd+K*(n*W/q8/S)^2)+Cdt
a8 = D8/m
v9 = v8 - a8*t

q9 = 1/2*p*v9^2;
D9 = q9*S*(Cd+K*(n*W/q9/S)^2)+Cdt
a9 = D9/m
v10 = v9 - a9*t

q10 = 1/2*p*v10^2;
D10 = q10*S*(Cd+K*(n*W/q10/S)^2)+Cdt
a10 = D10/m
v11 = v10 - a10*t

q11 = 1/2*p*v11^2;
D11 = q11*S*(Cd+K*(n*W/q11/S)^2)+Cdt
a11 = D11/m
v12 = v11 - a11*t

q12 = 1/2*p*v12^2;
D12 = q12*S*(Cd+K*(n*W/q12/S)^2)+Cdt
a12 = D12/m
v125 = v12 - a12*t

q125 = 1/2*p*v125^2;
D125 = q125*S*(Cd+K*(n*W/q125/S)^2)+Cdt
a125 = D125/m
v14 = v125 - a125*t

q14 = 1/2*p*v14^2;
D14 = q14*S*(Cd+K*(n*W/q14/S)^2)+Cdt
a14 = D14/m
v15 = v14 - a14*t
```

```
q15 = 1/2*p*v15^2;
D15 = q15*S*(Cd+K*(n*W/q15/S)^2)+Cdt
a15 = D15/m
v16 = v15 - a15*t

q16 = 1/2*p*v16^2;
D16 = q16*S*(Cd+K*(n*W/q16/S)^2)+Cdt
a16 = D16/m
v17 = v16 - a16*t

q17 = 1/2*p*v17^2;
D17 = q17*S*(Cd+K*(n*W/q17/S)^2)+Cdt
a17 = D17/m
v18 = v17 - a17*t

q18 = 1/2*p*v18^2;
D18 = q18*S*(Cd+K*(n*W/q18/S)^2)+Cdt
a18 = D18/m
v19 = v18 - a18*t

q19 = 1/2*p*v19^2;
D19 = q19*S*(Cd+K*(n*W/q19/S)^2)+Cdt
a19 = D19/m
v20 = v19 - a19*t

q20 = 1/2*p*v20^2;
D20 = q20*S*(Cd+K*(n*W/q20/S)^2)+Cdt
a20 = D20/m
v21 = v20 - a20*t

q21 = 1/2*p*v21^2;
D21 = q21*S*(Cd+K*(n*W/q21/S)^2)+Cdt
a21 = D21/m
v22 = v21 - a21*t

q22 = 1/2*p*v22^2;
D22 = q22*S*(Cd+K*(n*W/q22/S)^2)+Cdt
a22 = D22/m
v23 = v22 - a22*t

q23 = 1/2*p*v23^2;
D23 = q23*S*(Cd+K*(n*W/q23/S)^2)+Cdt
a23 = D23/m
v24 = v23 - a23*t

q24 = 1/2*p*v24^2;
D24 = q24*S*(Cd+K*(n*W/q24/S)^2)+Cdt
a24 = D24/m
v25 = v24 - a24*t

q25 = 1/2*p*v25^2;
D25 = q25*S*(Cd+K*(n*W/q25/S)^2)+Cdt
a25 = D25/m
v26 = v25 - a25*t

q26 = 1/2*p*v26^2;
D26 = q26*S*(Cd+K*(n*W/q26/S)^2)+Cdt
a26 = D26/m
v27 = v26 - a26*t


q27 = 1/2*p*v27^2;
```

```
D27 = q27*S*(Cd+K*(n*W/q27/S)^2)+Cdt
a27 = D27/m
v28 = v27 - a27*t

q28 = 1/2*p*v28^2;
D28 = q28*S*(Cd+K*(n*W/q28/S)^2)+Cdt
a28 = D28/m
v29 = v28 - a28*t

q29 = 1/2*p*v29^2;
D29 = q29*S*(Cd+K*(n*W/q29/S)^2)+Cdt
a29 = D29/m
v30 = v29 - a29*t

q30 = 1/2*p*v30^2;
D30 = q30*S*(Cd+K*(n*W/q30/S)^2)+Cdt
a30 = D30/m
v31 = v30 - a30*t

q31 = 1/2*p*v31^2;
D31 = q31*S*(Cd+K*(n*W/q31/S)^2)+Cdt
a31 = D31/m
v32 = v31 - a31*t

q32 = 1/2*p*v32^2;
D32 = q32*S*(Cd+K*(n*W/q32/S)^2)+Cdt
a32 = D32/m
v33 = v32 - a32*t


q33 = 1/2*p*v33^2;
D33 = q33*S*(Cd+K*(n*W/q33/S)^2)+Cdt
a33 = D33/m
v34 = v33 - a33*t

