

Final Project

Jacob Lancaster¹, Sukhyeon Lim¹, Samuel Rockaway¹

¹[Department of Aerospace Engineering, University of Michigan](#)

Abstract. This report details the design and analysis of a multi-stage launch vehicle capable of meeting specific mission requirements including placing a spacecraft into lunar orbit, transferring cargo, and returning to low earth orbit. Using a top-down approach, the design began with the upper stage, working down towards the first stage. Each stage was designed using an iterative approach, starting with chamber pressure and nozzle throat area. Programs such as NASA's combustion equilibrium analysis tool and the provided 2-D method of characteristics tool were leveraged to obtain performance metrics and iterate on the design. Key assumptions, such as frozen combustion and 2-D flow, provide simplifications but add limitations to the design. Some challenges faced included nozzle performance at different altitudes, which emphasize the importance for refinement in future designs. The design presented in this report provides a strong foundation for future work and emphasizes the importance of addressing assumptions.

1. Introduction

This report presents the comprehensive design and analysis of a multi-stage launch vehicle developed to meet specific mission requirements. The objective of this project is to design a vehicle capable of placing a spacecraft into Low Earth Orbit (LEO), where the spacecraft will then transfer to lunar orbit, transfer cargo, and return to LEO with an equivalent payload. A diagram of the planned mission can be found below in Figure 1. Another goal of this design is to maximize performance metrics. The overall approach for the design process can be considered a top down method, starting with the design of the upper stage and moving down to the design of lower stages. This rational can be better shown as a flow chart in Figure 2. In brief, the design process begins with the upper stage, which has a known payload mass and ΔV outlined in the mission requirements. Stemming from the mission requirements, suitable propellant combinations are chosen, enabling the calculation of chamber conditions using NASA's Combustion Equilibrium Analysis (CEA). These chamber conditions directly impact the performance metrics of the minimum-length nozzle design. Once the performance metrics are established, the mass and payload ratios for the upper stage are calculated. This process is then applied iteratively to the lower stages, where each stage is designed to support the payload mass of the stages above it, ensuring a cohesive and optimized vehicle configuration. The sections below discuss the design process in detail, and present how each element of the final launch vehicle was developed.

2. Launch Vehicle Configuration

The optimal configuration of the multi-stage launch vehicle consists of three stages (i.e. 1st stage, 2nd stage, and upper stage). Clarifying some terminology, the wet mass of the stage refers to the sum of the payload, propellant, and structural masses. Similarly, the dry mass, or burnout mass, of the stage refers to the sum of the payload and structural masses. Here the structural mass of the stage refers to the mass of the propellant tanks, rocket engine, avionics, and any other components that are not fuel or payload on that stage. The propellant mass of the stage is solely the mass of the fuel and oxidizer for that stage. The payload of the stage is the mass of the payload that the stage is carrying. It is crucial to recognize that a stage's payload encompasses the total mass of all components above it. For example, the 1st stage's payload mass is the wet mass of both the 2nd and upper stages.

Each stage of the multi-stage vehicle has a ΔV that it must achieve. For this mission, the 1st and 2nd stages are responsible for transporting the spacecraft to LEO, which yield a combined ΔV of $8.0 \frac{km}{s}$. This is Maneuver ID 1 in Figure 3 below. The ΔV that the 1st stage and 2nd stage will experience was derived from the optimization seen in later sections of this report. From this optimization, it was determined that the ΔV of the first and second stage are $5.0 \frac{km}{s}$ and $3.0 \frac{km}{s}$ respectively. As for the upper

stage, it must travel from LEO to the moon and back to LEO, which yields a ΔV of $9.6 \frac{km}{s}$. This is Maneuver ID 2-10 in Figure 3.

The engine performance metrics of each stage are also derived from the analysis below. With the knowledge of the ΔV and the performance metrics required for each stage, specifically the specific impulse (Isp), the mass ratios and quantities can be calculated for each stage. In general, Equation 1 was used to determine the mass ratio, R, using the rocket equation. Knowing that the payload is the weight of everything above the stage, the payload ratio can be calculate using Equation 2. Note that the payload equation requires the structural coefficient, which is determined from Figure 4 using the fuel for each stage. With both the wet-to-burnout mass ratio and payload ratio, Equation 4 was used to solve for the wet, or initial, mass of the rocket. Plugging this wet mass back into the calculation for the mass ratio, R, the dry, or final mass can be determined. Both the structural mass and propellant mass can be solved by using Equations 5 and 6 respectively.

The engine specifications, number of engines, and configuration of engines for each stage is detailed in later sections of this report. The optimum values for wet mass, dry mass, structural mass, propellant mass, number of engines, engine specifications, and engine configuration can be found below in Table 1.

3. Rocket Design for First Stage

The propellants used for the first stage are Liquid Oxygen (LOX) and Liquid Methane (LCH₄). This propellant combination is renowned for its optimality of having both decent specific impulse and thrust, able to be perform as an all-rounder including the first stage. Compared to LOX/RP-1 propellants combination, LOX/LCH₄ has higher specific impulse, guaranteeing better efficiency. Also, LOX/CH₄ combination can have higher thrust-to-weight ratio, more suitable for the first stage. It is also assumed that the combustion chamber pressure p_0 is 30 MPa. In modern engines, such as SpaceX's raptor vacuum engine, the chamber pressure is approximately 30 MPa, which makes this assumption valid [1].

Using the assumed fuels and combustion chamber pressure, the other conditions inside of the combustor can be described using NASA's Combustion Equilibrium Analysis (CEA) tool. The 'hp' solution is used which calculates the combustion equilibrium conditions where enthalpy and pressure are known. This approach is commonly used to model steady-state combustion processes in rocket engines, where the chamber pressure is known, and the enthalpy is determined by the energy released from the combustion of a specific propellant mixture. CEA requires the initial temperatures of the propellants, so 110- and 90-degrees Kelvin were chosen for LCH₄ and LOX respectively. These values were determined by analyzing the boiling point of both propellants [2]. The final input that CEA requires an oxidizer-to-fuel ratio, or mixture ratio, to calculate chamber conditions. Figure 5 depicts the specific impulse for LOX/LCH₄ plotted as a function of mixture ratio. The performance metrics for this figure were calculated with the assumption of ideal nozzle using the chamber pressure defined above. To maximize performance metrics for fully expanded nozzle, a mixture ratio of 3.21 will be used. With the mixture ratio, chamber pressure, and propellant temperature known, the CEA results for adiabatic flame temperature, ratio of specific heats, and molecular weight of the products for the first stage are tabulated in Table 2 below.

From the selected propellant combination, Figure 6 can be used to determine the characteristic length of the combustion chamber. The LOX/CH₄ combination gives a heritage characteristic length, L^* , range of 80 – 300 centimeters [3]. As an initial assumption of 190 centimeters will be used. The combustion chamber can continue to be described by selecting the chamber-to-throat area ratio. Figure 7 plots chamber-to-throat area ratio, $\frac{A_c}{A_t}$, as a function of chamber-to-throat pressure ratio, $\frac{p_0}{p_t}$, and chamber mass, m_c . The pressure losses across the combustion chamber occur from Rayleigh heat addition. This

pressure drop hinders performance, which drives the decision to a large $\frac{A_c}{A_t}$. However, at large $\frac{A_c}{A_t}$, m_c is nearly doubled, which drives the decision to a smaller $\frac{A_c}{A_t}$. At a chamber-to-throat area ratio of 2.5, Rayleigh losses are minimized while having a respectable m_c . With both $\frac{A_c}{A_t}$ and L^* known, Equation 7 can be used to calculate the length of the combustion chamber, L_c . Chamber thickness was derived by utilizing the hoop stress equation shown in Equation 8. From research on other rocket engines the selected wall material is 70/30 tungsten copper with a yield strength of approximately 500 MPa. Given this yield strength and the pressure of the chamber, the calculated minimum wall thickness is 5mm.

Assuming the throat area of nozzle to be 0.1 square meters, the mass flow rate through the nozzle can be calculated using Equation 9. This mass flow rate is very important for performance metric calculations and determining the injector design. For this system, a doublet impinging stream pattern will be used. Specifically, unlike doublet patterns will be used to promote mixing between fuel and oxidizer. In this setup, it is imperative that the combined streams cancel the tangential momentum to prevent backsplash onto the injector. To increase the discharge coefficient, a short tube with a rounded entrance will be used for the orifice type. From Figure 8, this yields a discharge coefficient, C_d , of 0.90 and an orifice diameter of 1.57 millimeters. To calculate mass flow rate of each injector, the pressure drop across the tube must be determined. As a rule of thumb, the pressure drop must be a minimum of 20% of the combustion chamber pressure [3]. The pressure drop ratio required for the fuel and oxidizer to get a mixture ratio of 3.21 can be calculated using Equation 10. Holding the fuel pressure drop at 20% of the combustion chamber pressure, the oxidizer pressure drop is 50% of the combustion chamber pressure. Now the mass flow rate and velocity of each injector can be calculated using Equation 11 and 12. The angle of impingement that will cancel the tangential flow can be found using Equation 13. The oxidizer angle is 10 degrees, and the fuel angle is 43 degrees. With the flow rates of each injector known, the total number of injectors required to achieve the total mass flow rate can be calculated using Equation 14. The total number of doublet impinging sets to achieve the required flow rate and mixture ratio is 2,998. All injector quantities of interest can be found below in Table 3. A to-scale schematic of the injector design is shown in Appendix A.

With the combustor and injectors fully defined, the design of the diverging portion of the nozzle can be facilitated. The ambient pressure varies with altitude, which directly influences the optimal area ratio for an ideal nozzle. This area ratio is determined based on the altitude where the nozzle is expected to achieve its maximum performance, as represented by the thrust coefficient, C_f . Using Equations (16) and (17), the area ratio for a specific ambient pressure and engine chamber pressure can be calculated. For the first stage, the nozzle's area ratio is set to 35 to maximize performance at sea level, aligning with the ambient pressure at this altitude. Although the nozzle does not achieve peak efficiency at other altitudes, its performance remains relatively close to ideal within the first stage's operational range, shown in Figure 9. This design ensures that the first stage operates effectively across the altitudes it encounters, maintaining strong performance even when conditions vary from sea level. Using this assumed exit-to-throat area ratio, a first pass of the performance metrics can be conducted. For this first pass, a very simple geometry of the nozzle, described in Equation 15, is used. Using the defined area ratios along the nozzle and the ratio of specific heat given by CEA, the Mach-Area relation, shown in Equation 16, can be used to solve for the Mach number anywhere along the nozzle. To ensure optimal performance, a bell-shaped nozzle geometry is selected due to its empirically proven advantages, including efficient expansion of exhaust gases, maximized thrust, and superior structural integrity. These characteristics make the bell-shaped nozzle ideal for optimizing rocket performance. The nozzle geometry (Figure 10) is designed based on the specified throat area and area ratio, utilizing parametric equations and quadratic Bézier curves. These curves determine the critical wall angles through the interpolation of empirical data, guided by the specified area ratios and throat radius [4,5].

The first-stage nozzle is designed to operate optimally at sea level. This means the nozzle operates in an under expanded condition for most of its operational duration. Consequently, the pressure and

temperature at any point along the nozzle can be calculated isentropically using Equations 17 and 18, respectively. With known temperatures along the nozzle, the speed of sound at any point can be calculated using Equation 19. The Mach number and speed of sound can be related to velocity using Equation 20. Now that the exit pressure and velocity of the nozzle is known, the thrust, thrust coefficient, and Isp can be found using Equations 21 – 23. The performance metrics of the first stage nozzle can be found below in Table 4. This specific impulse calculated here was used above to calculate the wet and dry mass of the first stage.

The thrust-to-weight ratio is a critical parameter used to determine the number of engines required to successfully propel the first stage. Based on Equation 24, the thrust-to-weight ratio for the first stage is calculated to be 0.45. To ensure a stage thrust-to-weight ratio greater than 1, three engines are allocated for the first stage. For symmetry and ease of gimbaling, the engines are arranged in a symmetrical configuration, spaced 120 degrees apart.

Since the combustion chamber pressure is the maximum pressure experienced in a rocket engine, the nozzle thickness is equivalent to the chamber wall thickness for simplicity. Due to the low nozzle mass and relatively small size, there is no need to trim this nozzle below the minimum length determined. Therefore, this nozzle is not considered lossy. The nozzle is drawn to-scale in and pictured in Appendix A below.

Using the propellant mass for the first stage obtained in section 2, the mass of the fuel and oxidizer tanks can be solved for. The mixture ratio can be used to determine how much of the propellant mass is dedicated to fuel and oxidizer. The mass of oxidizer and fuel can be determined using Equation 25. The shape of the fuel storage tanks was designed to be simple cylinders. Though spherical tanks offer better total volume and heat transfer properties, it was ultimately determined that the spherical tanks would be too wide to be practical for this mission. With the mass and density of both propellants known, the volume of the propellants can be calculated using Equation 26. The height of the oxidizer tank is assumed to be 15 meters and the height of the fuel tank is assumed to be 10 meters. The diameter of each tank can be calculated using Equation 27. For this stage, a turbopump feed system will be used to transport propellants from the storage tanks to the combustion chamber. This design allows for low propellant tank pressures, so the internal pressure of each tank can be assumed to be 0.3 MPa. For reference, the typical propellant temperature tank pressure in turbopump feed systems lies between 0.07 MPa and 0.34 MPa [2]. Because this setup allows for low storage pressure, it is not necessary to use rounded caps on the cylindrical storage tanks. Using internal pressure, the wall thickness can be estimated by rearranging the hoop stress formula shown in Equation 8. The material chosen for these tanks is aluminum 2219, which has a yield stress of 310 MPa. This material has a similar density to the wall density given in the mission statement and is non-reactive with the propellants, making it a great option. This yields a fuel tank wall thickness of 3.2 millimeters and an oxidizer tank thickness of 2.9 millimeters. From this, the mass of each empty storage tank can be calculated using Equation 28. The empty fuel tank and oxidizer tank have a mass of approximately 4037.46 kg and 4997.82 kg respectively. All fuel and oxidizer tank properties can be found below in Table 5.

Performing a quick check, the combined mass of the nozzle, empty oxidizer and empty fuel tank is roughly 9000 kg. The structural mass in section 1 was calculated to be approximately 19717.97 kg. This indicates that there are 10000 kg to be dedicated to the structural mass of the rocket body, avionics, insulation, turbo pump and piping necessary. The rocket body was made large enough to enclose both propellant tanks and support the payload, i.e. cargo bay and actual payload. The material used for the structure was again chosen to be aluminum 2219 because of its similar density to the one in the specified mission statement. The body thickness was designed to withstand maximum aerodynamic loading, also known as “max Q”, without deformation. During max Q, the rocket can expect to see about 0.03 MPa of aerodynamic loading. This load is far less than the yield strength of the aluminum alloy chosen. For this reason, the wall thickness was calculated to minimize weight. The wall thickness of the first stage was calculated using Equation 8. This extra wiggle room still left in the structural mass can be used for

excess propellant as a safety factor. Because both propellants are cryogenic liquids, it is possible that they will partially vaporize, causing a loss of useable fuel. The to-scale drawings of the propellant tanks and rocket structure can be shown in Appendix A. All structural masses are also reported below in Table 1.

This concludes the analysis of the first stage, which served as a critical step in enabling a successful evaluation of the lower stages. By completing this stage of the analysis, a solid foundation was established for understanding and addressing the performance requirements of the subsequent stages.

4. Rocket Design for Second Stage

Moving on to the second stage, assume the propellants used are Liquid Oxygen (LOX) and Liquid Methane (CH₄) [3]. This propellant combination was selected for its capability to create significant ΔV for a low structural coefficient. For simplicity, all stages have a chamber pressure p_0 , of 30 MPa. Justification for this value was given in the previous.

Using the assumed fuels and combustion chamber pressure, the other conditions inside of the combustor can be described using NASA's Combustion Equilibrium Analysis (CEA) tool. The 'hp' solution is used which calculates the combustion equilibrium conditions where enthalpy and pressure are known. This approach is commonly used to model steady-state combustion processes in rocket engines, where the chamber pressure is known, and the enthalpy is determined by the energy released from the combustion of a specific propellant mixture. CEA requires the initial temperatures of the propellants, so 90- and 111.7-degrees Kelvin were chosen for LOX and CH₄ respectively. These values were determined by analyzing the boiling point of both propellants [2]. The final input that CEA requires is an oxidizer-to-fuel ratio, or mixture ratio, to calculate chamber conditions. Figure 5 depicts the specific impulse plotted as a function of mixture ratio. The performance metrics for this figure were calculated at specific altitude conditions using the chamber pressure defined above, and a mixture ratio of 3.21 will be used. With the mixture ratio, chamber pressure, and propellant temperature known, the CEA results for adiabatic flame temperature, ratio of specific heats, and molecular weight of the products for the second stage are tabulated in Table 2 below.

From the selected propellant combination, a section within Sutton can be used to determine the characteristic length of the combustion chamber. The LOX/LCH₄ combination gives a heritage characteristic length, L^* , range of 80 – 300 centimeters. As an initial assumption, L^* of 190 centimeters will be used since it is halfway between the recommended values. The combustion chamber can continue to be described by selecting the chamber-to-throat area ratio. Figure 7 plots chamber-to-throat area ratio, $\frac{A_c}{A_t}$, as a function of chamber-to-throat pressure ratio, $\frac{p_0}{p_t}$, and chamber mass, m_c . The pressure losses across the combustion chamber occur from Rayleigh heat addition. This pressure drop hinders performance, which drives the decision to a large $\frac{A_c}{A_t}$. However, at large $\frac{A_c}{A_t}$, m_c is nearly doubled, which drives the decision to a smaller $\frac{A_c}{A_t}$. At a chamber-to-throat area ratio of 3, Rayleigh losses are minimized while having a respectable m_c . With both $\frac{A_c}{A_t}$ and L^* known, Equation 7 can be used to calculate the length of the combustion chamber, L_c . Chamber thickness was derived by utilizing the hoop stress equation shown in Equation 8. From research on other rocket engines the selected wall material is 70/30 tungsten copper with a yield strength of approximately 500 MPa. Given this yield strength and the pressure of the chamber, the calculated minimum wall thickness is 8mm.

Assuming the throat area of nozzle to be 0.02 square meters, the mass flow rate through the nozzle can be calculated using Equation 9. This mass flow rate is very important for performance metric calculations and determining the injector design. For this system, a doublet impinging stream pattern will be used. Specifically, unlike doublet patterns will be used to promote mixing between fuel and oxidizer. In this

setup, it is imperative that the combined streams cancel the tangential momentum to prevent backsplash onto the injector. To increase the discharge coefficient, a short tube with a rounded entrance will be used for the orifice type. From Figure 8, this yields a discharge coefficient, C_d , of 0.90 and an orifice diameter of 1.57 millimeters. To calculate mass flow rate of each injector, the pressure drop across the tube must be determined. As a rule of thumb, the pressure drop must be a minimum of 20% of the combustion chamber pressure [3]. The pressure drop ratio required for the fuel and oxidizer to get a mixture ratio of 3.21 can be calculated using Equation 10. Holding the fuel pressure drop at 20% of the combustion chamber pressure, the oxidizer pressure drop is 50% of the combustion chamber pressure. Now the mass flow rate and velocity of each injector can be calculated using Equation 11 and 12. The angle of impingement that will cancel the tangential flow can be found using Equation 13. The oxidizer angle is 10 degrees, and the fuel angle is 43.63 degrees. With the flow rates of each injector known, the total number of injectors required to achieve the total mass flow rate can be calculated using Equation 14. The total number of doublet impinging sets to achieve the required flow rate and mixture ratio is 599. All injector quantities of interest can be found below in Table 3. A to-scale schematic of the injector design is shown in Appendix A.

With the combustor and injectors fully defined, the design of the diverging portion of the nozzle can be facilitated. The ambient pressure varies with altitude, which directly influences the optimal area ratio for an ideal nozzle. This area ratio is determined based on the altitude where the nozzle is expected to achieve its maximum performance, as represented by the thrust coefficient, C_f . Using Equations (16) and (17), the area ratio for a specific ambient pressure and engine chamber pressure can be calculated. For the second stage, the nozzle's area ratio is set to 100 to maximize performance at 10 km altitude, aligning with the ambient pressure at this altitude. Although the nozzle does not achieve peak efficiency at other altitudes, its performance remains relatively close to ideal within the first stage's operational range, shown in Figure 9. This design ensures that the second stage operates effectively across the altitudes it encounters, maintaining strong performance even when conditions vary from its optimal condition. Using this assumed exit-to-throat area ratio, a first pass of the performance metrics can be conducted. For this first pass, a very simple geometry of the nozzle, described in Equation 15, is used. Using the defined area ratios along the nozzle and the ratio of specific heat given by CEA, the Mach-Area relation, shown in Equation 16, can be used to solve for the Mach number anywhere along the nozzle. To ensure optimal performance, a bell-shaped nozzle geometry is selected due to its empirically proven advantages, including efficient expansion of exhaust gases, maximized thrust, and superior structural integrity. These characteristics make the bell-shaped nozzle ideal for optimizing rocket performance. The nozzle geometry (Figure 11) is designed based on the specified throat area and area ratio, utilizing parametric equations and quadratic Bézier curves. These curves determine the critical wall angles through the interpolation of empirical data, guided by the specified area ratios and throat radius [4,5].

The second-stage nozzle is designed to operate optimally at 10 km altitude. This means the nozzle operates in an under expanded condition for huge portions of its operational duration. Consequently, the pressure and temperature at any point along the nozzle can be calculated isentropically using Equations 17 and 18, respectively. The Mach number and speed of sound can be related to velocity using Equation 20. Now that the exit pressure and velocity of the nozzle is known, the thrust, thrust coefficient, and Isp can be found using Equations 21 – 23. The performance metrics of the second stage nozzle can be found below in Table 4.

To organize the engine configuration, another calculation was performed to determine thrust to weight ratio using equation 24. The result yields a thrust to weight ratio of 0.192; therefore, at least 6 engines will be required to achieve a thrust to weight ratio greater than 1. Due to the size of the rocket and second stage nozzle, the 6 engines were arranged in a star pattern which can be viewed in Table 1. Since the combustion chamber pressure is the maximum pressure experienced in a rocket engine, the nozzle thickness is equivalent to the chamber wall thickness for simplicity. Due to the low nozzle mass and relatively small size, there is no need to trim this nozzle below the minimum length determined.

Therefore, this nozzle is not considered lossy. The nozzle is drawn to-scale in and pictured in Appendix A below.

Using the propellant mass for the second stage obtained in the vehicle configuration section, the mass of the fuel and oxidizer tanks can be solved for. The mixture ratio can be used to determine how much of the propellant mass is dedicated to fuel and oxidizer. The mass of oxidizer and fuel can be determined using Equation 25. The shape of the fuel storage tanks was designed to be simple cylinders. Though spherical tanks offer better total volume and heat transfer properties, it was ultimately determined that the spherical tanks would be too wide to be practical for this mission. With the mass and density of both propellants known, the volume of the propellants can be calculated using Equation 26. The height of the oxidizer tank is assumed to be 10 meters and the height of the fuel tank is assumed to be 15 meters. The diameter of each tank can be calculated using Equation 27. For this stage, a turbopump feed system will be used to transport propellants from the storage tanks to the combustion chamber. This design allows for low propellant tank pressures, so the internal pressure of each tank can be assumed to be 0.3 MPa. For reference, the typical propellant temperature tank pressure in turbopump feed systems lies between 0.07 MPa and 0.34 MPa [3]. Because this setup allows for low storage pressure, it is not necessary to use rounded caps on the cylindrical storage tanks. Using internal pressure, the wall thickness can be estimated by rearranging the hoop stress formula shown in Equation 8. The material chosen for these tanks is aluminum 2219, which has a yield stress of 310 MPa. This material has a similar density to the wall density given in the mission statement and is non-reactive with the propellants, making it a great option. This yields a fuel tank wall thickness of 2.7 millimeters and an oxidizer tank thickness of 2.4 millimeters. From this, the mass of each empty storage tank can be calculated using Equation 28. The empty fuel tank and oxidizer tank have a mass of approximately 2283 kg and 886 kg respectively. All fuel and oxidizer tank properties can be found below in Table 5.

Performing a quick check, the combined mass of the nozzle, empty oxidizer and empty fuel tank is roughly 6575 kg. The structural mass in section 2 was calculated to be approximately 13700 kg. This indicates that there are 7150 kg to be dedicated to the structural mass of the rocket body, avionics, insulation, and piping necessary. The material used for the structure rocket body was again chosen to be aluminum 2219 because of its similar density to the one in the specified mission statement. The body thickness was designed to withstand maximum aerodynamic loading, also known as "max Q", without deformation. During max Q, the rocket can expect to see about 0.03 MPa of aerodynamic loading. This load is far less than the yield strength of the aluminum alloy chosen. For this reason, the wall thickness was calculated to minimize weight. The wall thickness of the second stage was calculated using Equation 8. The to-scale drawings of the propellant tanks and rocket structure can be shown in Appendix A. All structural masses are also reported below in Table 1.

5. Rocket Design for Upper Stage

Moving on to the upper stage, assume the propellants used are Liquid Oxygen (LOX) and Liquid Hydrogen (LH₂). This propellant combination is renowned for its high specific impulse and capability to create large ΔV , which is essential for this lunar mission. Though LOX/LH₂ systems generally have a larger structural coefficient, thrust-to-weight ratio is less of a concern in a vacuum, which is where the spacecraft will be operating. It is also assumed that the combustion chamber pressure, p_0 , is 30 MPa. In modern engines, such as SpaceX's raptor vacuum engine, the chamber pressure is approximately 30 MPa, which makes this assumption valid [1].

Using the assumed fuels and combustion chamber pressure, the other conditions inside of the combustor can be described using NASA's Combustion Equilibrium Analysis (CEA) tool as done before in previous sections. CEA requires the initial temperatures of the propellants, so 20- and 90-degrees Kelvin were chosen for LH₂ and LOX respectively. These values were determined by analyzing the boiling point of both propellants [2]. The final input that CEA requires is an oxidizer-to-fuel ratio, or mixture ratio, to calculate chamber conditions. Figure 12 depicts the specific impulse plotted as a function of

mixture ratio. The performance metrics for this figure were calculated at vacuum conditions using the chamber pressure defined above. To maximize performance metrics in a vacuum, a mixture ratio of 6.25 will be used. With the mixture ratio, chamber pressure, and propellant temperature known, the CEA results for adiabatic flame temperature, ratio of specific heats, and molecular weight of the products for the upper stage are tabulated in Table 2 below.

From the selected propellant combination, Figure 6 can be used to determine the characteristic length of the combustion chamber. The LOX/LH₂ combination gives a heritage characteristic length, L^* , range of 76 – 102 centimeters. As an initial assumption, L^* of 80 centimeters will be used. The combustion chamber can continue to be described by selecting the chamber-to-throat area ratio. Figure 7 plots chamber-to-throat area ratio, $\frac{A_c}{A_t}$, as a function of chamber-to-throat pressure ratio, $\frac{p_0}{p_t}$, and chamber mass, m_c . The pressure losses across the combustion chamber occur from Rayleigh heat addition. This pressure drop hinders performance, which drives the decision to a large $\frac{A_c}{A_t}$. However, at large $\frac{A_c}{A_t}$, m_c is nearly doubled, which drives the decision to a smaller $\frac{A_c}{A_t}$. At a chamber-to-throat area ratio of 2.5, Rayleigh losses are minimized while having a respectable m_c . With both $\frac{A_c}{A_t}$ and L^* known, Equation 7 can be used to calculate the length of the combustion chamber, L_c . Chamber thickness was derived by utilizing the hoop stress equation shown in Equation 8. From research on other rocket engines the selected wall material is 70/30 tungsten copper with a yield strength of approximately 500 MPa. Given this yield strength and the pressure of the chamber, the calculated minimum wall thickness is 5mm.

Assuming the throat area of nozzle to be 0.01 square meters, the mass flow rate through the nozzle can be calculated using Equation 9. This mass flow rate is very important for performance metric calculations and determining the injector design. For this system, a doublet impinging stream pattern will be used. Specifically, unlike doublet patterns will be used to promote mixing between fuel and oxidizer. In this setup, it is imperative that the combined streams cancel the tangential momentum to prevent backsplash onto the injector. To increase the discharge coefficient, a short tube with a rounded entrance will be used for the orifice type. From Figure 8, this yields a discharge coefficient, C_d , of 0.90 and an orifice diameter of 1.57 millimeters. To calculate mass flow rate of each injector, the pressure drop across the tube must be determined. As a rule of thumb, the pressure drop must be a minimum of 20% of the combustion chamber pressure [3]. The pressure drop ratio required for the fuel and oxidizer to get a mixture ratio of 6.25 can be calculated using Equation 10. Holding the fuel pressure drop at 20% of the combustion chamber pressure, the oxidizer pressure drop is 50% of the combustion chamber pressure. Now the mass flow rate and velocity of each injector can be calculated using Equation 11 and 12. The angle of impingement that will cancel the tangential flow can be found using Equation 13. The oxidizer angle is 56 degrees, and the fuel angle is 20 degrees. With the flow rates of each injector known, the total number of injectors required to achieve the total mass flow rate can be calculated using Equation 14. The total number of doublet impinging sets to achieve the required flow rate and mixture ratio is 356. All injector quantities of interest can be found below in Table 3. A to-scale schematic of the injector design is shown in Appendix A.

With the combustor and injectors fully defined, the design of the diverging portion of the nozzle can be facilitated. It is known that in a vacuum, the ideal nozzle is infinite length. However, due to size and weight constraints, this is not a practical nozzle design. By plotting specific impulse as a function of exit-to-throat area, shown in Figure 13, the specific impulse levels off asymptotically around $\frac{A_e}{A_t}$ of 40. While this nozzle will not technically yield the perfectly expanded results, it is close enough to providing the ideal results while maintaining practicality. Using this assumed exit-to-throat area ratio, a first pass of the performance metrics can be conducted. For this first pass, a very simple geometry of the nozzle, described in Equation 15, is used. Using the defined area ratios along the nozzle and the ratio of specific heat given by CEA, the Mach-Area relation, shown in Equation 16, can be used to solve for the Mach number anywhere along the nozzle. The exit Mach number is of particular interest and can be used as

an accurate input to the 2-D method of characteristics code, found in the Appendix, for a minimum length nozzle. This code takes the assumed throat area and initial Mach number guess that was made under the assumption of quasi-one-dimensional flow with a conical nozzle and generates a minimum length bell shaped nozzle. This minimum length is determined by cancelation of characteristic waves within the nozzle, creating uniform flow at the exit of the nozzle. The minimum length of the nozzle is 1300 millimeters and has an exit-to-throat area of 22, shown in Figure 14. The 2-D method of characteristics code outputs the wall coordinates of the diverging portion of the nozzle, which can be used to generate an area function for the optimal nozzle design.

Armed with this new geometry and exit-to-throat area given by the 2-D method of characteristics code, the optimized performance metrics can be obtained. The upper engine operates in a vacuum, which indicates that the nozzle is under expanded. Hence, the pressure and temperature anywhere along the nozzle can be found isentropically using Equations 17 and 18 respectively. With known temperatures along the nozzle, the speed of sound at any point can be calculated using Equation 19. The Mach number and speed of sound can be related to velocity using Equation 20. Now that the exit pressure and velocity of the nozzle is known, the thrust, thrust coefficient, and Isp can be found using Equations 21 – 23. The performance metrics of the upper stage nozzle can be found below in Table 4. This specific impulse calculated here was used above to calculate the wet and dry mass of the upper stage.

The trust-to-weight ratio, found in Equation 24, would typically be used to determine the number of engines required to successfully propel the upper stage. However, because the upper stage only operates in a vacuum, the structure is weightless, so only one engine is required. For symmetry and easy gimbling, the engine will be configured in the center.

Since the combustion chamber pressure is the maximum pressure experienced in a rocket engine, the nozzle thickness is equivalent to the chamber wall thickness for simplicity. Due to the low nozzle mass and relatively small size, there is no need to trim this nozzle below the minimum length determined. Therefore, this nozzle is not considered lossy. The nozzle is drawn to-scale in and pictured in Appendix A below.

Using the propellant mass for the upper stage obtained in section 2, the mass of the fuel and oxidizer tanks can be solved for. The mixture ratio can be used to determine how much of the propellant mass is dedicated to fuel and oxidizer. The mass of oxidizer and fuel can be determined using Equation 25. The shape of the fuel storage tanks was designed to be simple cylinders. Though spherical tanks offer better total volume and heat transfer properties, it was ultimately determined that the spherical tanks would be too wide to be practical for this mission. With the mass and density of both propellants known, the volume of the propellants can be calculated using Equation 26. The height of the oxidizer tank is assumed to be 20 meters and the height of the fuel tank is assumed to be 15 meters. The diameter of each tank can be calculated using Equation 27. For this stage, a turbopump feed system will be used to transport propellants from the storage tanks to the combustion chamber. This design allows for low propellant tank pressures, so the internal pressure of each tank can be assumed to be 0.3 MPa. For reference, the typical propellant temperature tank pressure in turbopump feed systems lies between 0.07 MPa and 0.34 MPa [3]. Because this setup allows for low storage pressure, it is not necessary to use rounded caps on the cylindrical storage tanks. Using internal pressure, the wall thickness can be estimated by rearranging the hoop stress formula shown in Equation 8. The material chosen for these tanks is aluminum 2219, which has a yield stress of 310 MPa. This material has a similar density to the wall density given in the mission statement and is non-reactive with the propellants, making it a great option. This yields a fuel tank wall thickness of 2 millimeters and an oxidizer tank thickness of 1.1 millimeters. From this, the mass of each empty storage tank can be calculated using Equation 28. The empty fuel tank and oxidizer tank have a mass of approximately 2283 kg and 886 kg respectively. All fuel and oxidizer tank properties can be found below in Table 5.

Performing a quick check, the combined mass of the nozzle, empty oxidizer and empty fuel tank is roughly 3300 kg. The structural mass in the launch vehicle configuration section was calculated to be approximately 7300 kg. This indicates that there are 4000 kg to be dedicated to the structural mass of the rocket body, avionics, insulation, and piping necessary. The rocket body was made large enough to enclose both propellant tanks and support the payload, i.e. cargo bay and actual payload. The material used for the structure was again chosen to be aluminum 2219 because of its similar density to the one in the specified mission statement. The body thickness was designed to withstand maximum aerodynamic loading, also known as "max Q", without deformation. During max Q, the rocket can expect to see about 0.03 MPa of aerodynamic loading. This load is far less than the yield strength of the aluminum alloy chosen. For this reason, the wall thickness was calculated to minimize weight. Hence the wall thickness was estimated to be 8 millimeters. For the scope of this analysis, the effects of the weight above the point of analysis were ignored and it was assumed there would be no buckling. Similarly to the propellant tanks, the mass of the rocket body was calculated to be 1150 kg. This leaves roughly 2200 kg for avionics, insulation, and piping. This extra wiggle room in the structural mass can be used for excess propellant as a safety factor. Because both propellants are cryogenic liquids, it is possible that they will partially vaporize, causing a loss of useable fuel. The to-scale drawings of the propellant tanks and rocket structure can be shown in Appendix A. All structural masses are also reported below in Table 1.

This concludes the analysis of the upper stage, which served as a critical step in enabling a successful evaluation of the lower stages. By completing this stage of the analysis, a solid foundation was established for understanding and addressing the performance requirements of the subsequent stages.

6. Overall Arrangement of Rocket

With all stages of the rocket now fully designed, the overall arrangement of the rocket is as follows. The complete rocket stands 97.84 meters tall and the maximum diameter, occurring on the first stage, is 6.68 meters. These overall dimensions appear to be nominal with other rockets that have performed successful lunar missions. For example, the Saturn V rocket stands at 111 meters tall and 11 meters wide [6]. As discussed in previous sections, the rocket body needed to enclose the propellant tanks and withstand the maximum aerodynamic load. To further specify, the wall thickness of the rocket body was designed to withstand maximum loading, and the height and diameter of each stage were designed to encompass the propellant tanks. It was ensured that the rocket body mass of each stage was less than the total structural mass for each stage. Though it was not discussed heavily in this analysis, there are other structural weight considerations such as piping, avionics, extra fuel, etc. The remaining structural mass allocated for each stage can be dedicated to the aforementioned considerations. Using these constraints, the height and diameter of the rocket was defined loosely, giving some leeway to the design of components outside the scope of the analysis.

The total mass of the rocket, including payload, structural, and propellant mass, came out to be 1,227,942 kilograms. The mass of each stage was calculated above in previous sections. This weighs about half as much as Saturn V, which is reported to have a mass of 2.8-2.9 million kilograms. This mass difference can be explained by the difference in diameter and payload between the designed rocket and Saturn V. The overall schematic of the rocket can be found below in Appendix A.

7. Conclusions and Discussion

The rocket design was developed using key assumptions that impose limitations on its feasibility and performance. Simplifications such as frozen combustion modeling and constant chamber pressure provided good initial judgement on the conditions inside of the nozzle but may not be an accurate approximation. For this analysis, the chamber pressure and throat area were assumed from the beginning, which drove the entire design process. Varying these quantities greatly affects the performance of the rocket. This indicates that more time and research should be invested in increasing

combustion chamber pressure for future designs. The research and design scope was limited by fixing the chamber pressure throughout all engines.