q34 = 1/2*p*v34^2;
D34 = q34*S*(Cd+K*(n*W/q34/S)^2)+Cdt
a34 = D34/m
v35 = v34 - a34*t

t2=7
q35 = 1/2*p*v35^2;
D35 = q35*S*(Cd+K*(n*W/q35/S)^2)+Cdt
a35 = D35/m
v36 = v35 - a35*t2

ttotal = t*35+t2

L = 1/2*p*v36^2*cl*S
D = 1/2*p*v36^2*Cd*S
af = (L-W*sind(45))/m

t5 =137.782
h = hre - v36*sind(7)*t5 + 1/2*af*t5^2

vfin = v36 - 1/2*af*cosd(7)*t5

talll = t5+ttotal
R = ttotal*(vre+v36)/2*57.3/10

gforce = (vre-v36)/ttotal
```

Code “orbit.m“ is used to find the orbital velocity and fuel required.

```
ra1 = 6538; %km
rp1 = ra1;
rp2 = 6378;
ra2 = 6538;
ra = 6378;
rp = ra;
eb = (ra2-rp2)/(ra2+rp2)

hb = sqrt(2*398600)*sqrt(rp2*ra2/(ra2+rp2))

h1 = sqrt(398600*ra)

va1 = hb/rp2
vb1 = hb/ra2
vb2 = h1/ra2
cv=vb2-vb1

mc = 1-exp(cv*1e3/340/9.81)
mc*21612*0.45
```

REFERENCES

[1] Jenkins, D.R., "Hypersonics Before the Shuttle: A Concise History of the X-15 Research Airplane," NASA SP-2000- 4518, June 2000.

[2] Yancey, R.B., "Flight Measurements of Stability and Control Derivatives of the X-15 Research Airplane to a Mach Number of 6.02 and an Angle of Attack of 25º," NASA TN D-2532, Nov. 1964.

[3] O'Donoghue, D., "Stella Nova Aeronatics: Reverse Engineering the X - 15," MAE 4350-001, Fall 2014.

[4] "Our Vehicles and Spaceships," *Virgin Galactic*, URL: http://www.virgingalactic.com/human-spaceflight/our-vehicles/, 2015.

[5] "About Blue Origin," *Blue Origin,* URL: http://www.blueorigin.com/about, 2015.

[6] "Ascender," *Bristol Spaceplanes,* URL: http://bristolspaceplanes.com/projects/ascender/, 2015.

[7] "Lynx Suborbital Spacecraft," *XCOR Aerospace,* URL: http://www.xcor.com/lynx/, 2014.

[8] "NASA Armstrong Fact Sheet: X-15 Hypersonic Research Program," *NASA*, URL: http://www.nasa.gov/centers/armstrong/news/FactSheets/FS-052-DFRC.html, Feb. 2014

[9] "X-15: To the Edge of Space," *NASA,* URL: http://www.nasa.gov/pdf/366588main_X-15_Poster.pdf, 2008.

[10] "Reusable Launch Vehicle Programs and Concepts," *Associate Administrator for Commercial Space Transportation (AST),* Jan. 1998.

[11] "Suborbital Reusable Launch Vehicles and Emerging Markets," *Federal Aviation Administration Office of Commercial Space Transportation,* Feb. 2005.

[12] Nicolai, L.M., and Carichner, G.E., "Airship Design," *Fundamentals of Aircraft and Airship Design,*edited by J.A. Shetz, Vol. 1, AIAA, VA, 2010.

[13] Phillips, W.F., *Mechanics of Flight,* 2nd ed., John Wiley & Sons, Hoboken, NJ, 2010.

[14] Curtis, H.D., *Orbital Mechanics for Engineering Students,* 2nd ed., Elsevier Ltd., Burlington, MA, 2009.

[15] Seedhouse, E., *Suborbital Industry at the Edge of Space,* Springer International Publishing, Switzerland, 2014.

[16] Mullins, B. R., Jr., Ph.D., "U.S. Standard Atmosphere," *Development of Standard Model*, MAE 3306-001, Spring 2014.

[17] Previous Performance team equations.

[18] "Limit Maneuvering Load Factors," Part 25 Subpart C Sec. 25.337, FAA, URL: http://rgl.faa.gov/Regulatory_and_Guidance_Library/rgFAR.nsf/0/5d3cea6c015dda8685256672004ec387!OpenDocument&ExpandSection=2#_Section2, May 1970.

[19] "Low Earth Orbit," *Merriam - Webster Dictionary,* URL: http://www.merriam-webster.com/dictionary/ low%20earth%20orbit, 2015.

BIOGRAPHICAL INFORMATION

Deirdra is an aerospace engineer who looks forward to applying the knowledge acquired from all of the courses attended in the pursuit of this degree. The experiences procured during the course of this team project will be instrumental in future endeavors. Opportunities pursuing further education will be explored, and futher experience in aerospace related industries will be realized.