This design faced limitations in the nozzle analysis, particularly during the transition from quasi-1D flow to 2D flow assumptions. Quasi-1D flow was initially used to estimate the required exit Mach number for each engine, providing a starting point for the nozzle design. However, significant challenges emerged when attempting to determine a minimum-length nozzle using the 2D method of characteristics. These complications highlighted the need for further refinement in the nozzle design process. Further, the 2D flow assumption is a good starting point, but a conical nozzle analysis should be done for future designs. Another future design task would involve optimizing the nozzle geometry for all altitudes. For example, in this analysis, the first stage engine nozzle was developed to perform best at sea-level as that is when thrust is most important for that stage. However, optimal design is only for a specific altitude. In future designs, it is worth considering a variable exit area geometry.

Other areas that were not discussed in this analysis are as follows. Thermal management was not fully addressed; including thermal systems and consideration is critical for all rocket analysis. Typically, part of rocket design includes body insulation from aerodynamic heating, and heat rejection from the nozzle. Removing the thermal system from our design allows our overall mass to seem lower and may even affect the nozzle geometry we selected since our design assumes some arbitrary method of cooling. To add, complex flow features inside the nozzle such as boundary effects were neglected. It is also important to note that the ΔV required for this mission was based on the impulsive maneuver assumption, which simplified thrust modeling but ignored burn durations. Despite these limitations, the design establishes a solid foundation for further improvement and cautions the use of assumptions.

8. Description of Contributions

Overall, the effort contributed to this analysis and discussion was spread equally between group members. The breakdown of contributions can be found below in Table 6. In this table, the contact information and contribution of each group member can be found for each element of the project. In brief, all group members worked diligently in conceptualizing the problem at hand. The strengths of each group member were utilized to carry out the rest of the analysis approach and numerical computations. Jake Lancaster was primarily responsible for designing the combustion chamber for each engine, all CAD drawings. Sukhyeon Lim was primarily responsible for characterizing the diverging portion of the nozzle for each engine. Samuel Rockaway was primarily responsible for designing the injection system, fuel tanks, and all relevant masses of each stage. Though members had primary responsibilities, this design was an iterative process that required an immense amount of collaboration, requiring group members to become experts on non-primary sections. As for the report, figures, and equations, all members split the writing equally. Samuel Rockaway wrote the introduction, launch configuration, upper stage, and contributions sections. Sukhyeon Lim wrote the first stage design and conclusion sections. Jake Lancaster wrote the second stage design and overall arrangement sections. It should be emphasized again that all group members contributed equally to the overall product of this assignment.

9. Equations, figures and tables

9.1 Equations

Equation 1: The mass ratio in terms of Isp, ΔV , and gravitational constant, $g_0 = 9.81 \frac{m}{s^2}$, can be written as:

$$\frac{m_0}{m_b} = R = e^{(\frac{\Delta V}{Isp * g_0})}$$

Equation 2: The payload ratio can be calculated using the mass ratio R and structural coefficient using:

$$\lambda = \frac{(\epsilon * R) - 1}{1 - R}$$

Equation 3: The wet mass as a function of payload ratio and payload mass can be written as:

$$m_0 = \frac{m_{pl} + (\lambda * m_{pl})}{\lambda}$$

Equation 4: The dry mass as a function of R and m_0 can be written as:

$$m_b = \frac{m_0}{R}$$

Equation 5: The structural mass as a function of m_b and m_{pl} can be written as:

$$m_s = m_b - m_{pl}$$

Equation 6: The propellant mass as a function m_0 , m_s , and m_{pl} can be written as:

$$m_p = m_0 - m_s - m_{pl}$$

Equation 7: The characteristic length of the combustion chamber can be written as:

$$L^* = \frac{A_c}{A_t} L_c$$

Equation 8: Wall thickness as a function of yield strength S, Diameter D, and Pressure P:

$$t_w = \frac{P * D}{2 * S}$$

Equation 9: The choked mass flow rate can be written as:

$$\dot{m} = \frac{p_0 A_t}{\sqrt{T_0}} \sqrt{\frac{\gamma}{R} \left(\frac{2}{\gamma + 1} \right)^{\frac{\gamma+1}{\gamma-1}}}$$

Equation 10: The pressure drop ratio between fuel and oxidizer across the injector inlet as a function of density and mixture ratio can be written as:

$$\frac{\Delta p_{fuel}}{\Delta p_{oxidizer}} = \frac{\rho_{oxidizer}}{\rho_{fuel}} * \frac{1}{\phi^2}$$

Equation 11: The mass flow rate of a single injector as a function of C_d , Δp , A, and ρ can be written as:

$$\dot{m}_i = C_d * A * \sqrt{2 * \rho * \Delta p}$$

Equation 12: The velocity of a single injector as a function of C_d , Δp , and ρ can be written as:

$$v_i = C_d * \sqrt{2 * \Delta p / \rho}$$

Equation 13: The oxidizer and fuel injector angles to eliminate any tangential flow can be written as:

$$\dot{m}_1 * v_1 * \sin a_1 = \dot{m}_2 * v_2 * \sin a_2$$

Equation 14: The total number of injectors can be written as:

$$\# \text{ of injectors} = \frac{\dot{m}_{total}}{\dot{m}_i}$$

Equation 15: The simple nozzle geometry for converging and diverging sections can be written as:

$$\frac{A(x)}{A_t} = \begin{cases} 1 - \left(\frac{A_e}{A_t} - 1\right) \frac{x}{L_c} & -1 < \frac{x}{L_c} < 0 \\ 1 + \left(\frac{A_e}{A_t} - 1\right) \frac{x}{L_D} & 0 < \frac{x}{L_D} < 1 \end{cases}$$

Equation 16: The Mach area relation as a function of Mach number and gamma only can be written as:

$$\frac{A}{A^*} = \sqrt{\frac{1}{M^2} \left[\frac{2}{\gamma + 1} \left(1 + \frac{\gamma - 1}{2} M^2 \right) \right]^{\frac{\gamma+1}{\gamma-1}}}$$

Equation 17: The isentropic pressure ratio as a function of Mach number and gamma only can be written as:

$$\frac{p(x)}{p_0} = \left(1 + \frac{\gamma + 1}{2} M^2 \right)^{\frac{-\gamma}{\gamma-1}}$$

Equation 18: The isentropic temperature ratio as a function of Mach number and gamma only can be written as:

$$\frac{T(x)}{T_0} = \frac{1}{1 + \frac{\gamma - 1}{2} M^2}$$

Equation 19: The speed of sound at any location in the nozzle can be written as:

$$a = \sqrt{\gamma RT}$$

Equation 20: The relationship between Mach number, speed of sound, and velocity can be written as:

$$M = \frac{u}{a}$$

Equation 21: The thrust equation can be written as:

$$F_t = \dot{m}v_e + (p_e - p_a)A_e$$

Equation 22: The specific impulse equation can be written as:

$$Isp = \frac{F_t}{\dot{m}g_0}$$

Equation 23: The thrust coefficient equation can be written as:

$$C_f = \frac{F_t}{P_c A_t}$$

Equation 24: The thrust-to-weight equation can be written as:

$$TW = \frac{F_t}{mass * g}$$

Equation 25: The mass of fuel and oxidizer as a function of the total propellant mass and the mixture ratio can be written as:

$$m_{fuel} = \frac{m_p}{\emptyset + 1}$$

$$m_{oxidizer} = m_p - m_{fuel}$$

Equation 26: The volume of each propellant can be written as:

$$V = \frac{m}{\rho}$$

Equation 27: The inner diameter of a propellant tank can be written as:

$$d = 2 * \sqrt{\frac{V}{\pi * h}}$$

Equation 28: The mass of an empty propellant tank can be written as:

$$m = \left[\frac{\pi}{4} ((2t + d)^2 h) - V \right] \rho$$

9.2 Figures

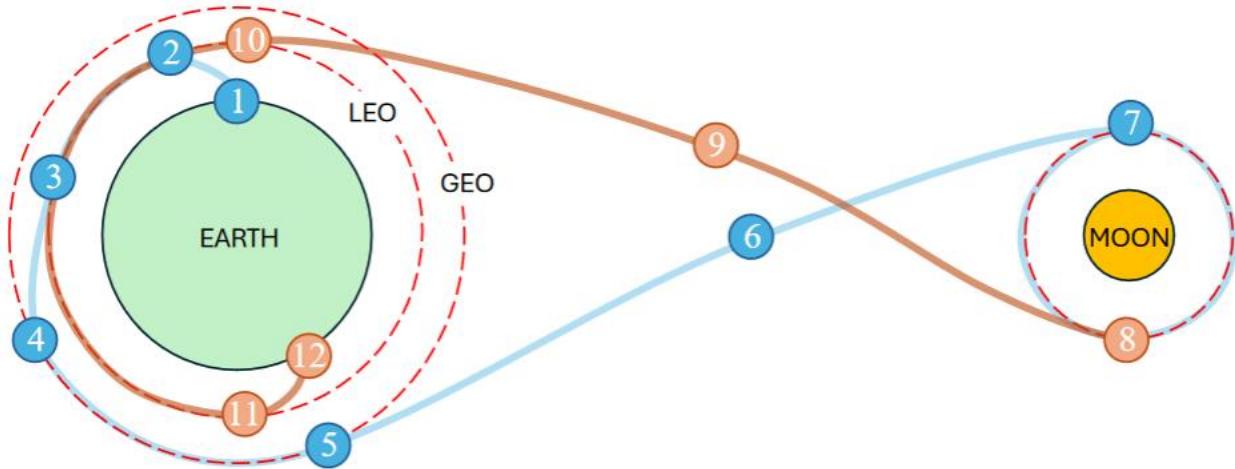


Figure 1. Mission Profile from Earth to Moon and back. Note that for this mission, maneuvers 1-10 will be performed.

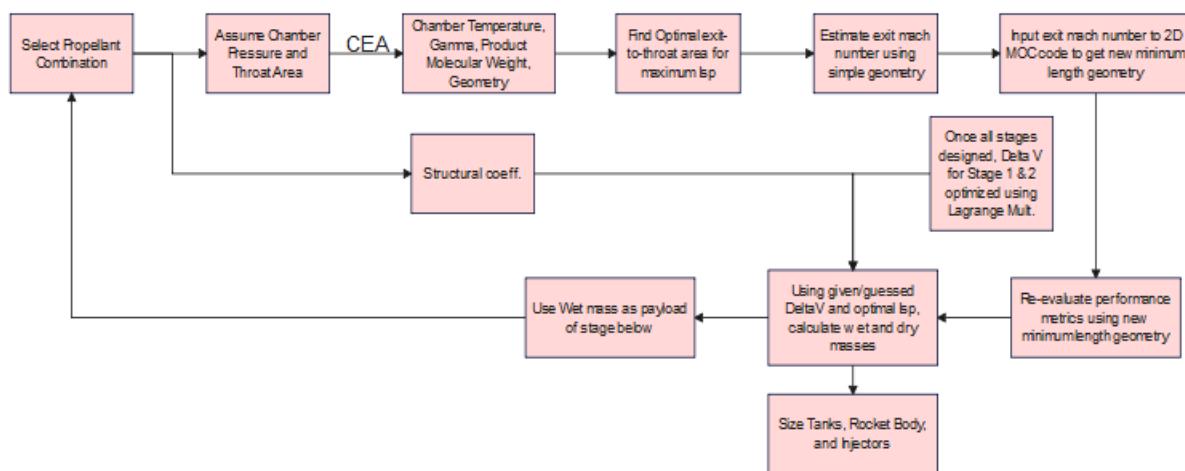


Figure 2. Overall design process to optimize a multi-stage vehicle. Follows a top-down method, starting with the upper stage and working downward. Notice how each stage is designed to support the payload mass of the stages above it.

Maneuver ID	Description	ΔV , km/s
1	Launch to LEO from Earth	8.0
2	Orbit circularization at 185 km	0.2
3	Transfer into geostationary transfer orbit (GTO)	2.46
4	Orbit circularization in GEO	1.48
5	Earth escape maneuver from GTO apogee	0.68
6	Midcourse correction	0.14
7	Lunar orbit insertion (100 km)	0.68
8	Moon escape maneuver	0.68
9	Midcourse correction	0.14
10	Transfer to LEO (185 km, assumes no aerobreaking)	3.14

Figure 3. Maneuver ID table, giving required change in velocity for each maneuver.

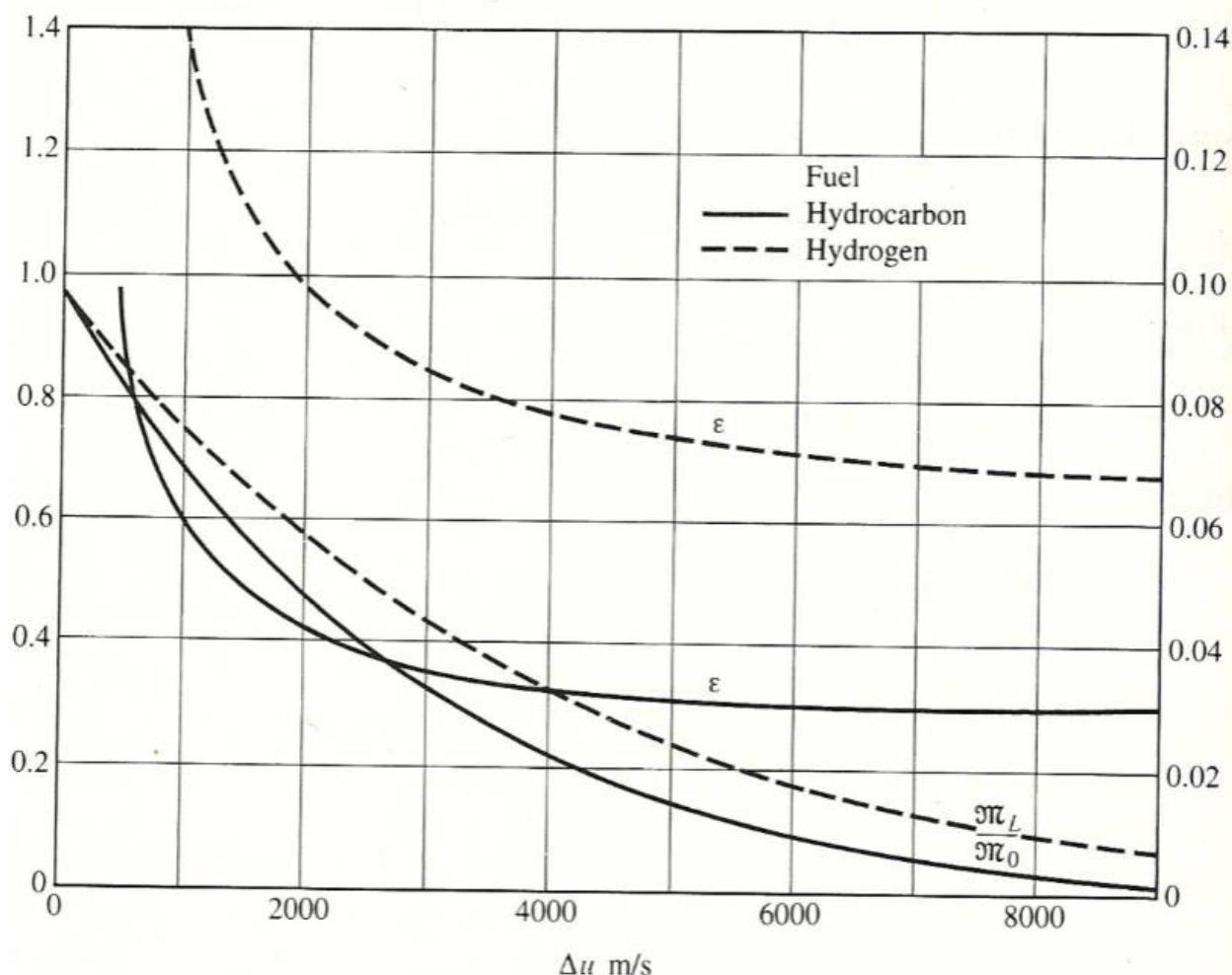
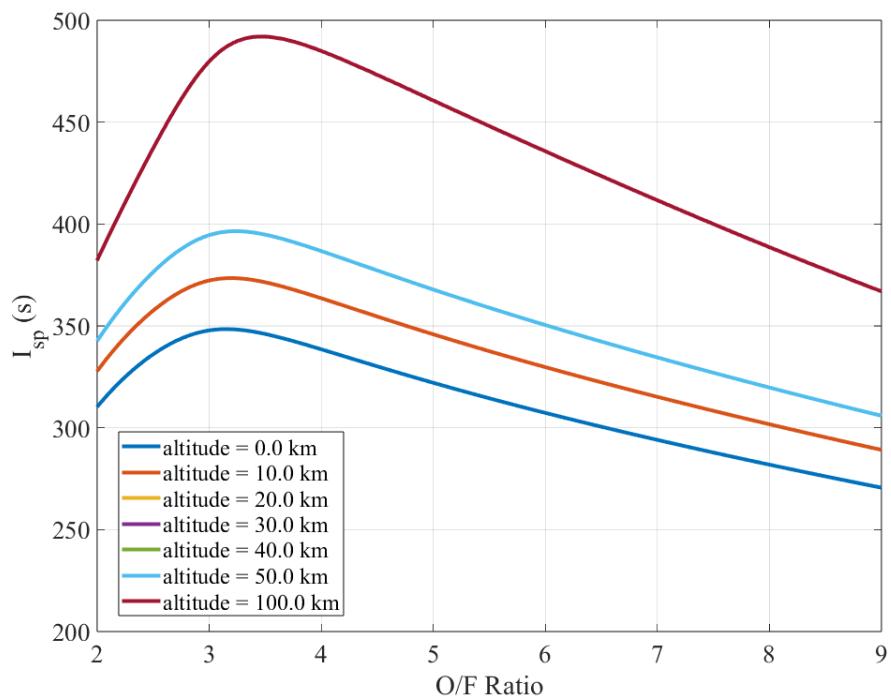


Figure 4. Typical values of structural coefficient as a function of velocity change and fuel combination.

Figure 5. LOX/LCH₄ I_{sp} - O/F ratio plot for various altitudes.**Table 1:** Chamber Characteristic Length, L*

Propellant Combination	L*, cm
Nitric acid/hydrazine-base fuel	76-89
Nitrogen tetroxide/hydrazine-base fuel	76-89
Hydrogen peroxide/RP-1 (including catalyst bed)	152-178
Liquid oxygen/RP-1	102-127
Liquid oxygen/ammonia	76-102
Liquid oxygen/liquid hydrogen (GH ₂ injection)	56-71
Liquid oxygen/liquid hydrogen (LH ₂ injection)	76-102
Liquid fluorine/liquid hydrogen (GH ₂ injection)	56-66
Liquid fluorine/liquid hydrogen (LH ₂ injection)	64-76
Liquid fluorine/hydrazine	61-71
Chlorine trifluoride/hydrazine-base fuel	51-89

Figure 5. Chamber characteristic length for various propellants combinations.

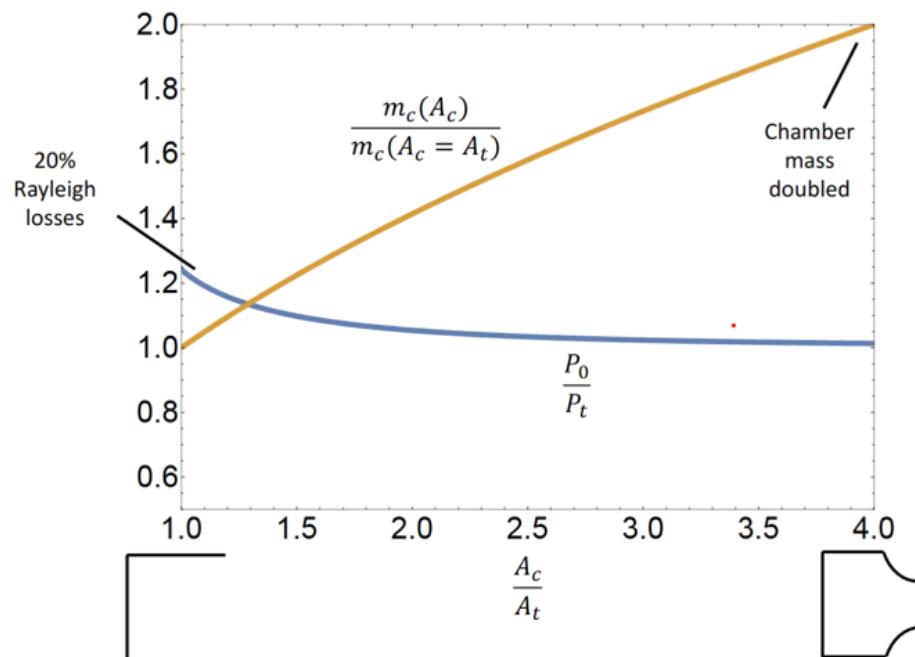


Figure 7. Rayleigh loss and chamber mass for chamber to throat area ratio.

Orifice Type	Diagram	Diameter (mm)	Discharge Coefficient
Sharp-edged orifice		Above 2.5 Below 2.5	0.61 0.65 approx.
Short-tube with rounded entrance $L/D > 3.0$		1.00 1.57 1.00 (with $L/D \sim 1.0$)	0.88 0.90 0.70
Short tube with conical entrance		0.50 1.00 1.57 2.54 3.18	0.7 0.82 0.76 0.84–0.80 0.84–0.78
Short tube with spiral effect		1.0–6.4	0.2–0.55
Sharp-edged cone		1.00 1.57	0.70–0.69 0.72

Figure 7. Injector types and characteristics.

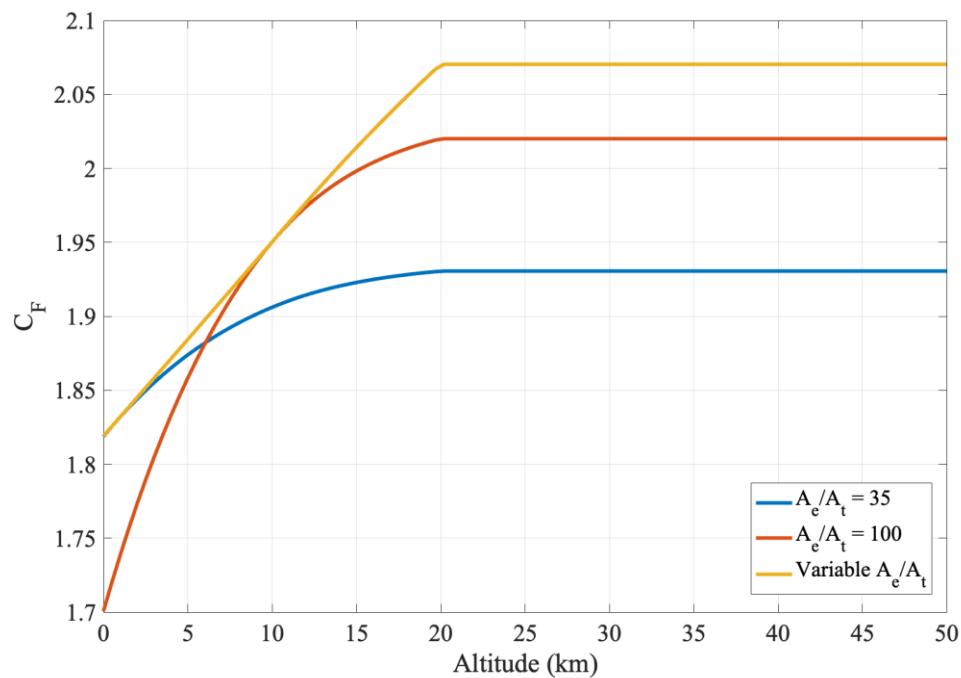


Figure 8. Thrust coefficient with different area ratio

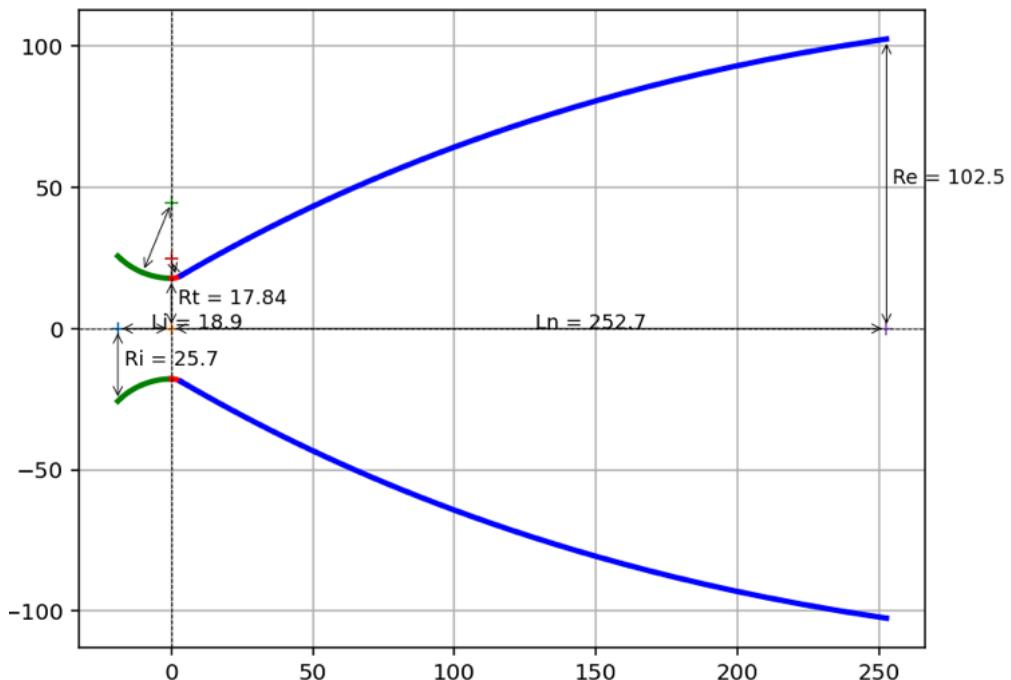


Figure 10. Bell-shaped nozzle contour for the first stage

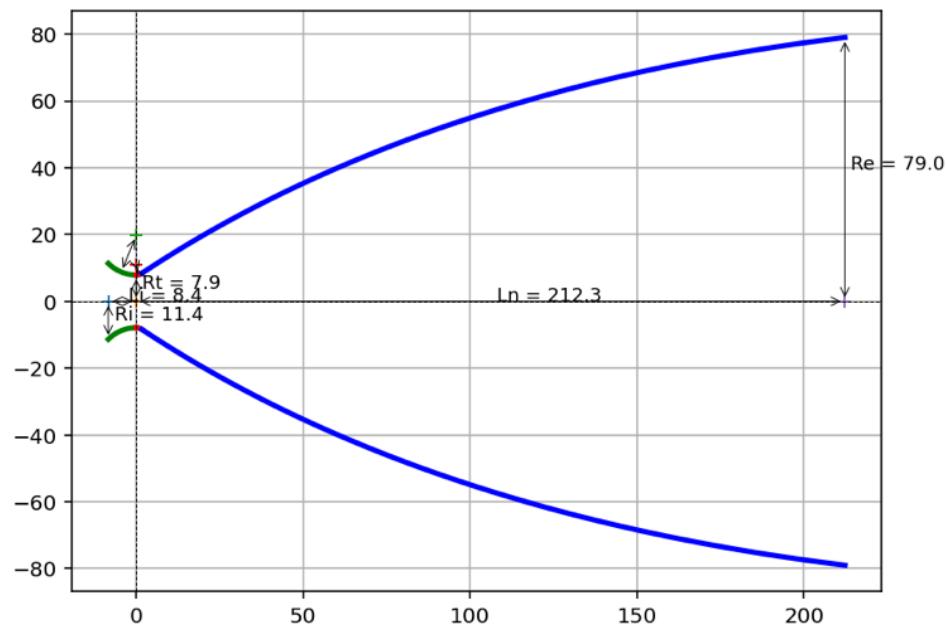
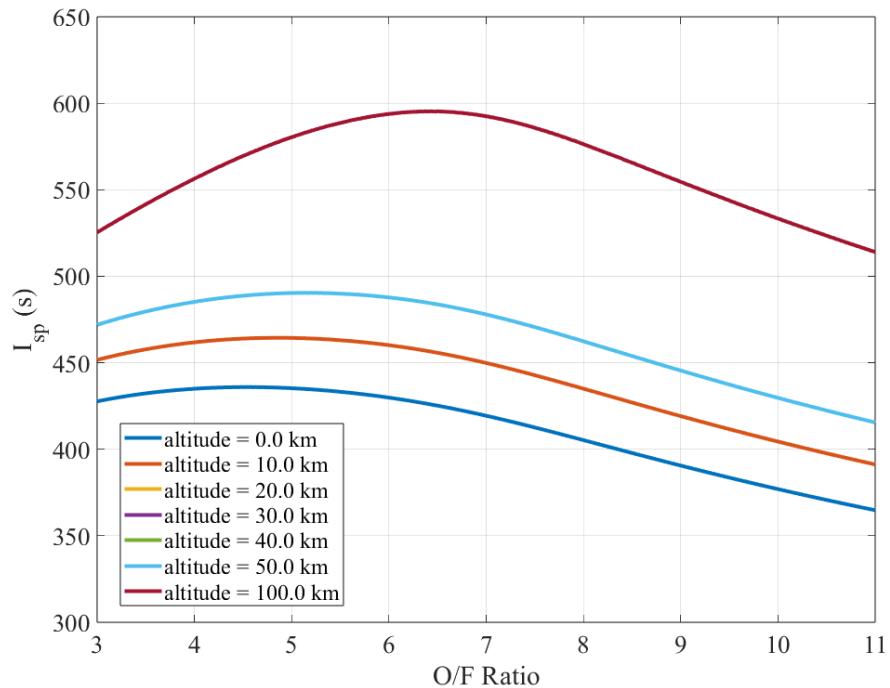


Figure 11. Bell-shaped nozzle contour for the first stage

Figure 12. LOX/LH₂ I_{sp} - O/F ratio plot for various altitudes.

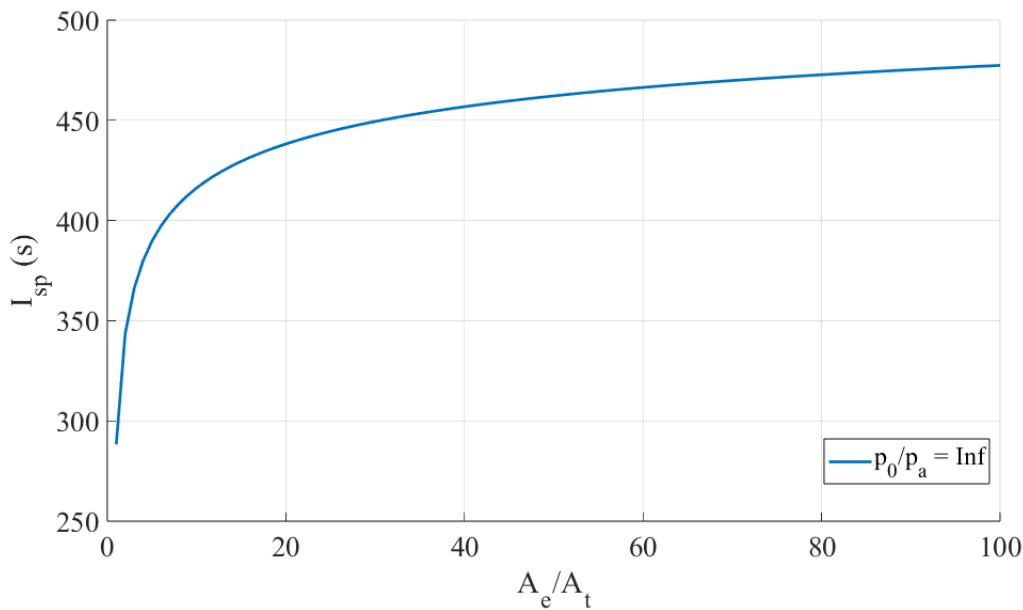


Figure 9. Specific impulse variation with area-ratio change.

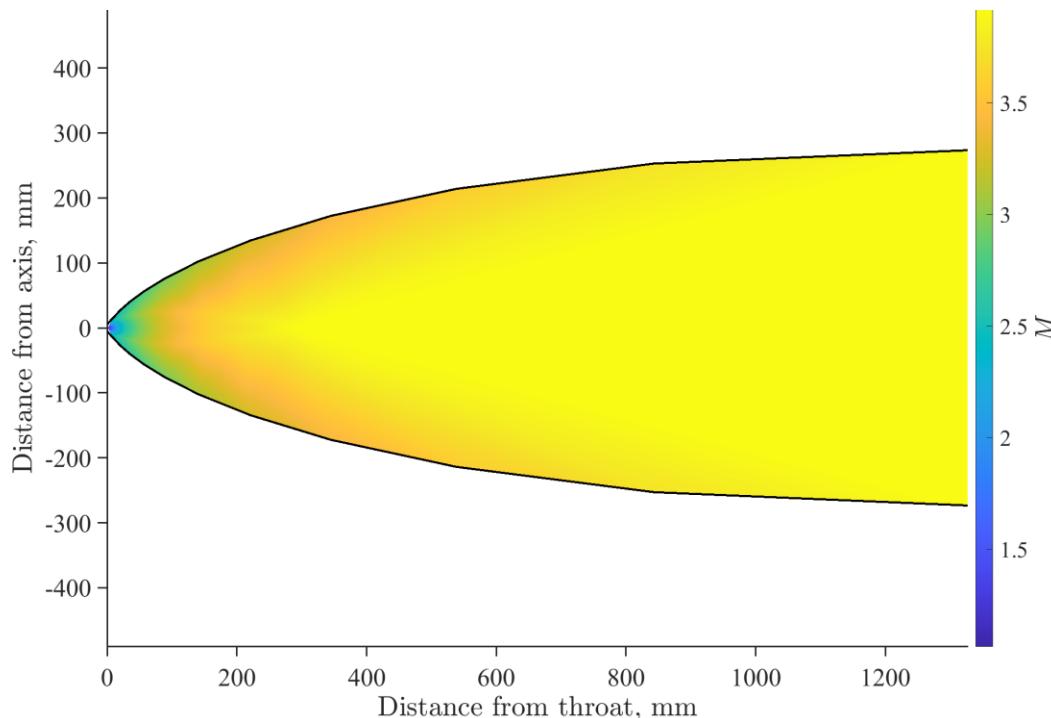


Figure 10. Mach number distribution in the upper stage nozzle, calculated using 2D MOC.

9.3 Tables

Table 1: Tabulation of masses, engine specifications, number of engines, and engine configuration for each stage.

	Stage 1	Stage 2	Upper Stage
Fuel Type	LOX/CH4	LOX/CH4	LOX/LH2
Structural Coeff.	0.0380	0.0300	0.0625

Dry Mass (kg)	590,393.95	126,942.22	12,304.75
Wet Mass (kg)	1,227,941.68	570,675.98	113,218.50
Structural Mass (kg)	19,717.97	13,723.72	7,304.75
Propellant Mass (kg)	63,7547.73	443,733.75	100,913.75
Isp (s)	348	373	440.94
Expansion Ratio	35	15	22
Total Thrust (kN)	5,486	1098	586
Total Mass Flow Rate ($\frac{kg}{s}$)	1,597.5	319	130.8
# of Engines	3	6	1

Engine Configuration

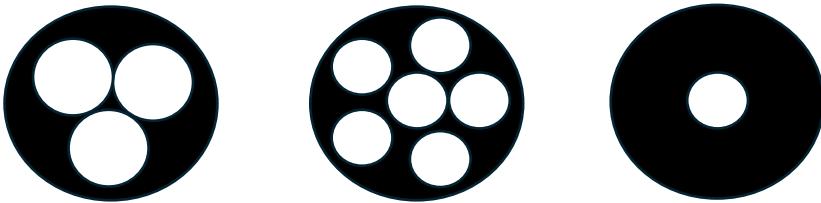


Table 2: NASA CEA Outputs for the combustor of each stage.

	Stage 1	Stage 2	Upper Stage
Chamber Pressure p_0 [MPa]	30	30	30
Gamma γ	1.1333	1.1333	1.363
Adiabatic Flame Temperature T_0 [K]	3761.26	3761.26	3793.6
Product Molecular Weight [kg/mol]	0.023562	0.023562	0.016231

Table 3: Injector design for each stage.

	Stage 1	Stage 2	Upper Stage
Injector Type	Unlike Doublet Impinging	Unlike Doublet Impinging	Unlike Doublet Impinging
Orifice Type	Short Tube Rounded Entrance	Short Tube Rounded Entrance	Short Tube Rounded Entrance
Discharge Coeff. C_d fuel & oxidizer	0.90	0.90	0.90
$D_{fuel \& oxidizer}$ [mm]	1.57	1.57	1.57
\dot{m}_i fuel [$\frac{kg}{s}$]	0.13	0.13	0.05
\dot{m}_i oxidizer [$\frac{kg}{s}$]	0.41	0.41	0.3175
v_i fuel [$\frac{m}{s}$]	148.63	148.63	370.39

$v_i \text{ oxidizer } \left[\frac{m}{s} \right]$	183.98	183.98	143.75
$\Delta p_{\text{fuel}} \text{ [MPa]}$	6	6	6
$\Delta p_{\text{oxidizer}} \text{ [MPa]}$	23.84	23.84	14.55
$\alpha_1 \text{ fuel [deg]}$	43.63	43.63	56.06
$\alpha_2 \text{ oxidizer [deg]}$	10.00	10.00	20
# of Doublet Injector Sets	2,998	599	356

Table 4: Nozzle and chamber design and performance for each stage.

	Stage 1 Engine	Stage 2 Engine	Upper Stage Engine
Throat Area $A_t \text{ [m}^2]$	0.1	0.02	0.01
Chamber Ratio $\frac{A_c}{A_t}$	3	3	2.5
Expansion Ratio $\frac{A_e}{A_t}$	35	100	22
Mass Flow Rate \dot{m} $\left[\frac{kg}{s} \right]$	1597.5	319.5	130.8
Length [m]	2.61	2.12	1.3
Isp Vacuum [s]	370.79	390.20	422.40
Thrust Vacuum [kN]	5811	1223	567
Coefficient of Thrust Vacuum	1.94	2.04	1.89
Isp Sea Level [s]	348.16	325.55	405.80
Thrust Sea-Level [kN]	5456	1020	545
Coefficient of Thrust	1.82	1.70	1.82
Material	ASTM B702 WCu	ASTM B702 WCu	ASTM B702 WCu
Chamber Thickness [mm]	8	8	5
Nozzle Thickness [mm]	8	8	5
Nozzle Mass [kg]	654	315	75.5

Table 5: Fuel and Oxidizer tank design for each stage.

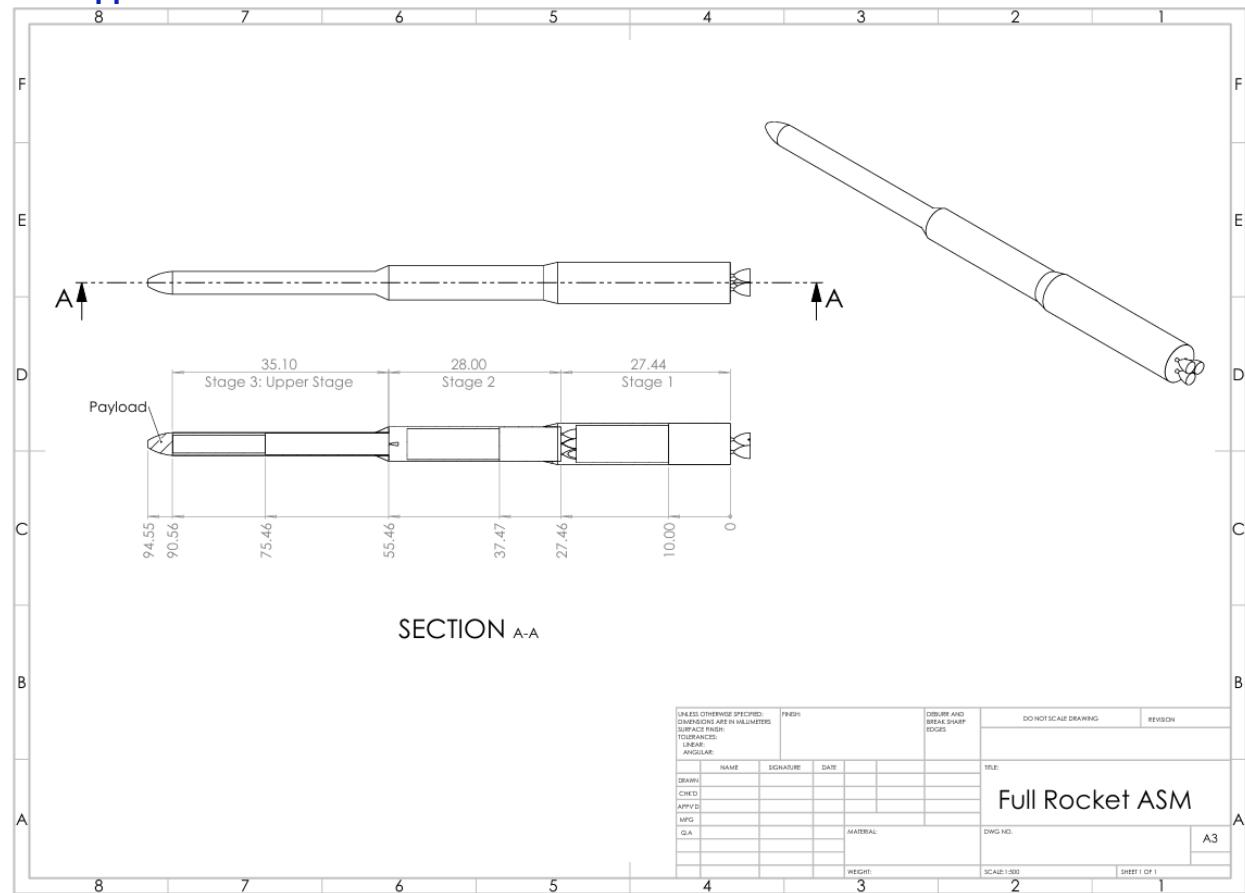
	Stage 1	Stage 2	Upper Stage
Tank Shape	Cylinder	Cylinder	Cylinder
Feed Type	Turbopump	Turbopump	Turbopump
Tank Pressure [MPa]	0.3	0.3	0.3
Fuel Tank OD [m]	6.66	5.53	4.09
Oxidizer Tank OD [m]	6.05	5.02	2.21
Fuel Tank Height [m]	10	10	15
Oxidizer Tank Height [m]	15	15	20
Fuel Tank Thickness [mm]	3.2	2.7	2.0
Oxidizer Tank Thickness [mm]	2.9	2.4	1.1
Fuel Tank Mass (Empty) [kg]	4,037.46	2,783.16	2,282.57
Oxidizer Tank Mass (Empty) [kg]	4,997.82	3,445.17	885.84
Fuel Volume [m³]	347.50	239.55	196.46
Oxidizer Volume [m³]	430.16	296.52	76.24
Fuel Mass [kg]	152,900.97	105,399.94	13,919.14
Oxidizer Mass [kg]	490,812.12	338,333.81	86,994.61

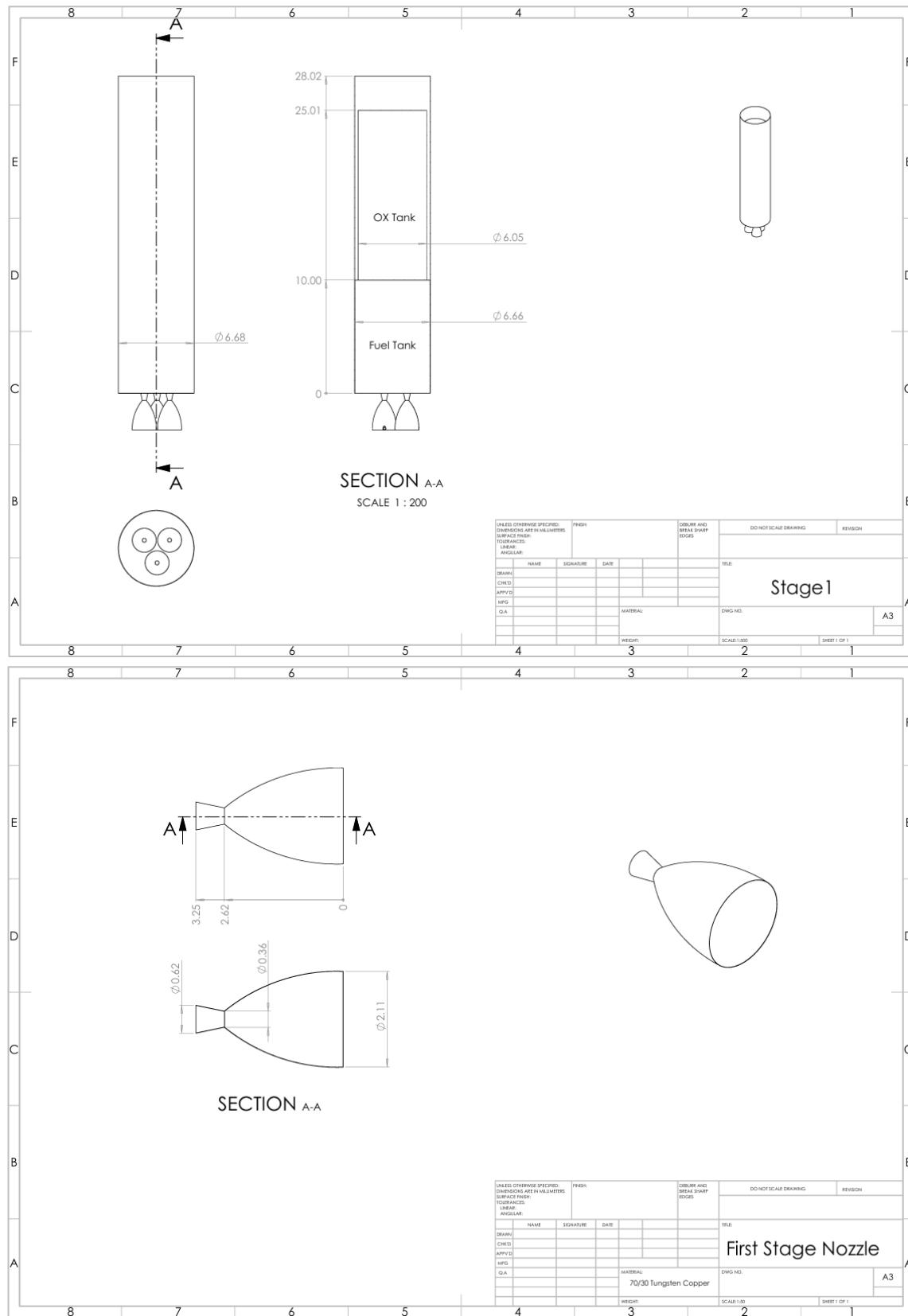
Table 6: Group contribution breakdown.

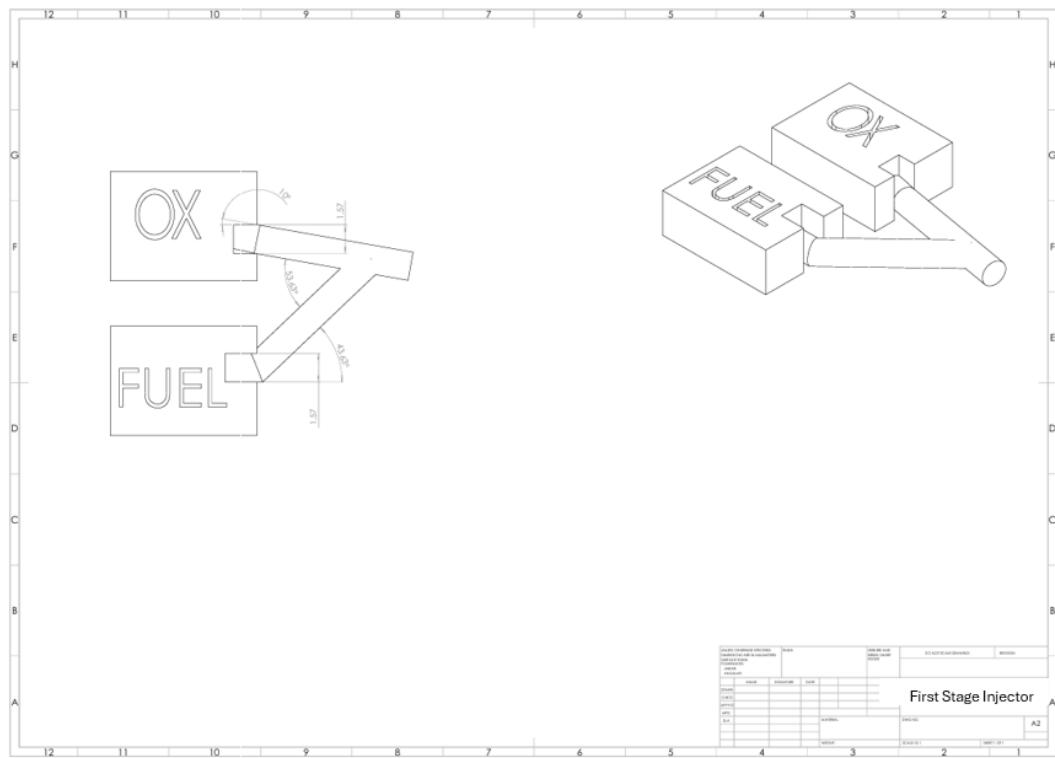
	Samuel Rockaway	Sukhyeon Lim	Jake Lancaster
Contact Information	srockawa@umich.edu	sukhyeon@umich.edu	jakelan@umich.edu
Conceptualization of Problem Contribution	Strongly contributed	Strongly Contributed	Strongly Contributed
Analysis Contribution	<ul style="list-style-type: none"> • Injectors • Fuel Tanks 	<ul style="list-style-type: none"> • Diverging portion of all engine nozzles 	<ul style="list-style-type: none"> • All CAD drawings

	<ul style="list-style-type: none"> Relevant masses of each stage 	<ul style="list-style-type: none"> NASA CEA 	<ul style="list-style-type: none"> Combustion Chamber Properties
Report, Equations, Figures, and Tables Contribution	<ul style="list-style-type: none"> Introduction Launch Config Upper Stage Contributions 	<ul style="list-style-type: none"> First Stage Conclusion 	<ul style="list-style-type: none"> Second Stage Overall Arrangement

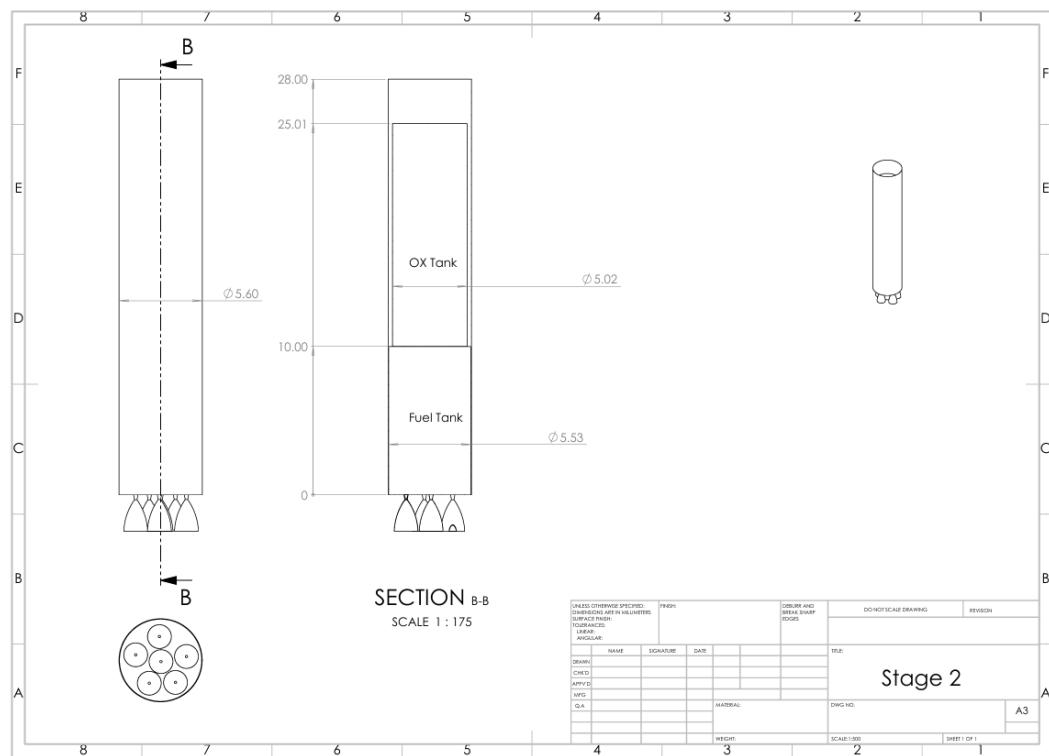
10. Appendix A

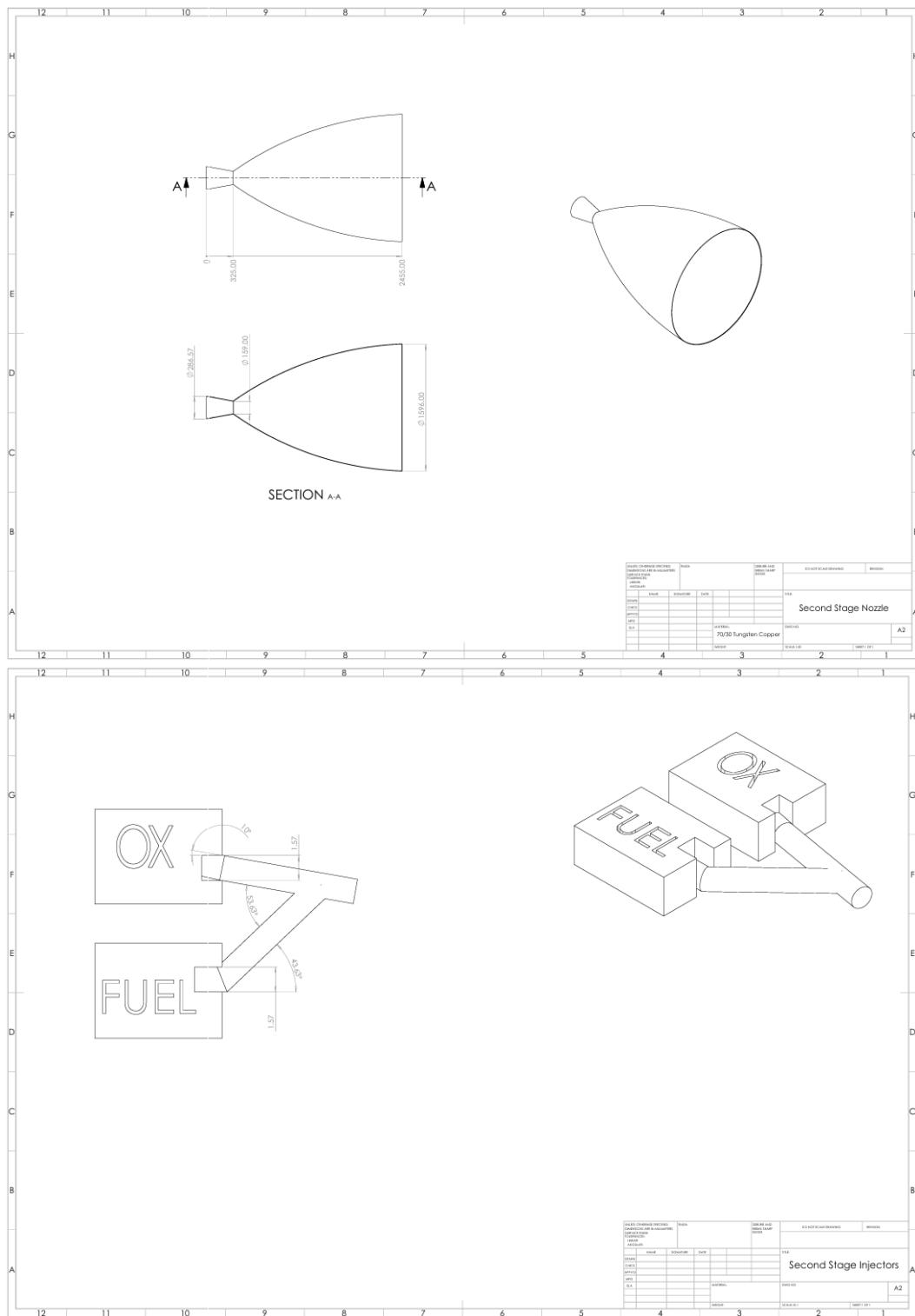


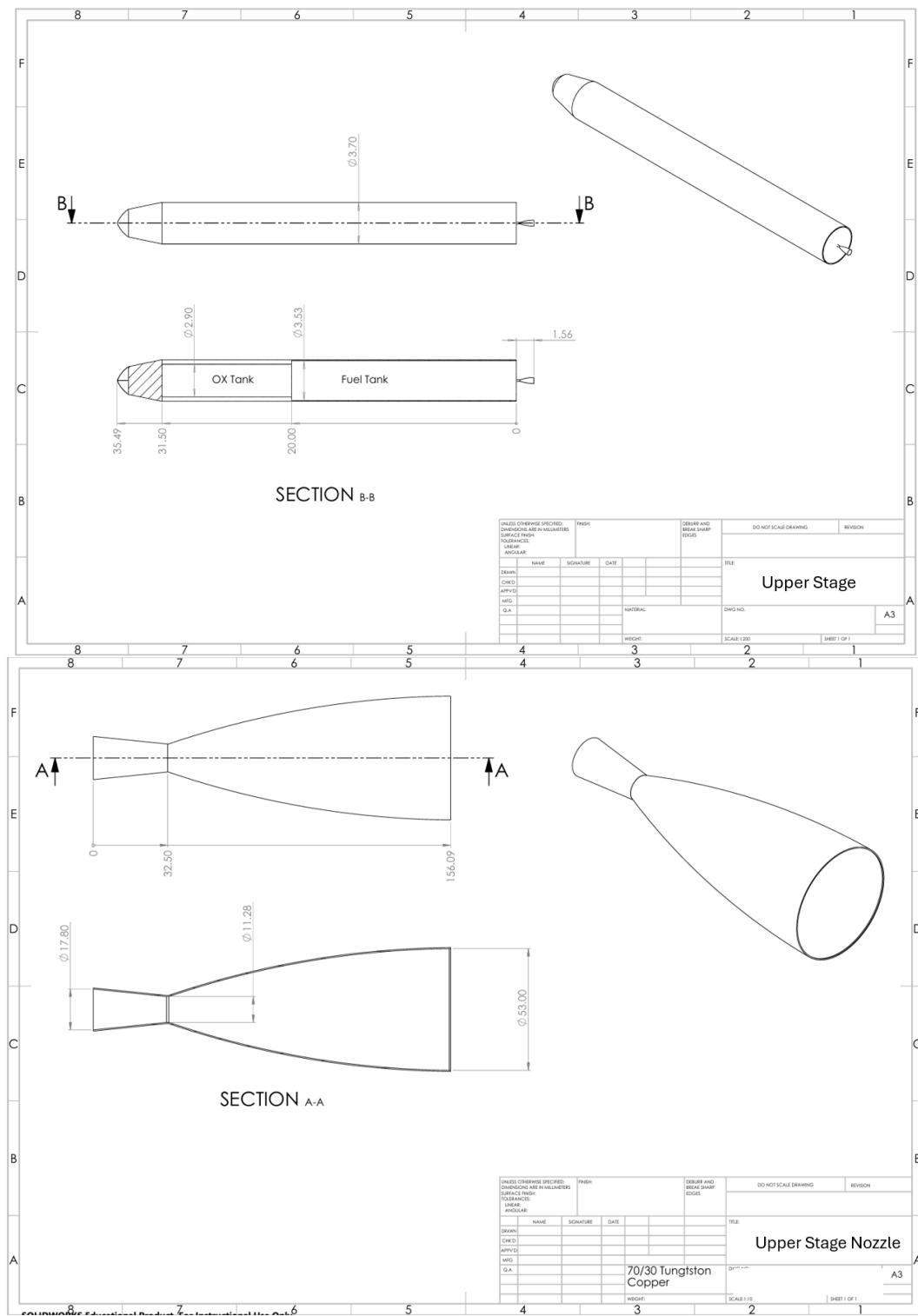


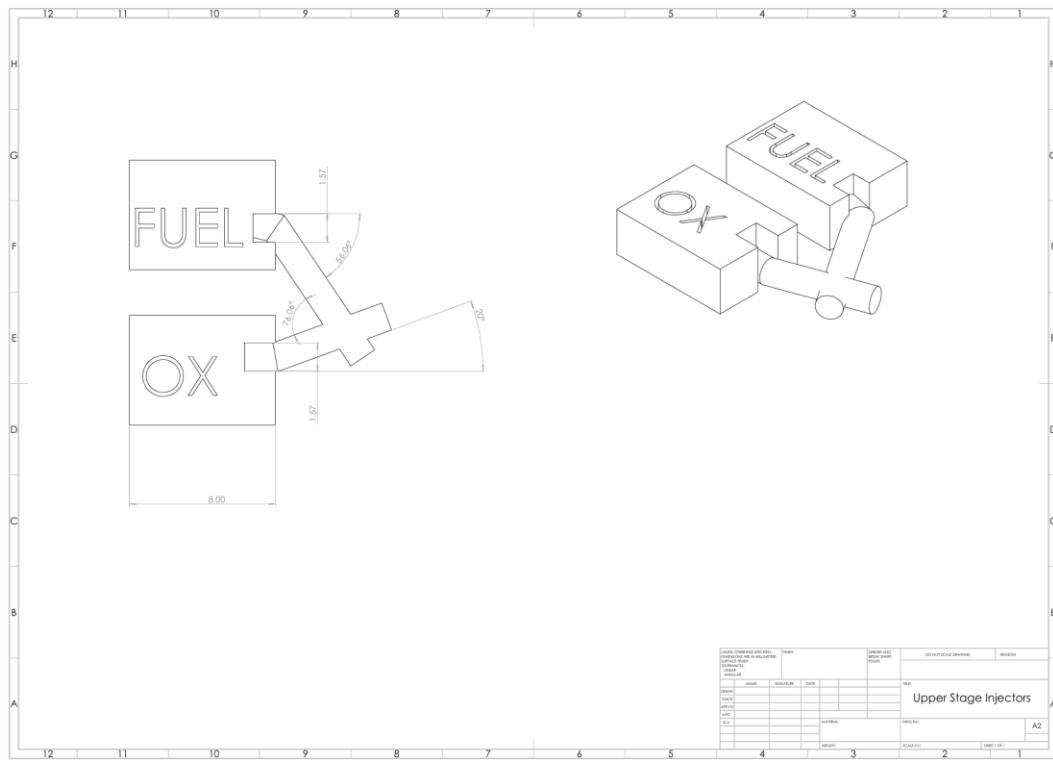


First Stage Injector









11. Appendix B

11.1 Isp Analysis

clear;

```

close all;
clc;

data_CEA = xlsread("data_CEA.xlsx");

OF_CEA = data_CEA(:,1);
T0_CEA = data_CEA(:,3); % Stagnation temperature (K)
p0_CEA = 1e5*data_CEA(:,2); % Stagnation pressure (Pa)
gamma_CEA = data_CEA(:,4); % Specific heat ratio
MW_CEA = 0.001*data_CEA(:,5); % Molecular weight (kg/mol)
g0 = 9.80665;

for n = 1:length(data_CEA(:,1))
    %% ----- Inputs and Constants -----
    % Combustion chamber properties
    T0_cc = T0_CEA(n,1); % Stagnation temperature (K)
    p0_cc = p0_CEA(n,1); % Stagnation pressure (Pa)
    gamma = gamma_CEA(n,1); % Specific heat ratio
    MW = MW_CEA(n,1); % Molecular weight (kg/mol)
    Ru = 8.31446261815324; % Universal gas constant (J/(mol·K))
    R_spec = Ru / MW_CEA(n,1); % Specific gas constant (J/(kg·K))

    % Ambient conditions
    altitude = linspace(0,20000,11);
    %altitude = [0];
    [~,~,pa,~,~,~] = atmосisa(altitude);
    altitude(1,length(altitude)+1) = 100000;
    pa(1,length(pa)+1) = 0;
    pa_p0 = pa./p0_cc; % Ambient to chamber pressure ratios to be analyzed
    %% ----- Ideal Specific Heat Calculation -----

    % Initialize result arrays for thrust coefficient and exit Mach number
    %C_F_opt = zeros(1, length(pa)); % Single array for C_F_opt corresponding to each pa_p0 value
    %Me_opt = zeros(size(pa_p0)); % Exit Mach number array for each pa_p0
    %Ae_At_FE = zeros(size(pa_p0)); % Exit Mach number array for each pa_p0

    % Iterate over the range of pressure ratios pa/p0
    for i = 1:length(pa_p0)
        % Calculate the optimal Mach number Me for the current pa_p0 ratio
        Me_opt(n,i) = sqrt((2 / (gamma - 1)) * ((1 / pa_p0(i))^(gamma - 1) / gamma) - 1);

        % Calculate the area ratio Ae/At corresponding to Mach number Me
        Ae_At_FE(n,i) = area_mach(Me_opt(i), gamma);

        % Calculate the thrust coefficient C_F_opt for each pressure ratio
        C_F_opt(n,i) = sqrt(2 * (gamma^2 / (gamma - 1)) * ...
            ((2 / (gamma + 1))^(gamma + 1) / (gamma - 1)) * ...
            (1 - (pa_p0(i))^(gamma - 1) / gamma));
        I_sp_opt(n,i) = C_F_opt(n,i) * sqrt((R_spec) * T0_cc / (gamma * (2/(gamma + 1))^(gamma + 1)/(gamma - 1))) ...
            / g0;
    end
end
%% ----- Plotting Results -----

% Plotting thrust coefficient C_F as a function of Ae/At for different pa/p0
a = 801;
b = 1;

```

```

f = figure;
f.Position = [0, 0, 1080, 800];
for i = 1:length(pa_p0)
for m = 1:b
    plot(OF_CEA(a*m-(a-1):a*m,1), I_sp_opt(a*m-(a-1):a*m,i),'DisplayName', sprintf('altitude = %.1f km, p_0 = %.0f bar', 1e-3*altitude(i), 1e-5*p0_CEA(a*m-(a-1),1)),'LineWidth',3);
    hold on
end
end
hold off
ax = gca; % Get current axes
ax.FontName = 'Times new roman';
ax.FontSize = 20;
%xticks(1.6:0.4:4)
xlim([1.5 9])
%ylim([200 620])
xlabel('O/F Ratio');
ylabel('I_{sp} (s)');
grid on;
legend show;
hold off;

for i = 1:length(pa_p0)
    for m = 1:b
        [max_Isp(i,m),x_max(i,m)] = max(I_sp_opt(a*m-(a-1):a*m,i));
        OFratio(i,m) = data_CEA(x_max(i,m),1);
        T0(i,m) = data_CEA(x_max(i,m),3);
        p0(i,m) = data_CEA(x_max(i,m),2);
        MW_max(i,m) = data_CEA(x_max(i,m),5);
        gamma_max(i,m) = data_CEA(x_max(i,m),4);
    end
end
altitude = transpose(altitude);
pa_p0 = transpose(pa_p0);

```

%% ----- Function Definitions -----

```

function area_ratio = area_mach(M, gamma)
    area_ratio = (1 ./ M) .* ((2 ./ (gamma + 1) .* (1 + 0.5 * (gamma - 1) .* M.^2)).^ ((gamma + 1) / (2 * (gamma - 1))));
end

```

```

% Function to compute Mach number from area ratio
function M = mach_from_area(A_ratio, gamma, flow)
    options = optimset('Display', 'off');
    if strcmp(flow, 'subsonic')
        M_guess = 0.5; % Initial guess for subsonic flow
    else
        M_guess = 2.0; % Initial guess for supersonic flow
    end
    M = fsolve(@(M) area_mach(M, gamma) - A_ratio, M_guess, options);
    if ~isreal(M) || M <= 0
        M = NaN;
    end

```

```

    end
end

% Function to compute isentropic pressure ratio
function p_p0 = isentropic_pressure_ratio(M, gamma)
    p_p0 = (1 + 0.5 * (gamma - 1) .* M.^2) .^ (-gamma / (gamma - 1));
end

% Function to compute downstream Mach number after normal shock
function M2 = normal_shock_M2(M1, gamma)
    M2_sq = ((gamma - 1) * M1.^2 + 2) / (2 * gamma * M1.^2 - (gamma - 1));
    if M2_sq > 0
        M2 = sqrt(M2_sq);
    else
        M2 = NaN;
    end
end

% Function to compute total pressure ratio across a normal shock
function p02_p01 = normal_shock_p02_p01(M1, gamma)
    term1 = ((gamma + 1) * M1.^2) / ((gamma - 1) * M1.^2 + 2);
    term2 = ((gamma + 1) / (2 * gamma * M1.^2 - (gamma - 1)));
    p02_p01 = term1 .^ (gamma / (gamma - 1)) * term2 .^ (1 / (gamma - 1));
end

% Function to determine backpressure condition
function [backpressure, pe_p0_FE, pe_p0_NS, pe_p0_CH, Me] = backpressure_function(Ae_At, gamma, pa_p0)
    % Fully expanded flow
    Me_FE = mach_from_area(Ae_At, gamma, 'supersonic');
    pe_p0_FE = isentropic_pressure_ratio(Me_FE, gamma);

    % Normal shock at exit
    M1_NS = Me_FE;
    M2_NS = normal_shock_M2(M1_NS, gamma);
    p02_p01_NS = normal_shock_p02_p01(M1_NS, gamma);
    pe_p0_NS = p02_p01_NS * isentropic_pressure_ratio(M2_NS, gamma);

    % Choked flow (subsonic solution)
    M_e_CH = mach_from_area(Ae_At, gamma, 'subsonic');
    pe_p0_CH = isentropic_pressure_ratio(M_e_CH, gamma);

    % Determine backpressure condition
    if pa_p0 < pe_p0_FE
        backpressure = 'underexpanded';
        Me = Me_FE;
    elseif pa_p0 > pe_p0_FE && pa_p0 <= pe_p0_NS
        backpressure = 'overexpanded(oblique_shock)';
        Me = Me_FE;
    elseif pa_p0 > pe_p0_NS && pa_p0 < pe_p0_CH
        backpressure = 'overexpanded(normal_shock)';
    end
end

```

```

Me = NaN; % Exit Mach number will be determined later
elseif abs(pa_p0 - pe_p0_FE) < 1e-6
    backpressure = 'fully_expanded';
    Me = Me_FE;
elseif pa_p0 > pe_p0_CH
    backpressure = 'subsonic';
    Me = mach_from_area(Ae_At, gamma, 'subsonic');
else
    backpressure = 'unknown';
    Me = NaN;
end
end

```

11.2 Area-ratio Analysis

```

% Rocket Nozzle Analysis Code (with Secondary Iteration for Shock Location)
% This script performs nozzle flow analysis for different flow regimes,
% calculates thrust, thrust coefficient, and specific impulse (Isp),
% and stores the relevant flow properties for multiple pa_p0 and Ae_At values.

```

```

clear;
close all;

```

%%% ----- Inputs and Constants -----

```

% Combustion chamber properties
T0_cc = 3699.7; % Stagnation temperature (K)
p0_cc = 300e5; % Stagnation pressure (Pa)
gamma = 1.1423; % Specific heat ratio
Mw = 0.001*21.49; % Molecular weight (kg/mol)
Ru = 8.31446261815324; % Universal gas constant (J/(mol·K))
R_spec = Ru / Mw; % Specific gas constant (J/(kg·K))

```

```

% Nozzle geometry
Ac_At = 5; % Chamber to throat area ratio
Ae_At = [16, 100, 8];

```

```

% Ambient conditions
altitude = linspace(0,20000,11);
[~,~,pa,~,~,~] = atmosisa(altitude);
pa_p0 = pa./p0_cc; % Ambient to chamber pressure ratios to be analyzed
% Tolerance for numerical methods
tolerance = 1e-7;

```

```

% Number of points for calculations
num_points = 10000; % Adjusted for computational efficiency

```

```

% Initialize arrays to store results
C_F = zeros(length(Ae_At), length(pa_p0));

```

```

l_sp = zeros(length(Ae_At), length(pa_p0));

%% ----- Flow Regime Determination -----

% Iterate over different Ae_At and pa_p0 values
for i = 1:length(Ae_At)
    for j = 1:length(pa_p0)
        current_Ae_At = Ae_At(i);
        current_pa_p0 = pa_p0(j);

        % Determine backpressure condition and exit Mach number
        [exit_pressure, pe_p0_FE, pe_p0_NS, pe_p0_CH, Me, Me_FE] = backpressure_function(current_Ae_At,
        gamma, current_pa_p0);

        fprintf('\n-- Analysis for Ae/At = %.2f and pa/p0 = %.4f --\n', current_Ae_At, current_pa_p0);

        switch exit_pressure
            case 'subsonic'
                fprintf('Subsonic flow condition. Me = %.6f\n', Me);
                pe_p0 = isentropic_pressure_ratio(Me, gamma);
                pe = pe_p0 * p0_cc;

            case {'fully_expanded', 'underexpanded', 'overexpanded(oblique_shock)'}
                fprintf('%s: Supersonic flow condition @exit, Me = %.6f\n', exit_pressure, Me);
                pe_p0 = isentropic_pressure_ratio(Me, gamma);
                pe = pe_p0 * p0_cc;

            case 'overexpanded(normal_shock)'
                % Define the range of area ratios in the diverging section
                Ax_At_min = 1 + 1e-6; % Slightly above the throat area ratio
                Ax_At_max = current_Ae_At - 1e-6; % Slightly below the exit area ratio
                num_points = 1000; % Number of points in the range for higher resolution
                Ax_At_values = linspace(Ax_At_min, Ax_At_max, num_points);
                tolerance = 1e-6; % Tolerance for matching p_e and p_a

                % Initialize variables to store the best match
                min_difference = inf;
                best_Ax_At = NaN;
                best_pe_cal = NaN;
                found_solution = false;

                options = optimset('Display', 'off');
                M_guess = 0.5; % Initial guess for subsonic flow

                for n = 1:length(Ax_At_values)
                    Ax_At = Ax_At_values(n);

                    % Compute M1 (supersonic solution)
                    M1 = mach_from_area(Ax_At, gamma, 'supersonic');
                    if ~isreal(M1) || M1 <= 1

```

```

    continue; % Skip invalid or subsonic solutions
end

% Compute downstream Mach number after normal shock
M2 = normal_shock_M2(M1, gamma);
if ~isreal(M2) || M2 <= 0
    continue; % Skip invalid solutions
end

% Compute area-Mach function for M2
f_M2 = area_mach(M2, gamma);

% Compute area ratio from shock to exit
Ae_Ax = current_Ae_At / Ax_At;

% Define the function to solve for Me
function_to_solve = @(Me_candidate) area_mach(Me_candidate, gamma) - Ae_Ax * f_M2;

% Use fsolve to find Me (subsonic solution)
Me_candidate = fsolve(function_to_solve, M_guess, options);

% Validity check
if ~isreal(Me_candidate) || Me_candidate <= 0 || Me_candidate >= 1
    continue; % Skip invalid solutions
end

% Compute exit pressure
pe_p0e = isentropic_pressure_ratio(Me_candidate, gamma);
p02_p01 = normal_shock_p02_p01(M1, gamma);
pe_cal = pe_p0e * p02_p01 * p0_cc;

% Calculate the difference between calculated and ambient pressures
difference = abs(pe_cal - pa);

% Check if this is the best match so far
if difference < min_difference
    min_difference = difference;
    best_Ax_At = Ax_At;
    best_pe_cal = pe_cal;
    Me = Me_candidate;
end

% Stop if within tolerance
if difference <= tolerance
    found_solution = true;
    break;
end
end

if isnan(best_Ax_At)

```

```

warning('No suitable shock location found within the given area ratios.');
fprintf('Minimum difference between p_e_cal and p_a: %.6f Pa\n', min_difference);
else
    % Secondary iteration: Refine around the best Ax_At found
    Ax_At_min_refined = max(1 + 1e-6, best_Ax_At - 0.1);
    Ax_At_max_refined = min(current_Ae_At - 1e-6, best_Ax_At + 0.1);
    num_points_refined = 10000;
    Ax_At_values_refined = linspace(Ax_At_min_refined, Ax_At_max_refined, num_points_refined);

    for n = 1:length(Ax_At_values_refined)
        Ax_At = Ax_At_values_refined(n);

        % Repeat the process for refined range
        M1 = mach_from_area(Ax_At, gamma, 'supersonic');
        if ~isreal(M1) || M1 <= 1
            continue;
        end

        M2 = normal_shock_M2(M1, gamma);
        if ~isreal(M2) || M2 <= 0
            continue;
        end

        f_M2 = area_mach(M2, gamma);
        Ae_Ax = current_Ae_At / Ax_At;
        Me_candidate = fsolve(@(Me_candidate) area_mach(Me_candidate, gamma) - Ae_Ax * f_M2,
        M_guess, options);

        if ~isreal(Me_candidate) || Me_candidate <= 0 || Me_candidate >= 1
            continue;
        end

        pe_p0e = isentropic_pressure_ratio(Me_candidate, gamma);
        p02_p01 = normal_shock_p02_p01(M1, gamma);
        pe_cal = pe_p0e * p02_p01 * p0_cc;

        difference = abs(pe_cal - pa);

        if difference < min_difference
            min_difference = difference;
            best_Ax_At = Ax_At;
            best_pe_cal = pe_cal;
            Me = Me_candidate;
        end

        if difference <= tolerance
            break;
        end
    end
end

```

```

fprintf('Refined shock location at Ax/At = %.6f\n', best_Ax_At);
pe = best_pe_cal; % Exit pressure from refined shock location
end

otherwise
error('Unknown flow condition: %s', exit_pressure);
end

%% ----- Main Computation -----

% Calculate thrust coefficient and specific impulse
At = 0.02; % Throat area (arbitrary value, as it cancels out in C_F)
Ae = current_Ae_At * At; % Exit area

% Thrust coefficient (C_F)
C_F(i, j) = sqrt(2 * (gamma^2 / (gamma - 1)) * ((2 / (gamma + 1))^(gamma + 1 / (gamma - 1))) * (1 -
(pe_p0)^(gamma - 1 / gamma)) + ((pe - pa(j)) * (current_Ae_At)) / p0_cc;
% Specific impulse (I_sp)
g0 = 9.81; % Gravity constant (m/s^2)
I_sp(i, j) = C_F(i, j) * sqrt(R_spec * T0_cc / (gamma * (2/(gamma + 1))^(gamma + 1)/(gamma - 1))) / g0;
F_t(i,j) = C_F(i,j) * p0_cc * At;
m_dot(i,j) = At*p0_cc*sqrt((gamma*(2/(gamma + 1))^(gamma + 1)/(gamma - 1))/(R_spec*T0_cc));

end
end

% Initialize result arrays for thrust coefficient and exit Mach number
C_F_opt = zeros(1, length(pa)); % Single array for C_F_opt corresponding to each pa_p0 value
Me_opt = zeros(size(pa_p0)); % Exit Mach number array for each pa_p0
Ae_At_FE = zeros(size(pa_p0)); % Exit Mach number array for each pa_p0

% Iterate over the range of pressure ratios pa/p0
for i = 1:length(pa_p0)
    % Calculate the optimal Mach number Me for the current pa_p0 ratio
    Me_opt(i) = sqrt((2 / (gamma - 1)) * ((1 / pa_p0(i))^(gamma - 1 / gamma) - 1));
    
    % Calculate the area ratio Ae/At corresponding to Mach number Me
    Ae_At_FE(i) = area_mach(Me_opt(i), gamma);

    % Calculate the thrust coefficient C_F_opt for each pressure ratio
    C_F_opt(i) = sqrt(2 * (gamma^2 / (gamma - 1)) * ...
        ((2 / (gamma + 1))^(gamma + 1 / (gamma - 1))) * ...
        (1 - (pa_p0(i))^(gamma - 1 / gamma)));
    I_sp_opt(i) = C_F_opt(i) * sqrt((R_spec) * T0_cc / (gamma * (2/(gamma + 1))^(gamma + 1)/(gamma - 1))) /
g0;

end

%% ----- Plotting Results -----

```

```
% Plotting thrust coefficient C_F as a function of Ae/At for different pa/p0
```

```
f = figure;
f.Position = [0, 0, 1080, 800];
plot(1e-3*altitude, C_F,'LineWidth',3);
hold on
plot(1e-3*altitude, C_F_opt,'LineWidth',3);
ax = gca; % Get current axes
ax.FontName = 'Times new roman';
ax.FontSize = 20;
xlabel('Altitude (km)');
ylabel('C_F');

grid on;
legend('A_e/A_t = 33','A_e/A_t = 100','Variable A_e/A_t');
hold off;
```

```
%% ----- Function Definitions -----
```

```
function area_ratio = area_mach(M, gamma)
    area_ratio = (1 ./ M) .* ((2 ./ (gamma + 1) .* (1 + 0.5 * (gamma - 1) .* M.^2)).^ ((gamma + 1) / (2 * (gamma - 1))));
end
```

```
% Function to compute Mach number from area ratio
function M = mach_from_area(A_ratio, gamma, flow)
    options = optimset('Display', 'off');
    if strcmp(flow, 'subsonic')
        M_guess = 0.5; % Initial guess for subsonic flow
    else
        M_guess = 2.0; % Initial guess for supersonic flow
    end
    M = fsolve(@(M) area_mach(M, gamma) - A_ratio, M_guess, options);
    if ~isreal(M) || M <= 0
        M = NaN;
    end
end
```

```
% Function to compute isentropic pressure ratio
function p_p0 = isentropic_pressure_ratio(M, gamma)
    p_p0 = (1 + 0.5 * (gamma - 1) .* M.^2) .^ (-gamma / (gamma - 1));
end
```

```
% Function to compute downstream Mach number after normal shock
function M2 = normal_shock_M2(M1, gamma)
    M2_sq = ((gamma - 1) * M1.^2 + 2) / (2 * gamma * M1.^2 - (gamma - 1));
    if M2_sq > 0
        M2 = sqrt(M2_sq);
    else
        M2 = NaN;
```

```

end
end

% Function to compute total pressure ratio across a normal shock
function p02_p01 = normal_shock_p02_p01(M1, gamma)
    term1 = ((gamma + 1) * M1.^2) / ((gamma - 1) * M1.^2 + 2);
    term2 = ((gamma + 1) / (2 * gamma * M1.^2 - (gamma - 1)));
    p02_p01 = term1 .^ (gamma / (gamma - 1)) * term2 .^ (1 / (gamma - 1));
end

% Function to determine backpressure condition
function [backpressure, pe_p0_FE, pe_p0_NS, pe_p0_CH, Me, Me_FE] = backpressure_function(Ae_At, gamma, pa_p0)
    % Fully expanded flow
    Me_FE = mach_from_area(Ae_At, gamma, 'supersonic');
    pe_p0_FE = isentropic_pressure_ratio(Me_FE, gamma);

    % Normal shock at exit
    M1_NS = Me_FE;
    M2_NS = normal_shock_M2(M1_NS, gamma);
    p02_p01_NS = normal_shock_p02_p01(M1_NS, gamma);
    pe_p0_NS = p02_p01_NS * isentropic_pressure_ratio(M2_NS, gamma);

    % Choked flow (subsonic solution)
    M_e_CH = mach_from_area(Ae_At, gamma, 'subsonic');
    pe_p0_CH = isentropic_pressure_ratio(M_e_CH, gamma);

    % Determine backpressure condition
    if pa_p0 < pe_p0_FE
        backpressure = 'underexpanded';
        Me = Me_FE;
    elseif pa_p0 > pe_p0_FE && pa_p0 <= pe_p0_NS
        backpressure = 'overexpanded(oblique_shock)';
        Me = Me_FE;
    elseif pa_p0 > pe_p0_NS && pa_p0 < pe_p0_CH
        backpressure = 'overexpanded(normal_shock)';
        Me = NaN; % Exit Mach number will be determined later
    elseif abs(pa_p0 - pe_p0_FE) < 1e-6
        backpressure = 'fully_expanded';
        Me = Me_FE;
    elseif pa_p0 > pe_p0_CH
        backpressure = 'subsonic';
        Me = mach_from_area(Ae_At, gamma, 'subsonic');
    else
        backpressure = 'unknown';
        Me = NaN;
    end
end

```

11.3 Bell-Shaped Nozzle Design

```
import math
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.patches import Arc
from bisect import bisect_left
```

=====

Reference: The thrust optimised parabolic (Rao) nozzle as per:
<http://www.aspirespace.org.uk/downloads/Thrust%20optimised%20parabolic%20nozzle.pdf>

Main Equations:

$$[\text{Eqn. 2}] \quad Re = \sqrt{\varepsilon} * Rt$$

$$[\text{Eqn. 3}] \quad LN = 0.8 * ((\sqrt{\varepsilon} - 1) * Rt) / \tan(15^\circ)$$

Entrant section:

$$[\text{Eqn. 4}] \quad x = 1.5 Rt \cos\theta$$

$$y = 1.5 Rt \sin\theta + 1.5 Rt + Rt$$

where: $-135 \leq \theta \leq -90$ degrees

Exit section:

$$[\text{Eqn. 5}] \quad x = 0.382 Rt \cos\theta$$

$$y = 0.382 Rt \sin\theta + 0.382 Rt + Rt$$

where: $-90 \leq \theta \leq (\theta_n - 90)$ degrees

The bell is a quadratic Bézier curve:

$$[\text{Eqn. 6}] \quad x(t) = (1-t)^2 Nx + 2(1-t)t Qx + t^2 Ex$$

$$y(t) = (1-t)^2 Ny + 2(1-t)t Qy + t^2 Ey, \text{ for } 0 \leq t \leq 1$$

Point N is defined by [Eqn. 5] at angle $(\theta_n - 90)$.

Point E is defined by Ex (from Eqn.3) and Ey (from Eqn.2).

Point Q is intersection of lines NQ and QE:

$$m1 = \tan(\theta_n), m2 = \tan(\theta_e)$$

$$C1 = Ny - m1 Nx$$

$$C2 = Ey - m2 Ex$$

$$Qx = (C2 - C1) / (m1 - m2)$$

$$Qy = (m1 C2 - m2 C1) / (m1 - m2)$$

=====

```
def bell_nozzle(k, aratio, Rt, l_percent):
    # Default entrant angle (usually -135 degrees)
    entrant_angle = -135
    ea_radian = math.radians(entrant_angle)

    # Define Lnp based on l_percent
    if l_percent == 60:
        Lnp = 0.6
```

```

elif l_percent == 80:
    Lnp = 0.8
elif l_percent == 90:
    Lnp = 0.9
else:
    Lnp = 0.8

# Find wall angles and nozzle length
angles = find_wall_angles(aratio, Rt, l_percent)
nozzle_length = angles[0]
theta_n = angles[1]
theta_e = angles[2]

data_intervel = 100

# Entrant section (Eqn.4)
ea_start = ea_radian
ea_end = -math.pi/2
angle_list = np.linspace(ea_start, ea_end, data_intervel)
xe = []
ye = []
for i in angle_list:
    xe.append(1.5 * Rt * math.cos(i))
    ye.append(1.5 * Rt * math.sin(i) + 2.5 * Rt)

# Exit section (Eqn.5)
ea_start = -math.pi/2
ea_end = theta_n - math.pi/2
angle_list = np.linspace(ea_start, ea_end, data_intervel)
xe2 = []
ye2 = []
for i in angle_list:
    xe2.append(0.382 * Rt * math.cos(i))
    ye2.append(0.382 * Rt * math.sin(i) + 1.382 * Rt)

# Bell section
# N (using eqn.5 at angle ( $\theta_n - 90^\circ$ ))
Nx = 0.382 * Rt * math.cos(theta_n - math.pi/2)
Ny = 0.382 * Rt * math.sin(theta_n - math.pi/2) + 1.382 * Rt
# E (from eqn.3 and eqn.2)
Ex = Lnp * ((math.sqrt(aratio) - 1) * Rt) / math.tan(math.radians(15))
Ey = math.sqrt(aratio) * Rt
# Gradients m1, m2 (Eqn.8)
m1 = math.tan(theta_n)
m2 = math.tan(theta_e)
# Intercepts C1, C2 (Eqn.9)
C1 = Ny - m1 * Nx
C2 = Ey - m2 * Ex
# Intersection Q (Eqn.10)
Qx = (C2 - C1) / (m1 - m2)

```

```

Qy = (m1 * C2 - m2 * C1) / (m1 - m2)

# Quadratic Bézier curve (Eqn.6)
int_list = np.linspace(0, 1, data_intervel)
xbell = []
ybell = []
for t in int_list:
    xbell.append(((1 - t)**2) * Nx + 2 * (1 - t) * t * Qx + (t**2) * Ex)
    ybell.append(((1 - t)**2) * Ny + 2 * (1 - t) * t * Qy + (t**2) * Ey)

# Mirror for lower half
nye = [-y for y in ye]
nye2 = [-y for y in ye2]
nybell = [-y for y in ybell]

return angles, (xe, ye, nye, xe2, ye2, nye2, xbell, ybell, nybell)

def find_wall_angles(ar, Rt, l_percent=80):
    # Wall-angle empirical data
    arvals = [4, 5, 10, 20, 30, 40, 50, 100]
    theta_n_60 = [26.5, 28.0, 32.0, 35.0, 36.2, 37.1, 35.0, 40.0]
    theta_n_80 = [21.5, 23.0, 26.3, 28.8, 30.0, 31.0, 31.5, 33.5]
    theta_n_90 = [20.0, 21.0, 24.0, 27.0, 28.5, 29.5, 30.2, 32.0]

    theta_e_60 = [20.5, 20.5, 16.0, 14.5, 14.0, 13.5, 13.0, 11.2]
    theta_e_80 = [14.0, 13.0, 11.0, 9.0, 8.5, 8.0, 7.5, 7.0]
    theta_e_90 = [11.5, 10.5, 8.0, 7.0, 6.5, 6.0, 6.0, 6.0]

    f1 = ((math.sqrt(ar) - 1) * Rt) / math.tan(math.radians(15))

    if l_percent == 60:
        theta_n_list = theta_n_60
        theta_e_list = theta_e_60
        Ln = 0.6 * f1
    elif l_percent == 80:
        theta_n_list = theta_n_80
        theta_e_list = theta_e_80
        Ln = 0.8 * f1
    elif l_percent == 90:
        theta_n_list = theta_n_90
        theta_e_list = theta_e_90
        Ln = 0.9 * f1
    else:
        theta_n_list = theta_n_80
        theta_e_list = theta_e_80
        Ln = 0.8 * f1

    # Find nearest AR index
    x_index, x_val = find_nearest(arvals, ar)

```

```

if round(arvals[x_index], 1) == round(ar, 1):
    # exact match
    return Ln, math.radians(theta_n_list[x_index]), math.radians(theta_e_list[x_index])

# If not exact, we interpolate
# Choose a slice around x_index
if x_index > 2 and x_index < len(arvals)-2:
    ar_slice = arvals[x_index-2:x_index+2]
    tn_slice = theta_n_list[x_index-2:x_index+2]
    te_slice = theta_e_list[x_index-2:x_index+2]
elif (len(arvals)-x_index) <= 1:
    # near the end
    ar_slice = arvals[x_index-2:len(arvals)]
    tn_slice = theta_n_list[x_index-2:len(arvals)]
    te_slice = theta_e_list[x_index-2:len(arvals)]
else:
    # near the start
    ar_slice = arvals[0:x_index+2]
    tn_slice = theta_n_list[0:x_index+2]
    te_slice = theta_e_list[0:x_index+2]

tn_val = interpolate(ar_slice, tn_slice, ar)
te_val = interpolate(ar_slice, te_slice, ar)

return Ln, math.radians(tn_val), math.radians(te_val)

def interpolate(x_list, y_list, x):
    # Check strictly ascending order
    if any((x2 - x1) <= 0 for x1, x2 in zip(x_list, x_list[1:])):
        raise ValueError("x_list must be in strictly ascending order!")

    intervals = list(zip(x_list, x_list[1:], y_list, y_list[1:]))
    slopes = [(y2 - y1) / (x2 - x1) for x1, x2, y1, y2 in intervals]

    if x <= x_list[0]:
        return y_list[0]
    elif x >= x_list[-1]:
        return y_list[-1]
    else:
        i = bisect_left(x_list, x) - 1
        return y_list[i] + slopes[i] * (x - x_list[i])

def find_nearest(array, value):
    array = np.asarray(array)
    idx = (np.abs(array - value)).argmin()
    return idx, array[idx]

```

```

def plot_nozzle(ax, title, Rt, angles, contour):
    nozzle_length = angles[0]
    theta_n = angles[1]
    theta_e = angles[2]

    xe, ye, nye, xe2, ye2, nye2, xbell, ybell, nybell = contour

    ax.set_aspect('equal')

    # Throat entrant
    ax.plot(xe, ye, linewidth=2.5, color='g')
    ax.plot(xe, nye, linewidth=2.5, color='g')

    # Inlet radius dimension
    x1 = xe[0]; y1 = 0
    x2 = xe[0]; y2 = nye[0]
    dist = math.sqrt((x2 - x1)**2 + (y2 - y1)**2)
    text = ' Ri = ' + str(round(dist, 1))
    ax.plot(xe[0], 0, '+')
    ax.annotate("", [x1, y1], [x2, y2], arrowprops=dict(lw=0.5, arrowstyle='<->'))
    ax.text((x1+x2)/2, (y1+y2)/2, text, fontsize=9)

    # Nozzle inlet length
    text = ' Li = ' + str(round(abs(xe[0]), 1))
    ax.plot(0, 0, '+')
    ax.annotate("", [0, 0], [xe[0], 0], arrowprops=dict(lw=0.5, arrowstyle='<->'))
    ax.text((xe[0]/2), 0, text, fontsize=9)

    # Enterant radius line
    i = int(len(xe)/2)
    xcenter = 0; ycenter = 2.5 * Rt
    xarch = xe[i]; yarch = ye[i]
    text = ' 1.5 * Rt = ' + str(round(1.5*Rt, 1))
    ax.plot(xcenter, ycenter, '+')
    ax.annotate("", [xcenter,ycenter], [xarch,yarch], arrowprops=dict(lw=0.5, arrowstyle='<->'))
    #ax.text((xarch+xcenter)/2, (yarch+ycenter)/2, text, fontsize=9)

    # Throat radius line
    text = ' Rt = ' + str(Rt)
    ax.annotate("", [0,0], [xe[-1], ye[-1]], arrowprops=dict(lw=0.5, arrowstyle='<->'))
    ax.text(xe[-1]/2, ye[-1]/2, text, fontsize=9)

    # Throat exit
    ax.plot(xe2, ye2, linewidth=2.5, color='r')
    ax.plot(xe2, nye2, linewidth=2.5, color='r')
    i = int(len(xe2)/2)
    xcenter2 = 0; ycenter2 = 1.382 * Rt
    xarch2 = xe2[i]; yarch2 = ye2[i]
    text = ' 0.382 * Rt = ' + str(round(0.382 * Rt, 1))
    ax.plot(xcenter2, ycenter2, '+')
    ax.annotate("", [xcenter2,ycenter2], [xarch2,yarch2], arrowprops=dict(lw=0.5, arrowstyle='<->'))

```

```

ax.annotate("", [xcenter2,ycenter2], [xarch2,yarch2], arrowprops=dict(lw=0.5, arrowstyle='<->'))
#ax.text((xarch2+xcenter2)/2, (yarch2+ycenter2)/2, text, fontsize=9)

# Draw theta_n arc
adj_text = 2
origin = [xe2[-1], nye2[-1]-adj_text]
#draw_angle_arc(ax, theta_n, origin, r'$\theta_n$')

# Bell section
ax.plot(xbell, ybell, linewidth=2.5, color='b')
ax.plot(xbell, nybell, linewidth=2.5, color='b')

# Exit radius line
text = ' Re = ' + str(round((math.sqrt((ybell[-1]/Rt)**2)*Rt), 1))
ax.plot(xbell[-1], 0, '+')
ax.annotate("", [xbell[-1],0], [xbell[-1], ybell[-1]], arrowprops=dict(lw=0.5, arrowstyle='<->'))
ax.text(xbell[-1], ybell[-1]/2, text, fontsize=9)

# Draw theta_e arc
origin = [xbell[-1], nybell[-1]]
#draw_angle_arc(ax, theta_e, origin, r'$\theta_e$')

# nozzle length line
text = ' Ln = ' + str(round(nozzle_length, 1))
ax.annotate("", [0,0], [xbell[-1], 0], arrowprops=dict(lw=0.5, arrowstyle='<->'))
ax.text(xbell[-1]/2, 0, text, fontsize=9)

ax.axhline(color='black', lw=0.5, linestyle="dashed")
ax.axvline(color='black', lw=0.5, linestyle="dashed")

ax.grid(True)
plt.title(title, fontsize=9)

def draw_angle_arc(ax, angle_rad, origin, degree_symbol=r'$\theta$'):
    # Convert angle to degrees for display
    angle_deg = math.degrees(angle_rad)
    startx, starty = origin
    length = 50
    endx = startx + np.cos(-angle_rad) * length * 0.5
    endy = starty + np.sin(-angle_rad) * length * 0.5
    # Draw the angled line
    ax.plot([startx,endx], [starty,endy], linewidth=0.5, color='k')
    # Draw arc
    arc_obj = Arc([startx, starty], 1, 1, angle=0, theta1=0, theta2=angle_deg, color='k')
    ax.add_patch(arc_obj)
    #ax.text(startx+0.5, starty+0.5, f'{degree_symbol} = {round(angle_deg,1)}°')

def ring(r, h, a=0, n_theta=30, n_height=10):

```

```

theta = np.linspace(0, 2*np.pi, n_theta)
v = np.linspace(a, a+h, n_height)
theta, v = np.meshgrid(theta, v)
x = r*np.cos(theta)
y = r*np.sin(theta)
z = v
return x, y, z

def set_axes_equal_3d(ax: plt.Axes):
    limits = np.array([
        ax.get_xlim3d(),
        ax.get_ylim3d(),
        ax.get_zlim3d(),
    ])
    origin = np.mean(limits, axis=1)
    radius = 0.5 * np.max(np.abs(limits[:, 1] - limits[:, 0]))
    _set_axes_radius(ax, origin, radius)

def _set_axes_radius(ax, origin, radius):
    x, y, z = origin
    ax.set_xlim3d([x - radius, x + radius])
    ax.set_ylim3d([y - radius, y + radius])
    ax.set_zlim3d([z - radius, z + radius])

def plot3D(ax, contour):
    xe, ye, nye, xe2, ye2, nye2, xbell, ybell, nybell = contour
    # Combine top half of the nozzle contour for revolution
    x = np.concatenate((xe, xe2, xbell))
    y = np.concatenate((ye, ye2, ybell))

    # ring thickness
    if len(x) > 1:
        thick = 5 * (x[1] - x[0])
    else:
        thick = 0.01

    # Create a 3D surface by revolving the contour around the center line
    for i in range(len(y)):
        X, Y, Z = ring(y[i], thick, x[i])
        ax.plot_surface(X, Y, Z, color='g')

    ax.set_box_aspect([1, 1, 1])
    set_axes_equal_3d(ax)
    ax.view_init(-170, -15)

def plot(title, throat_radius, angles, contour):

```

```

fig = plt.figure(figsize=(12,10))
ax1 = fig.add_subplot(121)
plot_nozzle(ax1, title, throat_radius, angles, contour)

ax2 = fig.add_subplot(122, projection='3d')
plot3D(ax2, contour)

fig.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()

if __name__=="__main__":
    # Constants
    k = 1.1443
    l_percent = 100

    # Example 1
    aratio = 33
    throat_radius = 17.84
    angles, contour = bell_nozzle(k, aratio, throat_radius, l_percent)
    title = f'{aratio}x'
    plot(title, throat_radius, angles, contour)

    # Example 2
    aratio = 100
    throat_radius = 7.9
    angles, contour = bell_nozzle(k, aratio, throat_radius, l_percent)
    title = f'{aratio}x'
    plot(title, throat_radius, angles, contour)

```

11.4 Stage, Propellant Tank and Injector Sizing

```

%% Propellant Tank Volume Calc
% Find mass ratio, payload ratio, and structural coeff of stage

clc
clear all
close all

% Given
mpl = 5000; %[kg]
Isp = 440.94; %[s]
g0 = 9.81;
mix_r = 6.25;
rho_fuel = 70.85; %70.85 for LH2 440 for CH4
rho_oxidizer = 1141;
rho_wall = 6000; %kg/m^3
delV = 1000*(0.2 + 2.46 + 1.48 + 0.68 + 0.14 + 0.68 + 0.68 + 0.14 + 3.14); %[m/s]
%delV = 2500;
mdot = 130.8;
Pc = 3e7; %[pa]

```

```
%-----Mass Ratios-----
epsilon = 0.0675;

R = (exp(delV/(Isp*g0)));

lambda = (epsilon*(exp(delV/(Isp*g0))) - 1) / (1 - exp(delV/(Isp*g0))); % Equation is good

m0 = (mpl + lambda*mpl)/(lambda);

mfoverm0 = (lambda + epsilon) / (lambda + 1) ;

mf = m0*mfoverm0;

ms = mf - mpl;

mp = m0 - ms - mpl;

massratio = m0/mf;

%-----Tank Sizing-----
% Propellant Masses
m_fuel = mp/ (mix_r + 1);
m_oxidizer = mp - m_fuel;

% Num Tanks for each propellant
Num_Tanks = 1;

% Propellant Volume
V_fuel = (m_fuel/rho_fuel)/Num_Tanks;
V_oxidizer = (m_oxidizer/rho_oxidizer)/Num_Tanks;

% Set Heights
h_fuel = 15;
h_oxidizer = 20;

% Tank Diameter Cylinder
D_fuel = 2*sqrt(V_fuel / (pi*h_fuel));
D_oxidizer = 2*sqrt(V_oxidizer / (pi*h_oxidizer));

% Wall thickness
tank_pressure = 0.3e6; %pa

t_fuel = (tank_pressure*D_fuel)/(2 * 310e6);
t_oxidizer = (tank_pressure*D_oxidizer)/(2 * 310e6);

% Volume of wall
V_outer_fuel = (pi*((2*t_fuel + D_fuel)^2) * h_fuel) / 4;
V_inner_fuel = V_fuel;
Wall_volume_fuel = V_outer_fuel - V_inner_fuel;

V_outer_oxidizer = (pi*((2*t_oxidizer + D_oxidizer)^2) * h_oxidizer) / 4;
V_inner_oxidizer = V_oxidizer;
Wall_volume_oxidizer = V_outer_oxidizer - V_inner_oxidizer;

% Mass of Wall
m_fuel_tank = Wall_volume_fuel*rho_wall;
m_oxidizer_tank = Wall_volume_oxidizer*rho_wall;
```

```
% Outer Diameter
OD_fuel = 2*t_fuel + D_fuel;
OD_oxidizer = 2*t_oxidizer + D_oxidizer;

%-----Injector Sizing-----

% Total Quantities
mdot_fuel = mdot / (1 + mix_r);
mdot_oxidizer = mdot - mdot_fuel;

% Short Tube w/ rounded entrance L/D > 3
D = .00157;
Cd = 0.90;
A = pi*(D/2)^2;

DelP_solution_fuel = 0.2*Pc;

Pf_Pox = (rho_oxidizer/rho_fuel)/((mix_r^2));
DelP_solution_oxidizer = DelP_solution_fuel / Pf_Pox;

mdot_fuel_injector_i = Cd * A* sqrt(2 * rho_fuel * DelP_solution_fuel);
mdot_oxidizer_injector_i = Cd * A *sqrt(2 * rho_oxidizer * DelP_solution_oxidizer);

v_fuel = Cd * sqrt((2 * DelP_solution_fuel) / rho_fuel);
v_oxidizer = Cd * sqrt((2 * DelP_solution_oxidizer) / rho_oxidizer);

a2 = 10; % PLAY WITH THIS
a1 = asind(((mdot_oxidizer_injector_i * v_oxidizer) / (mdot_fuel_injector_i * v_fuel)) *
sin(a2));

Num_fuel_injectors = mdot_fuel / mdot_fuel_injector_i;
Num_oxidizer_injectors = mdot_oxidizer / mdot_oxidizer_injector_i;

% Display results
fprintf('mdot_oxidizer_injector_i / mdot_fuel_injector_i = %.2f\n', ...
    mdot_oxidizer_injector_i / mdot_fuel_injector_i);
fprintf('a1 = %.2f degrees\n', a1);
fprintf('a2 = %.2f degrees\n', a2); %% Propellant Tank Volume Calc
% Find mass ratio, payload ratio, and structural coeff of upper

clc
clear all
close all

% Given
mpl = 5000; %[kg]
Isp = 440.94; %[s]
g0 = 9.81;
mix_r = 6.25;
rho_fuel = 70.85; %70.85 for LH2 440 for CH4
rho_oxidizer = 1141;
rho_wall = 6000; %kg/m^3
delV = 1000*(0.2 + 2.46 + 1.48 + 0.68 + 0.14 + 0.68 + 0.68 + 0.14 + 3.14); %[m/s]
%delV = 2500;
mdot = 130.8;
```

```

Pc = 3e7; %[pa]

%-----Mass Ratios-----
epsilon = 0.0675;

R = (exp(delV/(Isp*g0)));

lambda = (epsilon*(exp(delV/(Isp*g0))) - 1) / (1 - exp(delV/(Isp*g0))); % Equation is good

m0 = (mpl + lambda*mpl)/(lambda);

mfoverm0 = (lambda + epsilon) / (lambda + 1) ;

mf = m0*mfoverm0;

ms = mf - mpl;

mp = m0 - ms - mpl;

massratio = m0/mf;

%-----Tank Sizing-----
% Propellant Masses
m_fuel = mp/ (mix_r + 1);
m_oxidizer = mp - m_fuel;

% Num Tanks for each propellant
Num_Tanks = 1;

% Propellant Volume
V_fuel = (m_fuel/rho_fuel)/Num_Tanks;
V_oxidizer = (m_oxidizer/rho_oxidizer)/Num_Tanks;

% Set Heights
h_fuel = 15;
h_oxidizer = 20;

% Tank Diameter Cylinder
D_fuel = 2*sqrt(V_fuel / (pi*h_fuel));
D_oxidizer = 2*sqrt(V_oxidizer / (pi*h_oxidizer));

% Wall thickness
tank_pressure = 0.3e6; %pa

t_fuel = (tank_pressure*D_fuel)/(2 * 310e6);
t_oxidizer = (tank_pressure*D_oxidizer)/(2 * 310e6);

% Volume of wall
V_outer_fuel = (pi*((2*t_fuel + D_fuel)^2) * h_fuel) / 4;
V_inner_fuel = V_fuel;
Wall_volume_fuel = V_outer_fuel - V_inner_fuel;

V_outer_oxidizer = (pi*((2*t_oxidizer + D_oxidizer)^2) * h_oxidizer) / 4;
V_inner_oxidizer = V_oxidizer;
Wall_volume_oxidizer = V_outer_oxidizer - V_inner_oxidizer;

% Mass of Wall

```

```

m_fuel_tank = Wall_volume_fuel*rho_wall;
m_oxidizer_tank = Wall_volume_oxidizer*rho_wall;

% Outer Diameter
OD_fuel = 2*t_fuel + D_fuel;
OD_oxidizer = 2*t_oxidizer + D_oxidizer;

%-----Injector Sizing-----

% Total Quantities
mdot_fuel = mdot / (1 + mix_r);
mdot_oxidizer = mdot - mdot_fuel;

% Short Tube w/ rounded entrance L/D > 3
D = .00157;
Cd = 0.90;
A = pi*(D/2)^2;

DelP_solution_fuel = 0.2*Pc;

Pf_Pox = (rho_oxidizer/rho_fuel)/((mix_r^2));
DelP_solution_oxidizer = DelP_solution_fuel / Pf_Pox;

mdot_fuel_injector_i = Cd * A* sqrt(2 * rho_fuel * DelP_solution_fuel);
mdot_oxidizer_injector_i = Cd * A *sqrt(2 * rho_oxidizer * DelP_solution_oxidizer);

v_fuel = Cd * sqrt((2 * DelP_solution_fuel) / rho_fuel);
v_oxidizer = Cd * sqrt((2 * DelP_solution_oxidizer) / rho_oxidizer);

a2 = 10; % PLAY WITH THIS
a1 = asind(((mdot_oxidizer_injector_i * v_oxidizer) / (mdot_fuel_injector_i * v_fuel)) * sind(a2));

Num_fuel_injectors = mdot_fuel / mdot_fuel_injector_i;

Num_oxidizer_injectors = mdot_oxidizer / mdot_oxidizer_injector_i;

% Display results
fprintf('mdot_oxidizer_injector_i / mdot_fuel_injector_i = %.2f\n', ...
    mdot_oxidizer_injector_i / mdot_fuel_injector_i);
fprintf('a1 = %.2f degrees\n', a1);
fprintf('a2 = %.2f degrees\n', a2);
fprintf('Num Fuel Injectors = %.2f\n', Num_fuel_injectors);
fprintf('Num Ox Injectors = %.2f\n', Num_oxidizer_injectors);
fprintf('Num Fuel Injectors = %.2f\n', Num_fuel_injectors);
fprintf('Num Ox Injectors = %.2f\n', Num_oxidizer_injectors);

```

12. References

[1] Elon Musk. Raptor V3 just achieved 350 bar chamber pressure. X, 2023.

https://x.com/elonmusk/status/1657249739925258240?ref_src=twsr%5Etwfw%7Ctwcamp%5Etweetembed%7Ctwterm%5E1657249739925258240%7Ctwgr%5Edf89957a12cf4eb1d80a02b7cd2efc40e6ebaca%7Ctwcon%5Es1 &ref_url=https%3A%2F%2Fwww.nextbigfuture.com%2F2023%2F05%2Fspacex-raptor-3-engine-is-test-fired-and-has-10-more-thrust.html

[2] S. Turns, An Introduction to Combustion: Concepts and Applications, 3rd edition. New York: McGraw Hill, 2011.

[3] G. P. Sutton and O. Biblarz, Rocket Propulsion Elements, 9th ed. Hoboken, NJ, USA: Wiley, 2017.

[4] J. L. Tuttle and D. H. Blount, "Perfect Bell Nozzle Parametric and Optimization Curves," NASA Reference Publication 1104, May 1983.

[5] R. Ravi, "Bell-Nozzle: This script plots Bell Nozzle (Thrust Optimised Parabolic (TOP) nozzle) for the given area-ratio and throat radius," GitHub repository, 2021. [Online]. Available: <https://github.com/ravi4ram/Bell-Nozzle>.

[6] NASA, "What Was the Saturn V? (Grades 5-8)," NASA, [Online]. Available: <https://www.nasa.gov/learning-resources/for-kids-and-students/what-was-the-saturn-v-grades-5-8/>. [Accessed: 17-Dec-2024].